Structural Study of Proteins Involved in Autophagy

Erik Walinda

2015

Structural Study of Proteins Involved in Autophagy

Erik Walinda

Department of Molecular Engineering
Graduate School of Engineering
Kyoto University

Thesis submitted for the degree of
Doctor of Philosophy

2015

Abstract

Insufficient clearance of proteins with aberrant structure and dysfunctional organelles in non-dividing neural cells has been implicated in severe human neurodegenerative disorders such as Alzheimer, Huntington, and Parkinson disease [1]. Such potentially harmful matter is now thought to be degraded by a complex cellular pathway termed "macroautophagy", commonly referred to as simply "autophagy". However, the molecular details of many steps in this degradation process are still poorly understood.

Selective autophagy consists of four major stages: 1. labeling of matter destined for degradation by covalent attachment of ubiquitin; 2. recognition of this label by receptor proteins; 3. assembly of a vesicle ("autophagosome") around these receptor proteins and finally 4. degradation of the vesicle's constituents by lysosomal degradation. This thesis aims at a molecular explanation of stages 1 and 2 (Fig. 1).

The initial step of the pathway, covalent attachment, of the protein "tag" ubiquitin is achieved by the action of ubiquitin ligases. A prominent example is the ubiquitin ligase parkin, which is responsible for labeling dysfunctional mitochondria for degradation. However, parkin is reported to reside in the cytosol in an inactive state under basal conditions. Indeed, x-ray crystallographic studies established that the catalytic center of parkin is buried inside the structure of parkin, thereby preventing its enzymatic action. This thesis aims at solving this contradiction (section 4). By combining the biophysical methods of NMR, ITC and fluorescence spectroscopy, I show that binding of an agonist (Ser^{65}-phosphorylated ubiquitin) induces partial opening of the catalytic center of parkin, thereby priming the enzyme for catalysis. Interestingly, I found that a part of parkin, the N-terminal ubiquitin-like (UBL) domain adopts an antagonistic role *in cis*, which suggests a regulatory mechanism by competition of agonist and antagonist for the same binding site on core parkin.

After covelent attachment to the autophagic matter, the ubiquitin label is then recognized by autophagy receptor proteins. Two of these receptor proteins – p62 and NBR1 – recognize ubiquitin by similar, yet distinct mechanisms (chapters 2, 3). p62 shows weak ubiquitin binding, whereas NBR1 was found to recognize ubiquitin $\approx$ 100-fold more avidly. NBR1-mediated ubiquitin recognition could be characterized in molecular detail by solving the solution structure of the NBR1 UBA:ubiquitin

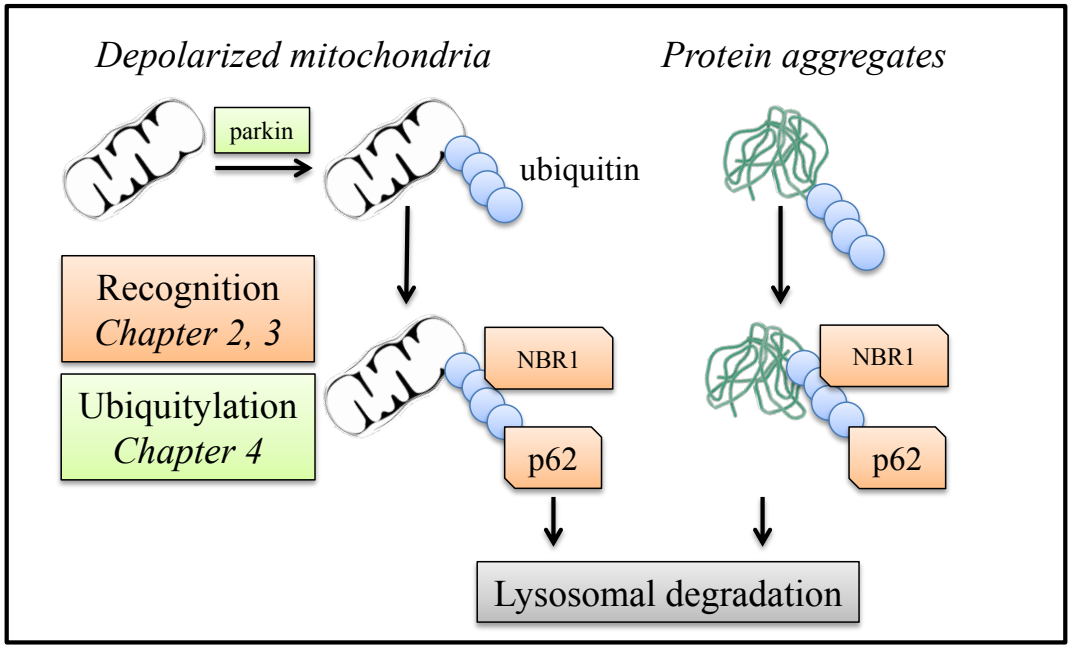

Figure 1: This thesis aims at a molecular explanation of two distinct steps in macroautophagy - substrate recognition and parkin activation.

complex by NMR-spectroscopy. This information was then used to build a model for p62-mediated ubiquitin recognition, which elegantly explains atomic details such as p62-phosphorylation at residue Ser^{403}. Detailed characterization of p62 by relaxation dispersion NMR spectroscopy spawned development of a novel automated method for efficient analysis which I named "Amaterasu" (chapter 3). Although developed for the characterization of p62, this method can be applied to study the dynamics of any protein sample and is already used in the study of three other proteins (polyubiquitin, FABP4[1] and HLA[2]).

In summary, this thesis explains yet unknown molecular aspects of macroautophagy at the steps of cargo recognition and parkin activation. It is my hope that these new insights will deepen the understanding of autophagy, which is a necessary prerequisite for development of novel therapeutic means to enhance autophagy activity.

[1]Fatty-acid binding protein 4
[2]human leukozyte antigen A

Preface

This thesis is submitted to Kyoto University for the degree of Doctor of Philosophy. All research described herein was conducted under supervision of Professor Masahiro Shirakawa, Department of Molecular Engineering, Graduate School of Engineering in the tiem from 2012 to 2015.

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

June 2015
Kyoto, Japan
Erik Walinda

Acknowledgements

This work was conducted under the supervision of Professor Dr. Masahiro Shirakawa. I would like to express my deep gratitude to him for kindly welcoming me as an international student in 2010 and again in 2011. I thank him for his guidance throughout this thesis. Especially, I appreciate providing me with the opportunity to work on these challenging projects in an excellent laboratory environment and always giving me the opportunity to present my results at overseas conferences.

I would like to thank Dr. Kenji Sugase for teaching me nearly everything I know about nuclear magnetic resonance in both theory and practice.

My sincere gratitude goes Dr. Mariko Ariyoshi for encouraging me through difficult problems in biochemistry. This work would not have been possible without the help of my collaborators. I thank Dr. Max Konuma and Dr. Naohiro Kobayashi for sharing with me key practical aspects of data processing and evaluation. I acknowledge the great help of Dr. Satoru Unzai by collecting high-quality analytical ultracentrifugation data on several of my samples.

Furthermore, I express my warmest gratitude to all the people who I had the pleasure to study protein NMR spectroscopy with: Dr. Ayako Furukawa, Dr. Saeko Yanaka and Shoko Mori. I deeply appreciate the fruitful advice and comments of my senior Dr. Daichi Morimoto that I received over the years. I am further deeply grateful to Dr. Shin Isogai to teach me protein structure calculation from NMR data.

I would like to express sincere thanks and warm regards to Yoko Imai for helping me through mountains of paper work for both university and visa-related affairs. I also thank her for continuously providing me with key information about Japanese scholarships for international students.

This thesis would not have been possible without financial support from JASSO (2012-2013) and the Honjo International Scholarship foundation (2013-2015). I am deeply grateful to both organizations. Needless to say, I thank my parents for their never-ending support in all of my endeavours.

List of Publications

Solution Structure of the Ubiquitin-associated (UBA) Domain of Human Autophagy Receptor NBR1 and Its Interaction with Ubiquitin and Polyubiquitin

Erik Walinda, Daichi Morimoto, Kenji Sugase, Tsuyoshi Konuma, Hidehito Tochio and Masahiro Shirakawa
Journal of Biological Chemistry 289.20 **(2014)**: 13890-13902.

Automatic Processing of $R_{1\rho}$-Relaxation Dispersion Data with Amaterasu

Erik Walinda, Mayu Nishizawa, Masahiro Shirakawa and Kenji Sugase
in preparation.

Conformational Rearrangement in Parkin Driven by Phosphoubiquitin Binding

Erik Walinda, Daichi Morimoto, Yutaka Ito, Kenji Sugase and Masahiro Shirakawa
submitted.

Other Publications

The Unexpected Role of Ubiquitin Chains in the Formation of Fibrillar Aggregates

Daichi Morimoto, Erik Walinda, Harumi Fukada, Yu-Shin Sou, Shun Kageyama, Masaru Hoshino, Takashi Fujii, Hikaru Tsuchiya, Yasushi Saeki, Kyohei Arita, Mariko Ariyoshi, Hidehito Tochio, Kazuhiro Iwai, Keiichi Namba, Masaaki Komatsu, Keiji Tanaka and Masahiro Shirakawa
Nature communications 6 **(2015)**.

Ubiquitin Directly Induces Folding Destabilization of Proteins

Daichi Morimoto, Erik Walinda, Kenji Sugase and Masahiro Shirakawa
in preparation.

Contents

Chapter 1

General Introduction

1.1 Proteins

As all matter is composed of atoms, modern science aims to explain all aspects of life comprehensively from the atomic level (biochemistry) to that of the entire organism (physiology). Life is mainly governed by the atoms that compose proteins, which is why enormous effort is currently being undertaken to understand protein function and interactions at atomic resolution. Proteins are extremely versatile molecular workhorses that join forces to fulfill virtually all actions living beings exert. For instance, every time we move a muscle, its contraction is mediated by the active movement of the protein myosin along structural filaments composed of a protein called actin. Proteins can also act as highly-efficient biochemical catalysts in the form of enzymes: food we eat is digested in the stomach by the action of the protein enzyme pepsin. Moreover, proteins can function as molecular messengers that amplify and transmit signals throughout an organism, mediate immune responses and can serve as scaffolds on which other biomolecules can assemble [2, pp.44-50].

Proteins are linear polymers built of a repertoire of twenty proteinogenic amino acids[1] which are connected by peptide bonds. The amino acid sequence of a given protein is determined by its genetic DNA sequence. This DNA sequence is first transcribed into an RNA sequence and subsequently translated into the target amino acid sequence, whereby a sequence of three RNA nucleotides encodes a single amino acid residue of a given protein. This picture, in which DNA serves as the building plan of proteins, RNA acts as a mediator between the two and proteins are the biomolecules that carry out the actual physiological function is commonly called *central dogma of molecular biology* [3].

$$\text{DNA} \rightarrow \text{RNA} \rightarrow \text{Protein}^{2}$$

[1] Rare exceptions such as selenocysteine and hydroxyproline exist.

[2] For brevity, only the simplest form of the dogma is stated here. After the original statement of

With the exception of misfolded or intrinsically disordered proteins, proteins fold into well-defined three-dimensional structures after biosynthesis. As 1972 Nobel Laureate Christian B. Anfinsen discovered, this three-dimensional structure is entirely determined by the amino acid sequence of a given protein [4]. Thus, any physiological function, however complex it may be, must eventually be explicable by the chemistry of twenty seemingly simple amino acids[3].

Indeed, this particular set proves to be chemically extremely versatile: proteinogenic amino acids vary in size, net charge at physiological pH, hydrophobicity, shape, chemical reactivity and hydrogen bonding capability [2, p.45]: the side chains of these twenty amino acids include aliphatic groups[4], aromatic structures[5], uncharged polar groups[6], charged polar groups[7], a sulfhydryl group[8] or simply a mere hydrogen atom[9]. The ionizable functional groups span a pK_a range from 3.1 to 12.5.

Despite this astonishing chemical repertoire, a given protein is rather confined to a specific, static role after biosynthesis, thereby limiting its overall functional versatility, a limitation which is imposed on it by its amino acid sequence. Accordingly, living organisms have evolved another kind of mechanism to fine-tune the chemical composition of their proteins in a highly dynamic and *reversible* manner.

1.2 Protein Modification

In the living cell, not every protein is final in its chemical composition just after biosynthesis. In a process termed *post-translational modification*, proteins can be trimmed or certain functional groups of amino acids can be further modified by the action of highly specific enzymes. Thus, for many proteins, the central dogma is extended to:

$$\text{DNA} \rightarrow \text{RNA} \rightarrow \text{Protein} \rightarrow \text{Modified protein}$$

the dogma, other pathways were found to exist. For example, a virus can replicate its own RNA. Moreover, in a process called reverse transcription, retroviruses such as HIV can even synthesize DNA from RNA [2, p.128].

[3] An important exception is not discussed here: several RNA molecules actively exerting physiological function, e.g. ribozymes (Ribonucleic enzymes, i.e. enzymes composed of RNA) are known. However, proteins carry out the vast majority of physiological processes.

[4] Aliphatic amino acids are alanine, valine, leucine, isoleucine, methionine and proline.

[5] As the name indicates, phenylalanine has a phenyl sidechain. Tyrosine is similar, but carries an additional hydroxyl group. Tryptophan has an indole ring side chain.

[6] Serine and threonine carry hydroxyl groups. Asparagine and glutamine have a terminal carboxamide group.

[7] Histidine has a pK_a value near neutral pH and can be charged positively or negatively, depending on the physiological context. Lysine and arginine carry positive charges at physiological pH. Aspartate and glutamate are negatively charged at physiological pH.

[8] Cysteine is unique in being the only proteinogenic amino acid carrying a thiol group.

[9] The side chain of glycine is simply a hydrogen atom.

For instance, histon proteins, which pack genomic DNA into structures called nucleosomes are subject to methylation of their Lysine and Arginine side chain amino-groups, a process which presumably regulates gene-expression [5].

A more prominent example is the phosphorylation of serine, threonine and tyrosine hydroxyl-groups[10], a reaction catalyzed by the enzymes termed *protein kinases*. The simple transfer of a double negative charge in form of a phosphate group (PO_4^{3-}) from ATP[11] to a hydroxyl group on a target protein is a key regulatory mechanism that can completely switch the activity of many proteins, such as enzymes on and off:

$$\mathrm{R{-}OH + ATP^{4-} \overset{\text{Protein kinase}}{\rightleftharpoons} R{-}O{-}PO_3^{2-} + ADP^{2-} + H^+} \tag{1.1}$$

in which R is a serine, threonine or tyrosine residue on a given protein. Most importantly, the action of the protein kinase on a protein can be reversed by removal of the phosphate group. This is accomplished by a class of enzymes termed *protein phosphatases*:

$$\mathrm{R{-}O{-}PO_3^{2-} + H_2O \overset{\text{Protein phosphatase}}{\rightleftharpoons} R{-}OH + HPO_4^{2-}} \tag{1.2}$$

This reversibility provides the living cell with an effective tool to switch certain cellular functions between active and inactive states in a highly regulated manner by a single, simple mechanism. Many more of such small (<100 Da) protein modifiers including acyl, adenyl and prenyl groups are known.

Since the 1980s it has become clear that another, albeit quite different type of post-translational modification also plays crucial roles in virtually any physiological function. That is, covalent attachment of an *entire protein* to target proteins.

1.3 Protein Ubiquitylation

The Nobel Prize in Chemistry 2004 was awarded to Aaron Ciechanover, Avram Hershko and Irwin Rose *for the discovery of ubiquitin-mediated protein degradation*. They discovered a system, in which intracellular proteins are marked for degradation by covalent attachment of a specific degradation signal. By this degradation system, the cell is able turn off undesired functions of proteins, thereby being an important regulatory switch for virtually every protein and therefore any physiological function. This signal, which tells the proteolytic machinery of the cell what to degrade, was found to be ubiquitin.

Ubiquitin is a small protein of 76 amino acid residues ($M_r \approx 8.5$ kDa, Fig. 1.1).

[10] Phosphorylation of histidine is also known, albeit far less prominent.
[11] Adenosine triphosphate

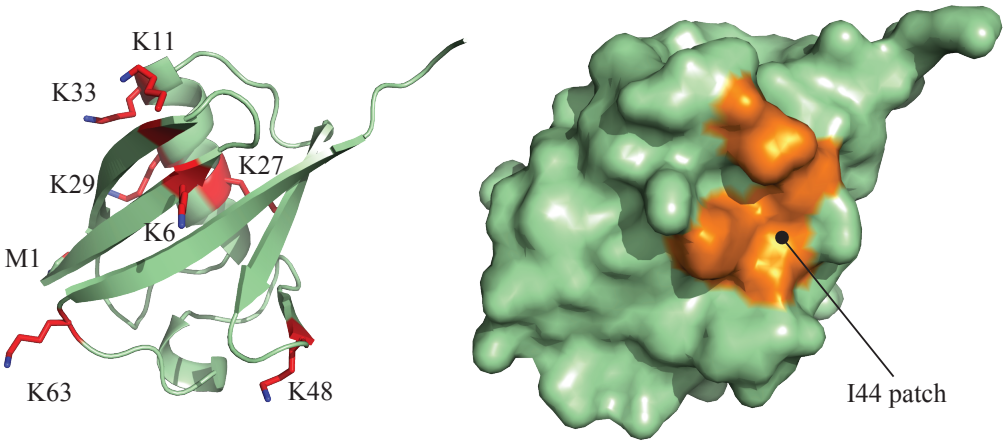

Figure 1.1: Left, cartoon representation of human ubiquitin drawn from its crystal structure (PDB ID 1UBQ). Ubiquitin is a small polypeptide of 76 amino acids. It consists of a five-stranded β-sheet, an α-helix and a short 3_{10}-helix. Eight special amino acids are highlighted. These include seven key solvent exposed lysine residues (Lys-6, -11, -27, -29, -33, -48, 63) as well as the amino-terminal methionine residue. Any one of these globally distributed eight amino groups can be ubiquitylated itself, thereby forming topologically distinct ubiquitin dimers or polymers. Right, surface representation of ubiquitin shown in the same orientation. The Ile^{44} hydrophobic patch (orange) is a common recognition site, which is used by many biomolecules to bind to ubiquitin.

It received its name since it is *ubiquitously* found in virtually all eukaryotic tissues [6]. Indeed, eubacteria and archea do not use the protein ubiquitin. Similar to phosphorylation, ubiquitin can be transferred to target proteins by formation of a covalent bond between a given protein and ubiquitin. This post-translational modification of proteins by a ubiquitin molecule is called *ubiquitylation*[12].

In contrast to phosphorylation, which only requires a single enzyme (a protein kinase) to exert modification of a target protein, a cascade of three enzymes is required to achieve protein ubiquitylation. In the first step of this cascade, the C-terminal carboxyl group of ubiquitin is activated by an ubiquitin-activating enzyme (commonly referred to as "E1 enzyme" or simply "E1"). This activation is facilitated by free energy from ATP hydrolysis resulting in formation of an energy-rich thioester linkage between a key cysteine residue on the E1 and the C-terminal carboxy-group of ubiquitin [6]:

$$\mathrm{E1{-}SH + Ub{-}COOH + ATP \rightleftharpoons E1{-}S{-}CO{-}Ub + AMP + PP_i} \tag{1.3}$$

[12]The term *ubiquitination* is also used.

in which ubiquitin is abbreviated as "Ub" and PP_i refers to pyrophosphate. It is noteworthy that the human genome encodes only two E1 enzymes, most probably owing to the universal nature of this activation step. In the next step, the activated ubiquitin is transferred onto a ubiquitin-conjugating enzyme (commonly referred to as "E2"; the human genome encodes about 50 different "E2s") by transthiolation:

$$\mathrm{E1{-}S{-}CO{-}Ub + E2{-}SH \rightleftharpoons E2{-}S{-}CO{-}Ub + E1{-}SH} \tag{1.4}$$

In the final reaction of the cascade, an ubiquitin-ligase ("E3 enzyme") catalyzes the transfer of the ubiquitin molecule onto the target substrate protein forming an isopeptide bond between the C-terminus of ubiquitin and the ϵ-amino group of a specific lysine residue of a target protein:

$$\mathrm{E2{-}S{-}CO{-}Ub + R{-}NH_3 \overset{\text{E3 enzyme}}{\rightleftharpoons} E2{-}SH + R{-}NH{-}CO{-}Ub} \tag{1.5}$$

in which R is a target lysine amino group on a given protein that is to be ubiquitylated. The human genome encodes ≈ 600 different E3 enzymes allowing for a wide range of substrate specificity. After all, it is the E3 enzyme which has to unambiguously recognize the substrate which is to be ubiquitylated.

1.4 Ubiquitin Ligases

E3 enzymes are further classified into two groups: RING-type and HECT-type ligases [6]. Apart from structural differences of the protein domains involved, the main difference lies in the reaction mechanism: RING-type ubiquitin ligases transfer the ubiquitin moiety directly from the E2 to the substrate, while both E2-Ub and the substrate are bound to the E3. Therefore, the E3 serves as a passive adapter platform on which the ubiquitin transfer from E2 to the substrate can occur. In stark contrast, HECT-type E3 ubiquitin ligases have a catalytically active cysteine residue, which receives the ubiquitin moiety from the E2 to form a E3-Ub intermediate. Then, ubiquitin is transferred from the E3 to the target substrate. Therefore, HECT E3 ligases act according to the two-step reaction:

$$\mathrm{E2{-}Ub + E3\text{-}SH \rightleftharpoons E2{-}SH + E3\text{-}Ub} \tag{1.6}$$

$$\mathrm{E3{-}Ub + R{-}NH_3 \rightleftharpoons E3{-}SH + R{-}NH{-}CO{-}Ub} \tag{1.7}$$

A schematic representation of the complete E1-E2-E3-ubiquitylation cascade is given in Fig. 1.2 with the E3 reacting by a HECT-type mechanism.

Interestingly, RING/HECT hybrid E3 ligases also exist. These ligases are com-

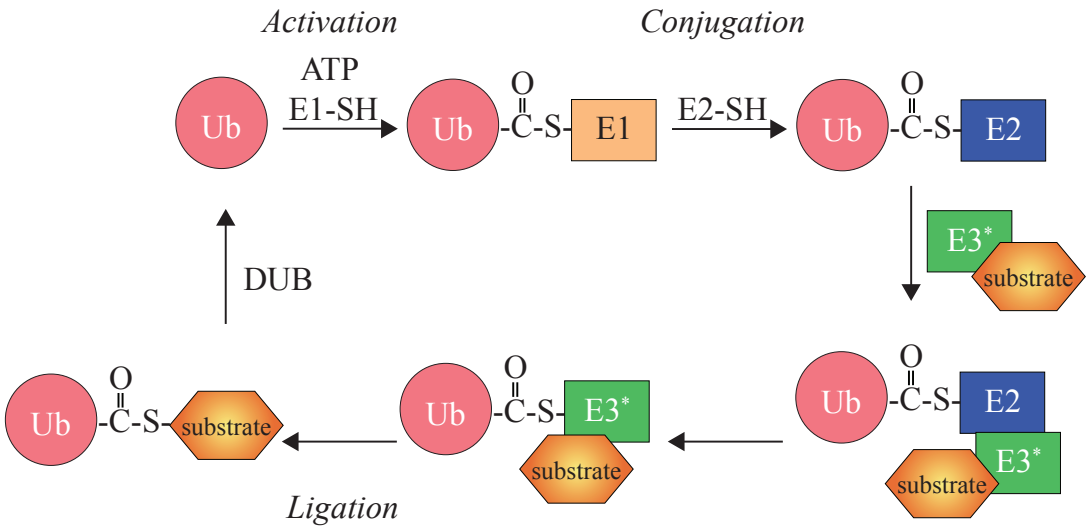

Figure 1.2: Ubiquitin conjugation cascade. Ubiquitin is first activated by a ubiquitin-activating enzyme (E1), transferred to a ubiquitin-conjugating enzyme (E2) and subsequently transferred to a target substrate protein (S) by a ubiquitin-ligase (E3). E3 enzymes can act by a RING or HECT-type mechanism. In this figure, the HECT-mechanism, in which a covalent thioester intermediate between ubiquitin and a key cysteine residue on the E3 enzyme is formed, is displayed. A single cycle of this cascade leads to formation of a monoubiquitylated protein, whereas multiple cycles result in polyubiquitylation of the target. The asterisk specifies the HECT-type nature of the E3 depicted. De-ubiquitylating enzymes (DUBs) can cleave (poly)ubiquitin from the substrate leading to termination of the the ubiquitin-encoded signal.

posed of protein domains of the RING family, but function according to the HECT mechanism, thereby forming a covalent E3-ubiquitin intermediate before transferring the ubiquitin moiety to the substrate. Famous examples of hybrid E3 ligases are the HOIP, the major component of the linear ubiquitin chain assembly complex (LUBAC) and parkin (section 4).

1.5 Polyubiquitin

Most importantly, ubiquitin cannot only be covalently conjugated to *other* proteins, but it itself can also act as a substrate. In this case, ubiquitin itself is ubiquitylated, thereby forming a ubiquitin dimer. Ubiquitin possesses seven solvent exposed lysine residues and thus carries eight amino-groups[13] (Fig. 1.1). In living cells, any of these eight amino-groups can be used to form an (iso-)peptide bond with the C-terminal carboxy-group of another ubiquitin molecule. Since this mechanism can link the C-terminus of a given ubiquitin molecule to any of eight possible sites, it allows for the formation of ubiquitin dimers of different topology (Fig. 1.3). Moreover, the reaction does not necessarily stop after the formation of dimeric ubiquitin. Indeed, polymeric ubiquitin chains of seemingly arbitrary length can be readily generated *in vitro* with recombinant E1, E2 and E3 enzymes. In living cells, very long ubiquitin chains of more than 10 units in length have been observed [6].

1.6 The Ubiquitin Code

The topological versatility of polyubiquitin chains clearly distinguishes ubiquitylation from more classical modifications such as methylation or phosphorylation. Therefore, it has been recently proposed that the distinct quarternary topology that is generated between adjacent ubiquitin units in a ubiquitin chain could act as a *code*. In this "ubiquitin code", the linkage type, for example a ubiquitin chain linked through isopeptide bonds at residue Lys^{48}, stores a certain bit of information which can be subsequently decoded by a specific downstream reader module. Such reader modules are proteins carrying ubiquitin-binding domains with the unique ability to recognize a specific ubiquitin linkage while showing no or very reduced binding towards other linkages. This hypothesis predicts that the ubiquitin code works as follows:

1. Formation of a ubiquitin chain of specific linkage-type (Lys^6, Lys^{11}, ...)

2. Recognition of the specific linkage-type by a reader module

[13] Including its N-terminal Met^1 α-amino group.

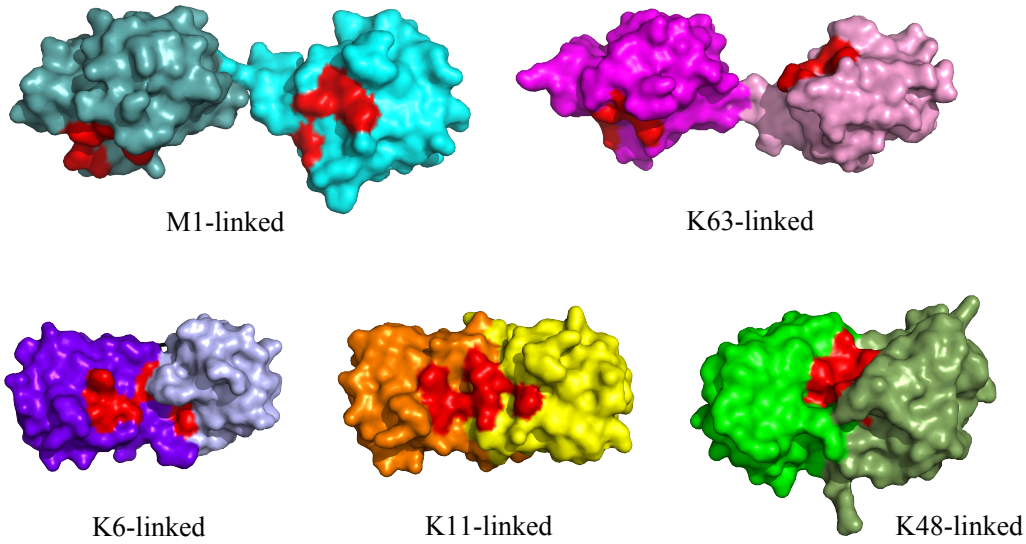

Figure 1.3: Several reported quarternary structures of ubiquitin chains. Different chain linkages result in different relative orientations of the ubiquitin molecules. The Ile^{44} hydrophobic patch is shown in red. Structures are drawn from PDB IDs 2W9N, 2XK5, 3NOB, 1AAR and 2JF5.

Linkage	Function
Met-1 "linear"	NF- κB activation
Lys-6	DNA repair (?)
Lys-11	Protein degradation and signal transduction
Lys-27	Protein degradation (?)
Lys-29	Unknown
Lys-33	Unknown
Lys-48	Protein degradation
Lys-63	DNA repair and signal transduction
Monoubiquitylation	Endocytosis

Table 1.1: The ubiquitin code as deciphered to date. Ubiquitin linkage-function relationships are shown as adapted from reference [7].

3. Exertion of a certain physiological function

Currently, much effort goes into decoding the relationship between specific ubiquitin-linkage and physiological function. Several such relationships have been described and are summarized in Table 1.1. So far, the most thoroughly characterized example which is the one that won the Nobel Prize in Chemistry 2004 is:

1. Attachment of a Lys^{48}-linked ubiquitin chain to a protein

2. Recognition of the polyubiquitin-tag by the ubiquitin-proteasome system

3. Degradation of the target protein by the 26S proteasome

Nevertheless, as can be seen from Table 1.1, many linkages are still of unknown function ("unknown") or the linkage-function relationship is not sufficiently established to date ("?").

The action of E3 ubiquitin-ligases on proteins can also be reversed by enzymes called de-ubiquitinating enzymes ("DUBs"). These act by hydrolyzing the amide bond between the ubiquitin chain and target proteins, thereby removing the specific ubiquitin tag from the target protein. This results in termination of transmission of the ubiquitin-encoded signal. Thus, DUBs counteract E3 enzymes with respect to the ubiquitin-code.

1.7 Ubiquitin-Binding Domains

Transmission of information by the ubiquitin-code naturally requires readout of the information stored in the particular ubiquitin linkage type. Non-covalent recognition of particular types of polyubiquitin (or monoubiquitin) is accomplished by proteins containing ubiquitin-binding domains (UBD). Several NMR and x-ray structures

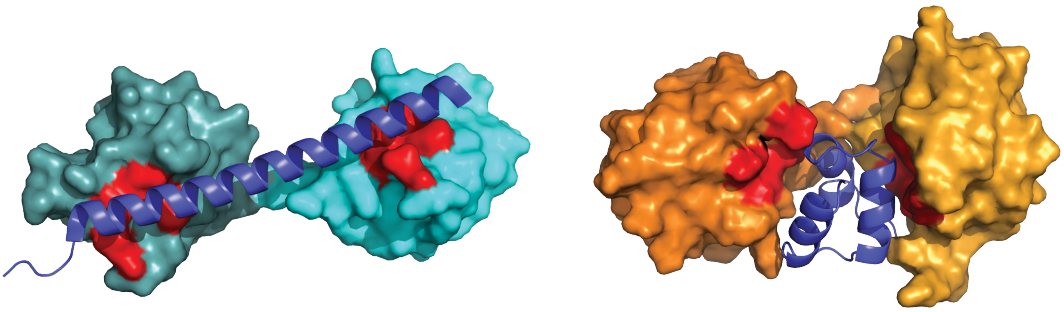

Figure 1.4: Specific recognition of a particular ubiquitin chain linkage-type (K48 or K63-linked diubiquitin). The Ile^{44} hydrophobic patch is shown in red. Structures are drawn from PDB IDs 1ZO6 and 3A1Q.

of ubiquitin-binding domains suggest that these proteins preferentially recognize a specific form of ubiquitin chain, thereby providing critical evidence for the ubiquitin code hypothesis.

In general, ubiquitin can be recognized by a manifold of protein domains. These domains contain α-helical, zinc-finger, plekstrin homology and Ubc-like domains [8]. Examples of ubiquitin-binding proteins specifically recognizing Lys^{48}-linked, Lys^{63}-linked and linear diubiquitin are shown in Fig. 1.4. The difference in quarternary orientation permits for specific recognition of the particular linkage type by different ubiquitin binding domains.

1.8 Protein Degradation

1.8.1 The Proteasome

It has been already briefly mentioned that ubiquitin marks protein for degradation by a large protease complex called "26S proteasome". This barrel-shaped complex is composed of a 19S regulatory and a 20S catalytic subunit. Ubiquitylated proteins can be delivered to proteasomes by adapter proteins such as Dsk2. The 19S regu-

latory subunit of the proteasome specifically binds to polyubiquitin to assure that indeed only ubiquitylated proteins are degraded [2, p.653]. ATP hydrolysis by six ATPases in the 19S subunit supplies the energy required to unfold proteins destined for degradation and deliver these into the 20S catalytic core of the proteasome (Fig. 1.5). The threonine-protease activity of the 20S subunit breaks down substrates into short peptide fragments. Cellular proteases can further process these peptides to yield single amino acids. Notably, ubiquitin itself is not degraded, but cleaved off by the 19S subunit allowing for the recycling of intact ubiquitin. The ubiquitin-proteasome system thus selectively degrades single proteins one at a time. By this orchestrated degradation mechanism, the cell regulates a vast variety of functions including cell-cycle progression, inflammatory response and circadian rhythm [2, p.654]. In addition, proteasomes can dispose misfolded proteins that may otherwise prove toxic to the cell.

1.8.2 Autophagy

An alternative, yet quite different means of protein degradation is known, yet not as well understood as the proteasome system: cells can digest their own cytoplasmic constituents in a process called autophagy[14]. The term autophagy stems from the ancient Greek words *auto* - "self" and *phagein* - "to eat". This process was first found to exist more than 50 years by Christian DeDuve and his coworkers [9].

In contrast to the ubiquitin-proteasome system, which can degrade only proteins that are small enough to fit into the proteasomal barrel, in autophagy, cellular constituents are delivered to and degraded in comparably large organelles called lysosomes. Depending on the type of matter which is to be degraded different pathways and terms are used. In the case of a protein aggregate, the term aggrephagy is used, whereas mitophagy describes autophagic degradation of mitochondria. Nevertheless, in any of these pathways, the lysosome eventually breaks down the substrate matter using its hydrolytic enzymes in an acidic environment (pH = 4-5). A schematic mechanism of this general autophagy mechanism is given in Fig. 1.6.

In the first step of the autophagy pathway, cytosolic matter which is destined for degradation is engulfed by a double-membrane. Autophagy can handle the degradation of not only soluble proteins, but also large protein aggregates and even protein organelles. Therefore, the so formed autophagosome may contain different types of cargo molecules. Autophagosome-lysosome membrane fusion permits access of the autophagic matter to lysosomal hydrolases which break down the constituents of the autophagosome. A large number of Atg ("autophagy") proteins is involved in this process, but has been omitted for clarity in this simplified description. The

[14]Strictly speaking, the correct term is macroautophagy. However, as common practice in the field, in this thesis it will be referred to as simply *autophagy*.

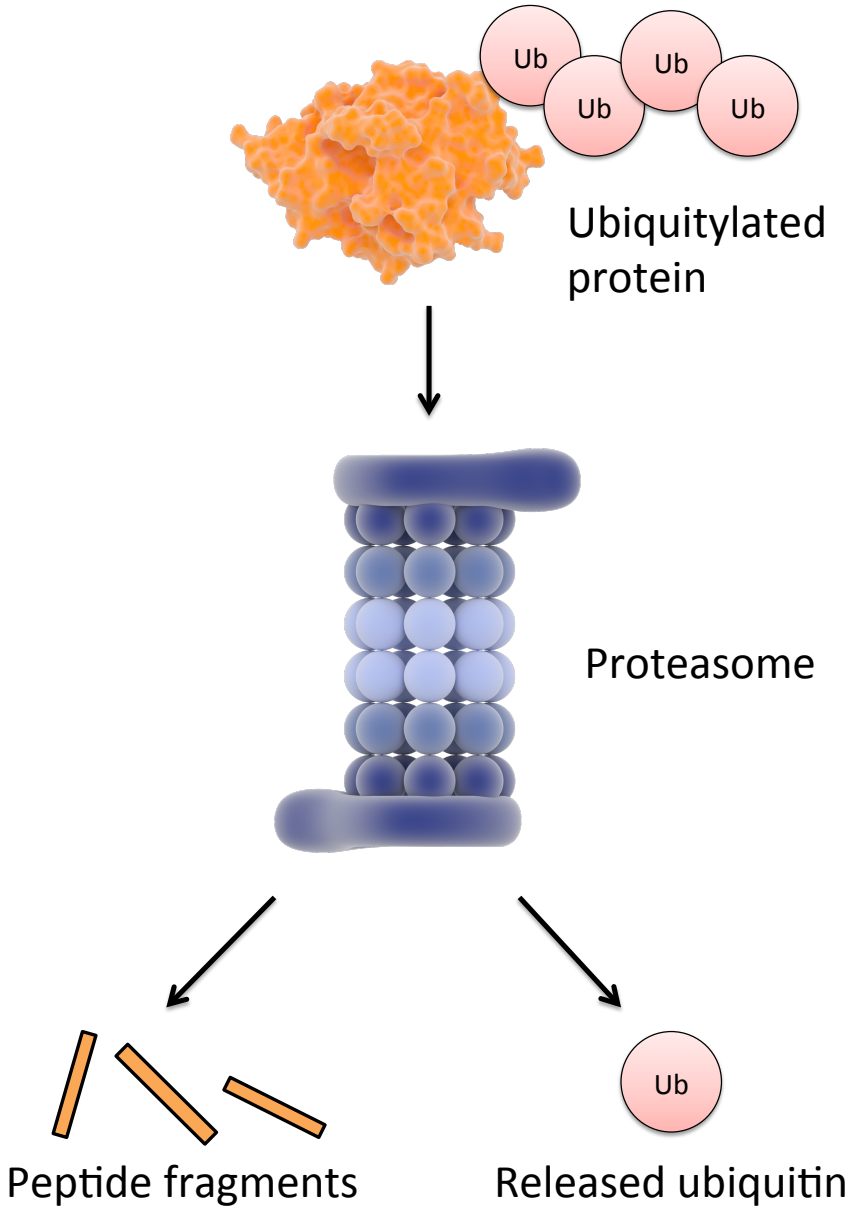

Figure 1.5: The ubiquitin-proteasome pathway. Proteins attached to a (Lys^{48}-linked) polyubiquitin chain are delivered to and cleaved by the 26S proteasome. Ubiquitin is released and recycled by the cell.

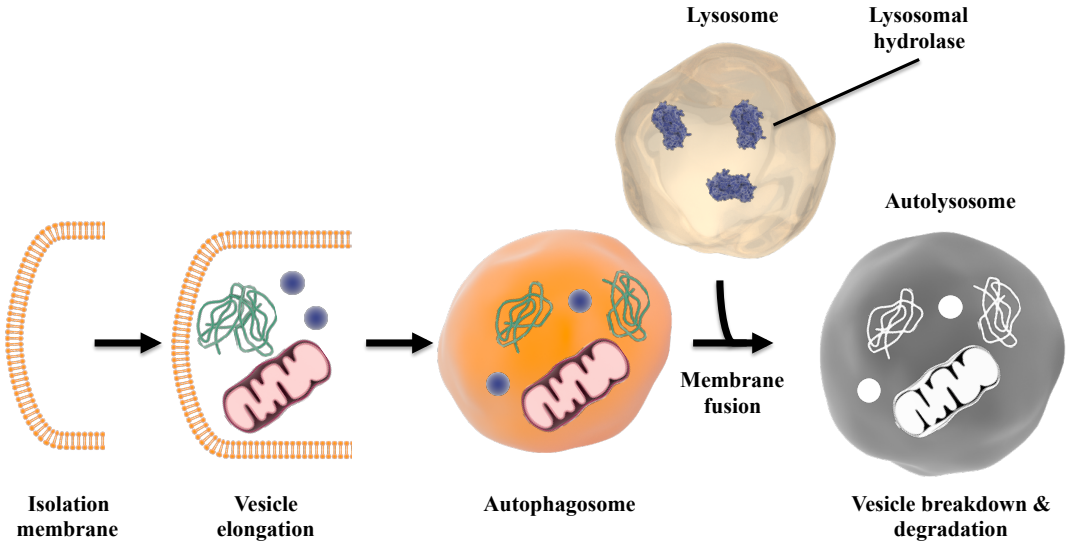

Figure 1.6: Simplified depiction of autophagy. Formation of an isolation membrane inside the cytosol engulfs matter which is to be degraded. An autophagosome formed this way may contain proteins, aggregates and damaged organelles. Subsequent fusion with a lysosome leads to degradation of the constituents of the autophagosome.

question of how autophagy determines what matter is to be degraded is the main topic of chapters 2-3 of this thesis.

1.9 Ubiquitin and Disease

1.9.1 Neurodegeneration

As mentioned above, ubiquitin plays a central role in both major cellular protein degradation systems: the ubiquitin-proteasome system and autophagy. Apart from the completely regulatory function of protein degradation (e.g. cell-cycle progression by time-dependent turnover of specific proteins), protein degradation is of crucial importance to avoid accumulation of misfolded proteins, protein oligomers and protein aggregates, by constantly disposing such matter. Non-dividing cells such as neurons are particularly vulnerable to toxic accumulation of such ill-fated proteins and mainly rely on autophagy for their clearance. Accordingly, dysfunction of both autophagy and the ubiquitin-proteasome system have been found to be associated

Disease	Etiology	Protein Deposited	Pathology
Huntington's disease	Huntingtin	Huntingtin	Intranuclear inclusions Cytoplasmic aggregates
ALS	Sporadic	unknown	Bodina bodies Axonal spheroids
	SOD-1	unknown	(same)
Alzheimer's disease	Sporadic	Aβ peptide tau protein	Plaques & tangles
	APP	(same)	(same)
	Presenilin 1, 2	(same)	(same)
Parkinson's disease	Sporadic α-synuclein	α-synuclein α-synuclein	Lewy bodies (similar)
AR-JP	parkin	(α-synuclein)	Lewy bodies infrequent
	PINK1	(α-synuclein)	unknown
	DJ-1	(α-synuclein)	

Table 1.2: Pathology of common neurodegenerative disorders (based on ref. [11] and [12]).

with many severe human neurodegenerative disorders such as Alzheimer's, Huntington's, and Parkinson's diseases [10].

Many neurodegenerative diseases show degeneration in specific regions of the brain with the observation of abnormal protein aggregates inside the neuron or in the extracellular space [11]. The etiology of the disease can be either sporadic (no known cause) or arise from genetic predisposition. Disease-causing mutations can either directly affect the aggregation-prone protein (for example α-synuclein in Parkinson's disease, Table 1.2), affect a protein on the same pathway as the aggregating protein (for example APP in Alzheimer's disease) or be of still elusive nature (such as superoxide dismutase-1 in ALS). Although the number of sporadic cases greatly outnumbers the rather rare familial forms of the particular disease, the genetic mutations responsible for these rare familial forms are greatly aiding investigation of the underlying etiology.

In Alzheimer's disease (AD), two distinct kinds of protein aggregates are observed: extracellular plaques composed of the Aβ-peptide (a cleavage product from the amyloid precursor protein APP) and intracellular aggregates comprised of the microtubule-associated tau protein (Table 1.2, [11]). In Huntington's disease (HD), the N-terminal region of the Huntingtin protein encodes a repeating glutamine sequence, which becomes increasingly prone to aggregation both *in vivo* and *in vitro* as the length of the poly-Gln sequence increases. In ALS, ubiquitylated aggregates are found in patient brains, although the aggregation-prone protein is elusive (Table 1.2).

Protein aggregates and inclusions as observed in these diseases are now thought to be the final outcome of a complex pathway comprising various intermediates.

According to this hypothesis, a monomeric disease protein adopts an abnormal conformation for example due to misfolding or a covalent modification that favors an alternative conformation. Such a protein may be toxic on itself or may tend to form larger globular oligomers due to the exposure of "sticky" hydrophobic regions. These oligomers are likely to be toxic and can further polymerize into protofibrils and amyloid fibrils the toxicity of which is debated. Finally, such fibrils can be converted into cellular inclusions which are now hypothesized to be cytoprotective [11].

Smaller protein oligomers may still be substrates of the ubiquitin-proteasome system. However, larger matter such as fibrils and inclusions can only be degraded by autophagy. The crucial importance of autophagy in protection against toxic protein aggregates has been shown in studies with healthy mice (carrying no genetic mutations that would predispose them to develop neurodegeneration) in which autophagy was selectively inactivated in the central nervous system. In these mice, loss of autophagy directly caused neuronal cell death and phenotypic symptoms of neurodegeneration [13, 14]. This showed for the first time that in principle anyone is susceptible to neurodegenerative disease: no genetic predisposition is required for the development of neurodegeneration. This model therefore takes the sporadic diseases fully into account. Accordingly, autophagy is the major protective mechanism against protein aggregation-caused neurodegeneration.

This thesis will mainly investigate the molecular mechanisms of autophagic pathways (aggrephagy and mitophagy) and discuss potential consequences of these mechanisms for the research of yet elusive etiology of neurodegenerative diseases. Specifically, proteins involved in aggrephagy are the topic of chapters 2 and 3. Dysfunction of mitophagy has been recently linked to parkinsonism (section 1.9.2) and is the main topic of chapter 4.

1.9.2 Parkinson's Disease

Parkinson's disease (PD) is an idiopathic, slowly progressing degenerative disorder of the central nervous system. Its characteristic motor symptoms include bradykinesia (slowed movement), resting tremor, rigidity, gait impairment and instability of posture [12]. These cardinal motor symptoms are referred to as *parkinsonism*. Although PD is the major cause of parkinsonism, other disorders can show a similar clinical picture. At later stages of the disease, non-motor symptoms, in particular dementia and depression may also occur. Approximately 1% of people above 65 years of age are affected by PD making it the second most common neurodegenerative disorder after Alzheimer's disease [12].

The hallmark of PD pathology is the accumulation of the protein α-synuclein

into inclusions termed *Lewy bodies* in the neuronal cytoplasm (Table 1.2) and the loss of dopamine-producing neurons in the *substantia nigra pars compacta* (SNc). Loss of dopaminergic cells leads to dysfunction of the basal ganglia, a part of the brain responsible for initiation and execution of movements [12]. As treatment, the motor symptoms of PD can be alleviated by administration of the neurotransmitter dopamine.

Until very recently, the existence of heritability in PD was highly doubted. Indeed, PD was the prime example of a sporadic disorder. Nevertheless, recent advances in genetics are beginning to revolutionize this picture. Although heavily dependent on the population in question, in one particular population, more than one-third of PD cases seem to have genetic cause [12]. Mutations in the SNCA gene, which encodes α-synuclein, the major constituent of Lewy bodies, have been observed in both familial and sporadic PD. Several other PD genes including leucine-rich repeat kinase 2 (LRRK2) have been identified [12]. A neurodegenerative disorder that is closely linked to PD and indeed is often clinically indistinguishable from PD is *autosomal-recessive juvenile parkinsonism*.

1.9.3 Autosomal-Recessive Juvenile Parkinsonism

Autosomal-recessive juvenile parkinsonism (AR-JP) is a genetically caused form of parkinsonism. The most common cause of AR-JP are inherited genetic mutations in the gene PARK2, which encodes the protein *parkin* [15]. Parkin is a ubiquitously expressed 465-residue protein, which functions as E3 ubiquitin ligase (section 1.4) which is now believed to function in the degradation of damaged mitochondria (see below). The second most common cause of AR-JP are mutations in the gene PINK1 (PTEN-induced kinase 1). The gene encodes a 581-residue kinase that localizes to mitochondria. Interestingly, PINK1 has been shown to act in the same pathway as parkin, i.e. the clearance of dysfunctional mitochondria and indeed PINK1-linked disease resembles that of parkin. Intriguingly, another gene responsible for AR-JP called DJ-1 also translocates to mitochondria upon exposure to oxidative stress.

AR-JP is distinct from sporadic PD in showing a much earlier mean age of onset. It also shows clinical features that are atypical for PD and lacks the non-motor symptoms of PD. For this reason, it is hypothesized that the pathology of AR-JP might be more restricted than in PD [12]. Although parkin-caused disease shows a neuropathology of dopaminergic cell loss in the SNc similar to PD, it does only rarely show Lewy body pathology [12]. Despite these differences of AR-JP and PD, it is a general hope that a detailed understanding of AR-JP will bring a better understanding of idiopathic PD etiology. The involvement of parkin, PINK1 and DJ-1 all hint towards the importance of mitochondrial quality control in parkinsonism.

Accordingly, the parkin-PINK1 pathway is a very active area of research.

1.9.4 Role of Mitophagy in AR-JP

Mitochondria are oval intracellular organelles of the approximate size of a bacterium. Like bacteria, they have an outer and an inner membrane. Thus, there are two compartments, the space between the outer and inner membranes and the space engulfed by the inner membrane which is called the mitochondrial matrix. Mitochondria are the site of the two main energy-providing metabolic pathways of the cell, the citric acid cycle and oxidative phosphorylation. The biochemical role of the citric acid cycle is to generate high-energy electrons in the form of NADH and $FADH_2$. During oxidative phosphorylation, the exergonic flow of electrons from NADH[15] and $FADH_2$[16] to the ultimate electron acceptor O_2 (thereby reducing oxygen to water) through three large protein complexes embedded in the inner mitochondrial membrane generates energy which is in turn used to pump protons out of the mitochondrial matrix into the cytoplasm [2, pp.525-528]. The arising unequal proton distribution creates an electric potential gradient across the inner mitochondrial membrane with the matrix side being charged negatively relatively to the cytosolic side. To restore the original charge distribution, protons flow back into the mitochondrial matrix through the enzyme complex ATP synthase driving the phosphorylation of ADP to yield the energy-rich molecule ATP. Thus, the production of ATP (oxidative phosphorylation) is coupled to the oxidation of fuels (citric acid cycle) by the proton gradient across the inner mitochondrial membrane [2, pp.526].

Uncoupling of the proton flow from ATP production will cause the protons to flow back into the mitochondrial matrix independent of ATP production. This results in a reduced mitochondrial membrane potential $\Delta\Psi_m$. Mitochondria with reduced $\Delta\Psi_m$ are called "depolarized" and are considered as "unhealthy" by the cellular machinery.

PINK1 (PTEN induced putative kinase 1) selectively localizes to depolarized mitochondria thereby sensing the state of the mitochondrion: healthy or unhealthy. Under normal conditions, PINK1 is localized to the mitochondrial surface. However it is rapidly degraded. Upon mitochondrial depolarization, degradation of PINK1 is prevented by a yet unknown mechanism and PINK1 can exhibit its kinase activity.

PINK1 acts upstream of the E3 ubiquitin ligase parkin [16] and phosphorylates both parkin and ubiquitin at the homologous residue Ser^{65} [17]. This serine is conserved between ubiquitin and parkin's N-terminal UBL domain. Parkin is a ubiquitin ligase (E3) that resides in the cytosol under normal conditions in an inactive form. Phosphorylation of parkin and ubiquitin by PINK1 is required for

[15] Nicotinamide adenine dinucleotide
[16] Flavin adenine dinucleotide

both releasing the latent E3 activity of parkin, as well as triggering its translocation onto depolarized mitochondria. Subsequently, parkin polyubiquitylates mitochondrial proteins such as VDAC and mitofusin-2 with a preference for proteins located on the mitochondrial outer membrane [18]. This event induces mitophagy, leading to degradation of depolarized, dysfunctional mitochondria. Thus, parkin is cytoprotective by protecting the cell from accumulation of low-quality depolarized mitochondria, the accumulation of which is associated with production of excess reactive oxygen species (ROS) such as the toxic superoxide (O_2^-), a major cause of oxidative stress. It has been shown that the substantia nigra pars compacta, which is most dramatically affected in both PD and AR-JP is particularly weak to oxidative stress [19], which indicates that the SNc may be particularly vulnerable to dysfunction of mitochondrial quality control.

Parkin is a multidomain protein composed of a N-terminal ubiquitin-like (UBL) domain, an $\approx$ 70 residue flexible linker, an atypical RING domain (RING0) and a RBR-type arrangement of RING domains (Fig. 1.8). The crystal structure of parkin has been reported and shows parkin in a closed conformation. Moreover, parkin shows no measurable ubiquitin-ligase activity in its full-length form, whereas removal of the N-terminal UBL-R0 fragment (residues 1-219) releases its E3 activity. Thus, parkin usually adopts an auto-inhibited, inactive conformation due to internal interactions [20].

In living cells, parkin activation requires both phosphorylation of Ser^{65} of its UBL domain and phosphorylation of ubiquitin. However, no experimental *in vitro* data are available so far. Moreover, it is not, clear how phosphorylation of ubiquitin converts parkin into an active form. Investigating the possibility of a direct interaction of parkin with ubiquitin and the possible impact of ubiquitin and parkin phosphorylation events is the main topic of chapter 4.

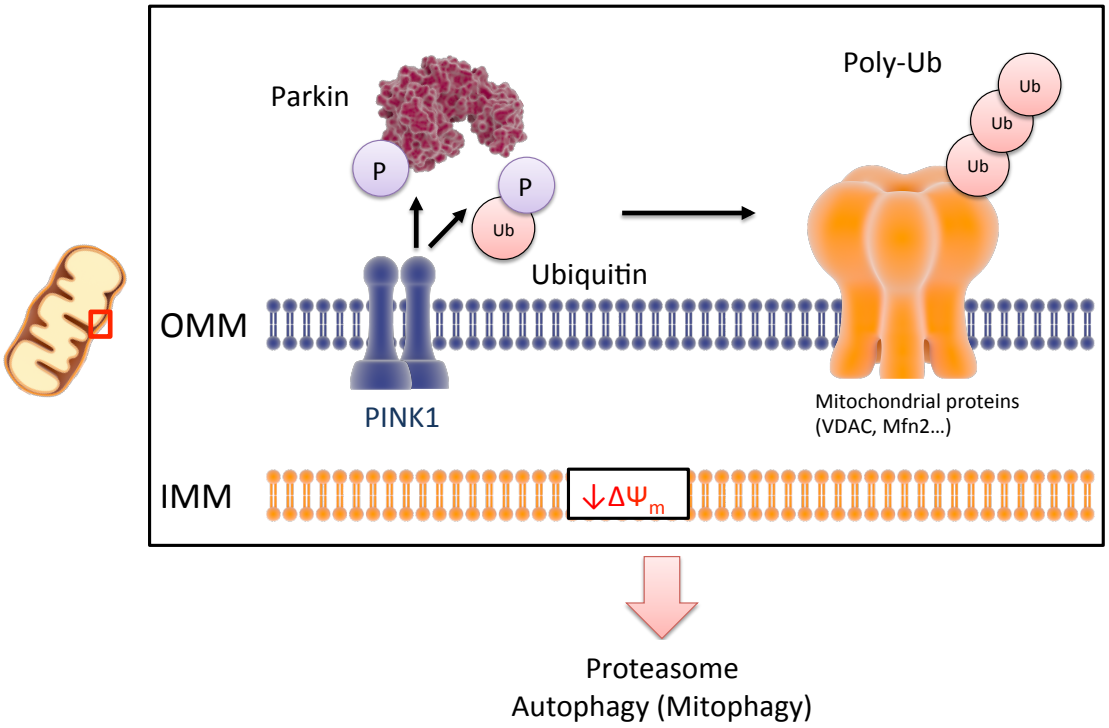

Figure 1.7: PINK1-parkin-mediated degradation of mitochondrial proteins and mitochondria by proteasomal degradation and mitophagy.

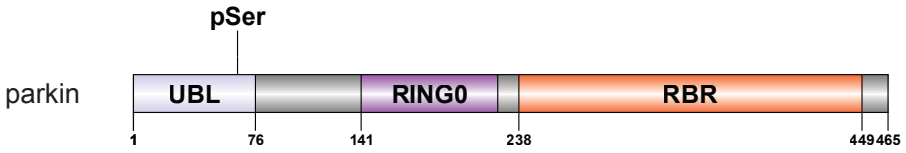

Figure 1.8: Domain architecture of parkin from *rattus norwegicus*. The N-terminal UBL domain and the adjacent linker and R0 domains exhibit an auto-inhibitory function on the catalytic center embedded in the C-terminal RBR domain and also occlude the E2 binding site on parkin. Residue Ser^{65} is phosphorylated by PINK1 upon mitochondrial depolarization.

Chapter 2

Recognition of Ubiquitin by Autophagy Receptor NBR1

2.1 Abstract

NBR1[1] is a multi-domain protein which is often found in ubiquitin-positive inclusion bodies which are a hallmark of many neurodegenerative diseases. Recently, NBR1 was confirmed as a receptor protein in autophagy. Owing to its high similarity in domain architecture to the well studied autophagy receptor p62[2], it is believed that NBR1 similarly binds to (poly)ubiquitylated autophagic substrates and delivers this cargo to autophagosomes for lysosomal degradation. However, in this study, me and my collaborators unexpectedly found that NBR1 displays key differences from p62 in its UBA structure and accordingly in its interaction with ubiquitin. The main structural differences are found on helix α-3, which experiences a farther tilt from helix α-2 and is extended by approximately one turn in NBR1. This precludes a p62-type selfdimerization of the NBR1 UBA domain and thereby results in a significantly higher affinity for monoubiquitin than p62 UBA. Derivation of the NBR1 UBA-ubiquitin complex structure by NMR spectroscopy revealed that the negative charge of the non-conserved side chain in front of the conserved MGF motif in the UBA plays an important auxiliary role in the recognition of ubiquitin by raising the ubiquitin-binding affinity of the UBA. Moreover, by using NMR and ITC experiments in conjunction, I was able to show that NBR1 UBA binds to each ubiquitin unit of a polyubiquitin with similar affinity and by the very same surface used for binding to monoubiquitin. These results imply that the UBA domain of

[1]NBR1 stands for "neighbor of BRCA1 gene 1". The name of this gene simply stems from the proximity of the location of the NBR1 gene on the chromosome to the tumor suppressor gene BRCA1. At the time that the gene name "NBR1" was chosen, the function of the protein encoded by this gene was completely unknown. There is no known relation of NBR1 to breast cancer, so at present the name may appear confusing.

[2]The gene of p62 is also known as SQSTM1 ("sequestosome 1").

NBR1 lacks a particular polyubiquitin linkage-type specificity, which agrees well with the nonspecific polyubiquitin linkages observed in intracellular ubiquitin-positive inclusions. Consequently, the results of this study implicate that the key structural differences between NBR1 UBA and p62 UBA result in a much higher affinity of NBR1 for ubiquitin. This indicates that NBR1 may bind ubiquitylated autophagic substrates more efficiently than p62.

2.2 Introduction

Insufficient degradation of intracellular protein aggregates has been observed in various severe human neurodegenerative disorders such as Alzheimer, Huntington, and Parkinson disease [10]. In eukaryotic cells, proteins can be degraded by two major pathways. The first one is the ubiquitin-proteasome pathway with the second one being autophagy (section 1.8.2). The most crucial difference between these pathways is the cargo that they can handle: the ubiquitin-proteasome pathway can only degrade proteins that, when unfolded, fit into the opening of the proteasome barrel which measures approximately 13 Å in diameter [21].

On the other hand, autophagy pathways degrade not only compact proteins, but also large aggregates of proteins (aggrephagy) and even entire organelles such as mitochondria (mitophagy). Autophagy is evolutionarily conserved and entails the formation of a double-membrane vesicle (autophagosome) in the cytosol. Thereby, a certain section of the cytosol is sealed in a new vesicle, whereby the certain constituents of the cytosol are engulfed. In the next step, this autophagosome fuses with a lysosome leading to the degradation of the autophagosome constituents by the action of lysosomal hydrolases under acidic conditions [22] (section 1.8.2).

One major function of autophagy is the supply of nutrients to the cell under conditions of starvation (starvation-induced autophagy). However, a second important function of autophagy is completely independent of nutrient-stress: this role of autophagy, termed constitutive autophagy, constantly sequesters damaged organelles and protein aggregates which could prove toxic to the cell [22]. This continuous removal of potentially toxic molecules is particularly important for neurons and accordingly the brain is most severely affected by lysosomal disorders [9]. Particular challenges for neurons in disposing dysfunctional organelles or protein aggregates are their large cytoplasm (both axonal and dendritic) [9] and their inability to undergo cell division. Indeed, the vital importance of the continuous removal of cytosolic protein aggregates in neurons was recently highlighted by studies in mice, which conclusively showed that loss of autophagy in neurons leads to accumulation of cytosolic, ubiquitylated protein aggregates, neuronal cell death and symptoms of neurodegeneration [13, 14]. Most importantly, the ubiquitin-proteasome system

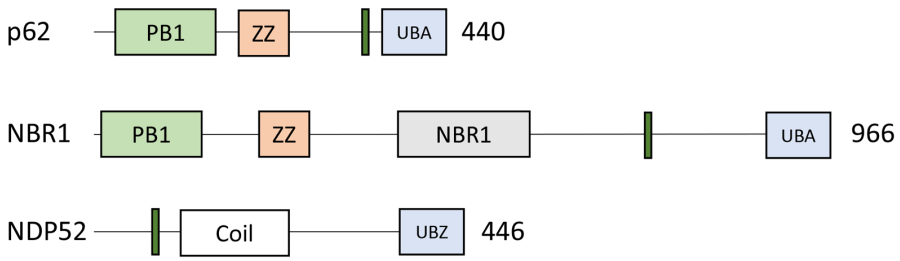

Figure 2.1: Mammalian autophagy receptors p62, NBR1, and NDP52 (adapted from ref. [23]). p62 and NBR1 show a similar overall architecture. The small green rectangle indicates the light-chain-3 interacting region (LIR).

functioned normally in the neurons of these mice. Therefore it could be clearly shown that it is the insufficient activity of constitutive autophagy which results in the inability to degrade protein aggregates leading to a gradual accumulation of such aggregates over time. This in turn triggers neuronal cell death and subsequently neurodegeneration [13, 14]. Therefore, these studies established a direct link between malfunction of constitutive autophagy and the onset of neurodegeneration.

While the ubiquitin-proteasome pathway has been reported to specifically deliver proteins marked by Lys^{48}-linked polyubiquitin to proteasomes for degradation, autophagy has been long believed to choose its substrates somewhat randomly. Nevertheless, the recent discovery and characterization of *autophagy receptor proteins* [22], implies the existence of more selective autophagy pathways. Although these pathways are still poorly understood, intriguingly it appears that is again *ubiquitin*, which marks target substrates for degradation [23]. Therefore it is now believed that in general autophagic substrates are not directly recognized by the cellular machinery. Instead, ubiquitin is used to mark such matter for autophagic degradation and autophagy receptors then recognize ubiquitin, thereby indirectly recognizing matter to be degraded.

So far, three mammalian autophagy receptors which contain ubiquitin-binding domains have been described: p62, NBR1 and NDP52. Interestingly, p62 and NBR1 show a similar domain architecture implying a similar function: they consist of an N-terminal PB1 domain, a ZZ-like zinc finger domain, a light-chain-3 interacting region and a C-terminal ubiquitin-associated (UBA) domain (section 1.7).

This domain organization is highly reminiscent of receptor proteins in the ubiquitin proteasome pathway such as Dsk2 or HR23A. This similarity has lead to the hypothesis that p62 and NBR1 may bind to autophagic substrates through their C-terminal ubiquitin-associated (UBA) domain and shuttle these substrates to autophagosomes by docking onto the autophagosomal membrane protein LC-3[3] via their light-chain-3 interacting region (LIR) motif [22].

[3]LC-3 is the abbreviated name of microtubule-associated protein 1 light chain 3.

p62 has already been intensively studied *in vivo* and is known to accumulate in ubiquitin-positive inclusions which are found in tissues affected by neurodegenerative diseases. Indeed, it is already used as a histological marker for intracellular inclusion bodies (sometimes called "p62-bodies"). Furthermore, the UBA domain of p62 has already been characterized by structural biology using methods of both x-ray crystallography and nuclear magnetic resonance [24, 25, 26].

In contrast, NBR1 was identified as an autophagy receptor only very recently and has not been studied by structural biology in detail yet [27]. In particular, it has not been established, if and how NBR1 binds to ubiquitin, how selective this interaction is and if the UBA shows a preference for a certain type of polyubiquitin. For these reasons, I set out to characterize the ubiquitin-associated (UBA) domain of NBR1, in particular in comparison to p62 to understand why mammals express both of these apparently similar autophagy receptors.

2.3 Experimental Procedures

2.3.1 Sample Preparation

First, the DNA sequence of the UBA domain of human NBR1 (amino acid residues 913–959) was cloned into a pGEX-6P-1 expression vector (GE Healthcare) using the restriction sites for BamHI and XhoI. The gene of the NBR1 UBA domain was a kind gift from Dr. Keiji Tanaka and Dr. Masaaki Komatsu (Tokyo Metropolitan Institute of Medical Science). All plasmids containing point mutations of either the NBR1 UBA domain or ubiquitin were constructed by PCR. The NBR1 UBA domain was expressed as a GST[4] fusion protein in *Escherichia coli* strain BL21(DE3) in LB[5] or (for isotope labeling purposes) in M9 minimal media containing [^{15}N]-ammonium chloride and [^{13}C]-glucose. GST-[NBR1 UBA] was first purified by Glutathione-Sepharose 4FF (GE Healthcare) column chromatography. After cleavage of the GST affinity tag by the PreScission protease (GE Healthcare), NBR1 UBA was further purified by size-exclusion chromatography on a HiLoad 16/60 Superdex 75pg column (GE Healthcare). Ubiquitin and its point mutants were expressed and purified as described previously [28]. Lys^{48}- and Lys^{63}-linked diubiquitin was prepared by enzymatic reaction using the enzymes UBA1, E2–25K, and Ubc13-Mms2 as described previously [28] and further purified by ion exchange on a Source-S column (GE Healthcare). In the case of Lys^{48}-linked diubiquitin, K48C and G76C ubiquitin mutants were used to prevent the formation of circular Lys^{48}-linked polyubiquitin chains in the enzymatic reaction.

[4] Glutathione S-transferase
[5] Lysogeny broth

2.3.2 NMR Spectroscopy

A [^{13}C, ^{15}N]-labeled sample of the UBA domain of NBR1 was prepared in 20 mM potassium phosphate, pH 6.6, 5 mM potassium chloride, 1 mM EDTA[6], 1 mM benzamidine, 1 mM DTT[7], 0.02% sodium azide, and 4% D_2O at a protein concentration of 1.2 mM in 5-mm Shigemi tubes. NMR spectra were acquired on a Bruker Avance 600 MHz NMR spectrometer at a temperature of 298 K.

After acquisition of double-resonance hydrogen-nitrogen and constant-time hydrogen-carbon HSQC[8] spectra, assignments for main and side-chain resonances were obtained from a series of the triple resonance experiments: HNCACB, CBCA(CO)NH, HBHA(CO)NH, H(CCO)NH, CC(CO)NH, and HCCH-TOCSY. DSS[9] was used as the chemical shift reference compound.

The free induction decays were apodized, zero-filled and Fourier-transformed using NMRPipe [29]. For many experiments with the exception of NOESY spectra, the resolution in the indirect dimension(s) was improved by linear prediction. The resulting spectra were analyzed using MAGRO and CCPN software [30, 31, 32]. Backbone resonances were assigned by a combination of manual assignment and automatic assignment using MARS [33], whereas side-chain resonances were assigned completely manually.

Proton-proton distance information was derived from three-dimensional [^{13}C]- and [^{15}N]-edited NOESY-HSQC spectra. NOESY cross-peaks were manually picked from the respective root resonances in the HSQC spectra and automatically assigned in the course of the structure calculation by the iterative CANDID algorithm implemented in CYANA [34]. Backbone torsion angles were predicted by TALOS+ [35] using the assigned chemical shifts of H_α, C_α, C_β, H^N and N. The final structure calculation was performed by CYANA 2.1 using NOE-derived distance and TALOS+ based dihedral restraints yielding a structural ensemble with statistics given in Table 2.2.

2.3.3 NMR Titration Experiments

[^{1}H, ^{15}N] HSQC spectra of 40 μM NBR1 UBA were acquired by gradually increasing the concentration of the ligand ubiquitin or diubiquitin. In order to correctly derive the stoichiometry of binding, the following definition was made: in the case of diubiquitin, the concentration of the ligand was defined as the concentration of monomeric ubiquitin units in the dimer. The normalized chemical shift perturbation

[6] Ethylenediaminetetraacetic acid (EDTA) and benzamidine were included to prevent sample degradation by metallo- and serine-proteases, respectively.

[7] Dithiothreitol (DTT) was added for reduction of cysteine residues.

[8] Heteronuclear single quantum coherence

[9] 4,4-Dimethyl-4-silapentane-1-sulfonic acid

was calculated as:

$$\Delta_{normalized} = \sqrt{(\Delta_H)^2 + \frac{1}{2}(\frac{\Delta_N}{7})^2} \tag{2.1}$$

where Δ_{H} and Δ_{N} are the chemical shift differences in the ^{1}H and ^{15}N dimensions, respectively.

Dissociation constants for the binding of the UBA domain to (di)ubiquitin were obtained by separately fitting the chemical shifts in the ^{1}H and ^{15}N dimensions to a classical binding model using the dissociation constant K_d as a global parameter in the fitting program GLOVE [36]. The classical binding model is given by:

$$\Delta_{obs} = \frac{\Delta_{fb}}{2[A_0]}[K_d + [A_0] + [B_0] - \sqrt{(K_d + [A_0] + [B_0])^2 - 4[A_0][B_0]}] \tag{2.2}$$

in which $[\mathrm{A_0}]$ and $[\mathrm{B_0}]$ are the concentrations of analyte protein and titrant protein, respectively. Δ_{obs} denotes the observed chemical shift change, and Δ_{fb} is the difference in chemical shift of the analyte protein in the free and complexed form.

The uncertainty in the K_d values was estimated by Monte Carlo methods using 100 synthetic data [37]. I also considered the uncertainty in the protein concentrations, which were determined using UV absorption and calculated extinction coefficients. Hence, a possible uncertainty in the protein concentration of the analyte protein of 10% was incorporated into the error analysis.

For the binding of NBR1 UBA to itself (self-association), the K_d value was estimated by gradually diluting a concentrated sample from 1.1 mM to 8 μM and following the resulting chemical shift changes. The concentration dependent chemical shifts were fitted to a selfdimerization model using GLOVE:

$$\Delta_{obs} = \frac{\delta_{dimer} - \delta_{monomer}}{2[A]}[K_d + 2[A] - \sqrt{(K_d + 2[A])^2 - 4[A]^2}] \tag{2.3}$$

in which [A] is the concentration of the NBR1 UBA domain.

2.3.4 Residual Dipolar Couplings

[^{1}H, ^{15}N] residual dipolar coupling (RDC) constants were measured from in-phase anti-phase (IPAP) [^{1}H, ^{15}N] HSQC experiments. The protein was weakly aligned along the static magnetic field using Pf1 Phage (ASLA Biotech) at a concentration of 12.5 mg / ml. Residual dipolar coupling constants were obtained from the difference of the splitting in the ^{15}N-dimension of IPAP spectra of NBR1 UBA in the presence ($\mathrm{D_{NH}}$ + $\mathrm{J_{NH}}$) and absence ($\mathrm{J_{NH}}$) of alignment medium (phage).

The resulting RDC constants of all available amide resonance pairs were independently analyzed in REDCAT and PALES [38, 39, 40] software. For structure validation of the NBR1 UBA domain, RDC constants from α-helical (i.e. well-structured) residues were used to calculate the Q-factor of the CYANA-derived structure.

Residual dipolar couplings for the UBA domain and ubiquitin in the UBA-ubiquitin complex were similarly obtained from in-phase anti-phase [^{1}H, ^{15}N] HSQC experiments. For these experiments, concentrations of 0.5 mM ^{15}N-labeled protein and 2 mM non-labeled ligand protein were used[10]. Again, 12.5 mg/ml Pf1 phage was used as the alignment medium. 150 mM sodium chloride was added to alleviate strong electrostatic interactions between proteins and the negatively charged phage.

2.3.5 Structure Determination of the UBA-Ubiquitin Complex

Intermolecular NOEs were obtained from a filtered NOESY experiment as described previously [41] on a Bruker Avance 800-MHz NMR spectrometer using a sample consisting of 4.5 mM non-labeled ubiquitin and 1 mM [^{13}C, ^{15}N]-labeled NBR1 UBA. At these concentrations, 99.9 % of NBR1 UBA are expected to be in the complexed form at equilibrium.

Side-chain assignments for unliganded ubiquitin were obtained from the Biological Magnetic Resonance Bank (BMRB) database (entry 6457). Side-chain assignments for both ubiquitin and NBR1 in the complexed form were obtained by following the carbon-hydrogen chemical shift correlations in constant-time HSQC spectra on complex formation.

The final structural ensemble of the NBR1 UBA-ubiquitin complex was calculated by HADDOCK [42] using ambiguous restraints derived from chemical shift perturbation data, orientational restraints derived from RDC data, and unambiguous distance restraints derived from NOEs. In the HADDOCK protocol "active" residues were chosen by considering chemical shift perturbation significance (above average) and filtered by relative solvent accessibility (cutoff: 43%) as calculated by the program NACCESS [43]. Residual dipolar coupling constants were used as susceptibility anisotropy (SANI) restraints in the docking protocol.

[10]Given the dissociation constant of NBR1 and ubiquitin as determined independently from NMR and ITC experiments, at these concentrations, more than 99 % of labeled (observed) protein is in the complexed form at equilibrium. Therefore, RDC constants are assumed to be unperturbed by the presence of residual isotope-labeled unbound protein.

Experiment		Pulse program (Bruker)
^{1}H, ^{15}N-HSQC		hsqcetfpf3gpsi2
^{1}H, ^{13}C-HSQC		hsqcctetgpsp
CBCANH	backbone	hncacbgp3d
CBCA[CO]NH	backbone	cbcaconhgp3d
HBHA[CO]NH	sidechain	hbhaconhgp3d
H[CCO]NH	sidechain	hccconhgp3d2
CC[CO]NH	sidechain	hccconhgp3d3
HCCH-TOCSY	sidechain	hcchdigp3d

Table 2.1: Assignment experiments for the NBR1 UBA domain.

2.3.6 Isothermal Titration Calorimetry

Isothermal titration calorimetry (ITC) experiments were conducted at room temperature (298 K) on a MicroCal ITC200 system. All protein samples were dialyzed against ITC buffer (20 mM HEPES, pH 7.5, 100 mM sodium chloride, 0.1 mM TCEP[11]) overnight and thoroughly degassed before each experiment. Samples were incubated at room temperature before each experiment. The concentration of NBR1 UBA in the sample cell was 72 μM, and 1 mM ubiquitin was injected into the cell at 3-min intervals. Experiments using point mutants of either ubiquitin or NBR1 were conducted using the same parameters. As in the NMR titration experiments, for Lys^{48}- and Lys^{63}-linked diubiquitin, the concentration of ubiquitin (1 mM) was defined as the concentration of monomeric ubiquitin units in the dimer. The resulting data were processed using Origin 7 (MicroCal Software, Inc.). Errors in the derived data represent the standard deviation of three independent experiments.

2.4 Results

2.4.1 Solution Structure of the NBR1 UBA Domain

After establishing the protein expression and purification protocol for NBR1 UBA, I prepared an [^{13}C, ^{15}N] isotope- labeled sample comprising residues 913–959 of human NBR1 and under guidance of Dr. Kenji Sugase conducted a series of triple resonance NMR experiments to obtain assignments of the backbone and side chain nuclei of the UBA domain. Using a number of double and triple resonance experiments (Table 2.1) all backbone and 95.7 % of the hydrogen, carbon and nitrogen nuclei in the NBR1 UBA domain could be unambiguously assigned (Fig. 2.2).

Next, I measured carbon- and nitrogen-edited 3D NOESY spectra to obtain proton-proton distance information to use in the structure calculation protocol. Using these distance restraints from ^{13}C- and ^{15}N- edited NOESY spectra in con-

[11] Tris(2-carboxyethyl)phosphine

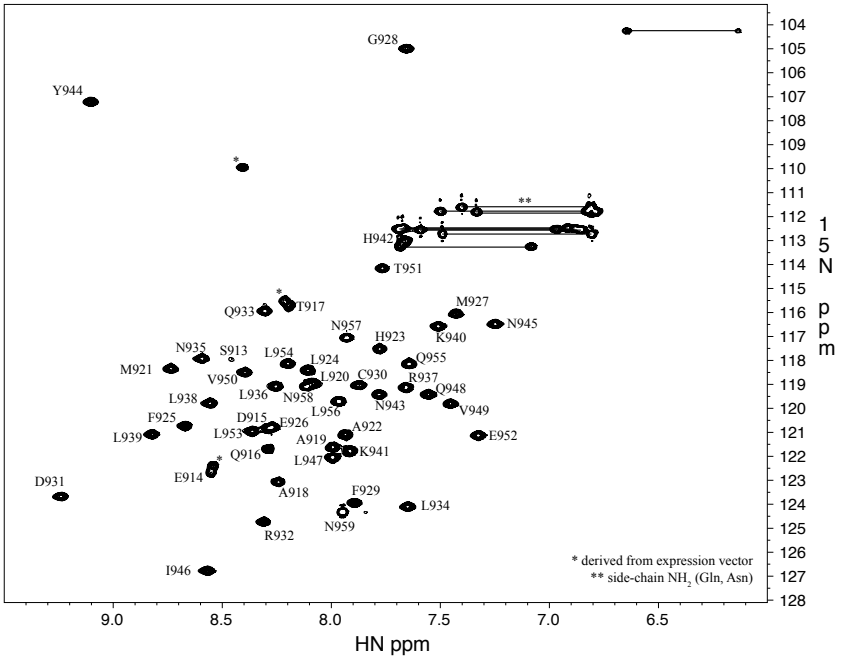

Figure 2.2: Fully assigned ^{1}H, ^{15}N- HSQC spectrum of the NBR1 UBA domain. Horizontal lines connect sidechain NH_2-groups.

junction with dihedral restraints obtained from chemical shifts using the TALOS+ package, I was able to calculate the solution structure of the NBR1 UBA domain. Statistics of the structure calculation are given in Table 2.2.

The final 20 structures of minimum energy converged well with a backbone r.m.s.d. of 0.27 ± 0.16 Å for residues 914–958 (Fig. 2.3 and Table 2.2).

I then measured residual dipolar couplings to validate the structural ensemble. To compare experimental RDC constants with the structural data, the average structure of the CYANA ensemble (Fig. 2.3) was calculated (Fig. 2.4). Then, RDC constants were predicted from the averaged CYANA ensemble and compared with experimentally measured RDCs using the program REDCAT [38]. The structure determined by CYANA (Fig. 2.4) agreed well with the experimental RDC constants (Fig. 2.5) as shown by the low Q-factor of 0.16.

The well converged residues of the NBR1 UBA domain fold into a bundle of three α-helices connected by two loops each comprising 4 and 3 residues (Fig. 2.4). This fold of three α-helices has been described before and is the canonical fold of UBA domains [25]. Accordingly, the solution structure of NBR1 UBA shows a high backbone similarity to and can be well superimposed on the crystal structure of the UBA domain of autophagy receptor p62 (Fig. 2.6). The corresponding C_α-backbone r.m.s.d. value is 1.9 Å as measured by Dali [44].

The main differences between the UBA domains of NBR1 and p62 are found on helix α-3 (Fig. 2.6). This helix encloses a larger angle with helix α-2 (16° difference)

Distance Restraints	
Total	1055
Sequential $[\|i-j\|] = 1$	542
Medium range $[\|i-j\|] \leq 4$	323
Long range $[\|i-j\|] > 4$	190
Statistics for Structure Ensemble Calculation	
Backbone atoms r.m.s.d.	0.27 ± 0.16 Å
All heavy atoms r.m.s.d.	0.81 ± 0.17 Å
Number of dihedral angle violations ≥ 3 °	None
Number of distance violations ≥ 0.2 Å	None
Ramachandran plot statistics (%)	
Residues in most favored regions	86.2 %
Residues in additional allowed regions	13.8 %
Residues in generously allowed regions	0 %
Residues in disallowed allowed regions	0 %

Table 2.2: Structure calculation of the NBR1 UBA domain.

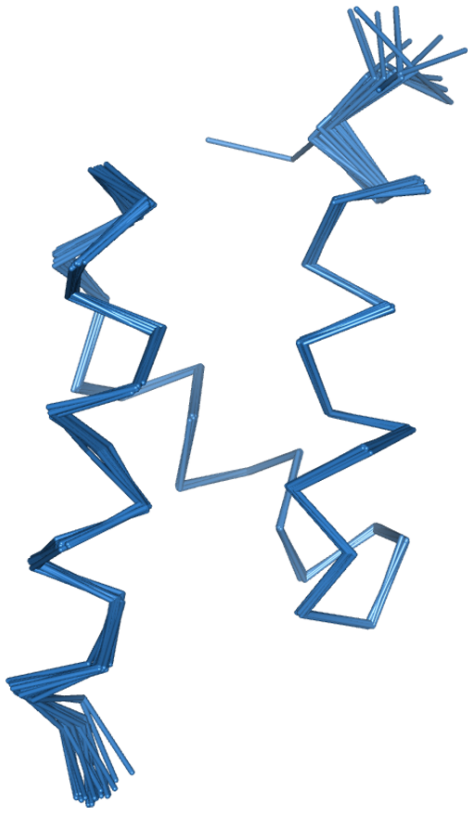

Figure 2.3: Ensemble of the 20 minimum energy structures of human NBR1 UBA.

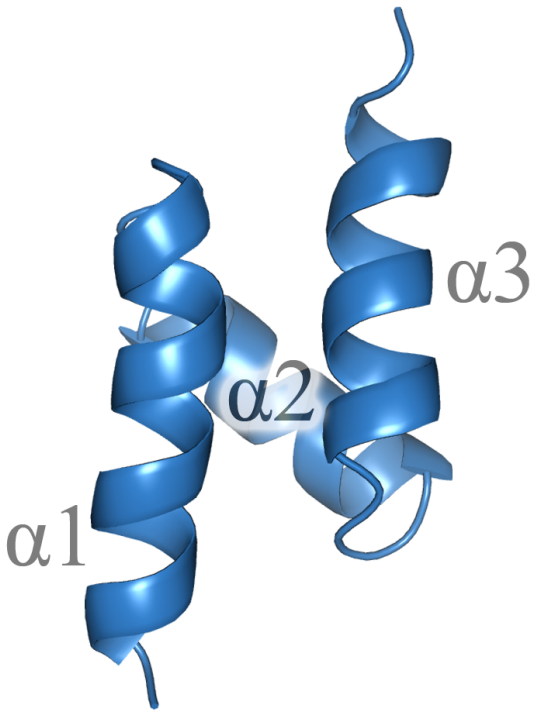

Figure 2.4: Schematic representation of the averaged structure calculated from the minimum energy ensemble.

and is extended by approximately one turn (4 residues) in NBR1 as compared with p62. Intriguingly, in the case of p62, this C-terminal region forms key interactions which lead to a strong dimerization tendency of the UBA domain. Therefore, I hypothesized that the extension of helix α-3 in the UBA domain of NBR1 might inhibit a similar strong dimer formation.

2.4.2 Dimerization Mechanism of the NBR1 UBA domain

To examine whether the NBR1 UBA domain forms a homodimer as had been previously reported for p62 [25], I prepared a concentrated sample of NBR1 UBA (1.1 mM) and by diluting this sample acquired [^{1}H, ^{15}N]-HSQC spectra of the UBA domain at various concentrations. As a result, the amide cross-peaks shifted in a concentration-dependent manner, indicating chemical exchange (Fig. 2.7). Resonance lines did not significantly broaden during the experiment indicating the exchange process was in fast exchange on the NMR timescale. The total chemical shift changes were rather small ($\leq$ 0.12 ppm, Fig. 2.8).

Several residues on helix α-1 and α-3 showed above-average chemical shift changes upon dilution. This observation indicated that NBR1 UBA might form a dimer in solution. However, the possibility of a higher oligomeric state (e.g. trimer, tetramer, ...) could not be ruled out completely. For this reason, my collaborator Dr. Satoru Unzai performed an analysis using analytical ultracentrifugation. Both

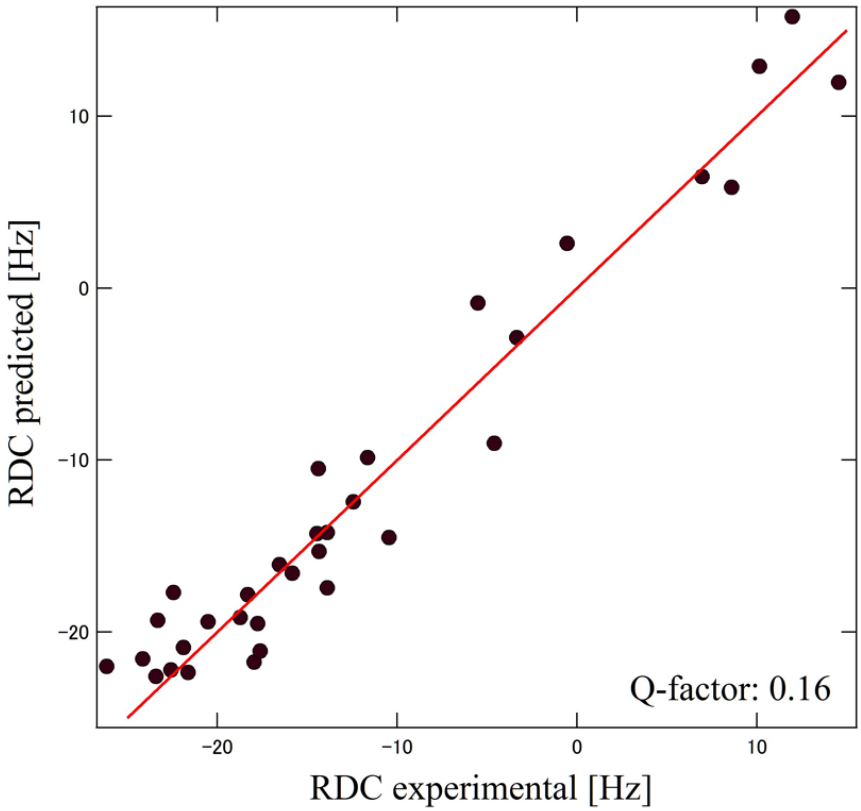

Figure 2.5: Correlation plot of RDCs calculated from the averaged solution structure of NBR1 UBA and experimentally determined RDC constants. The red line indicates perfect correlation.

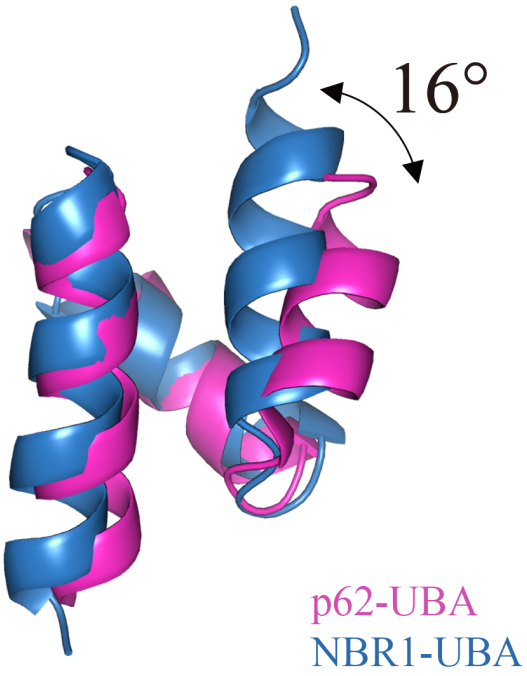

Figure 2.6: Overlay of the UBA domain structures of NBR1 (blue, this study) and that of p62 (magenta, drawn from PDB ID 3BOF). In the case of NBR1, helix α-3 is tilted further away from helix α-2 and is extended by about one turn (4 residues) as compared with p62.

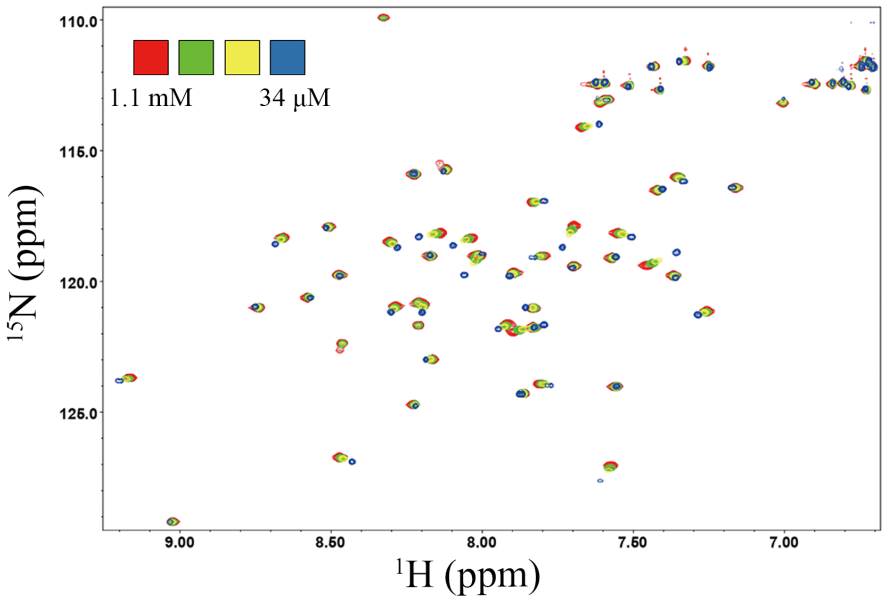

Figure 2.7: Amide chemical shift changes upon gradually diluting a concentrated sample of NBR1 UBA from 1.1 mM to 34 μM.

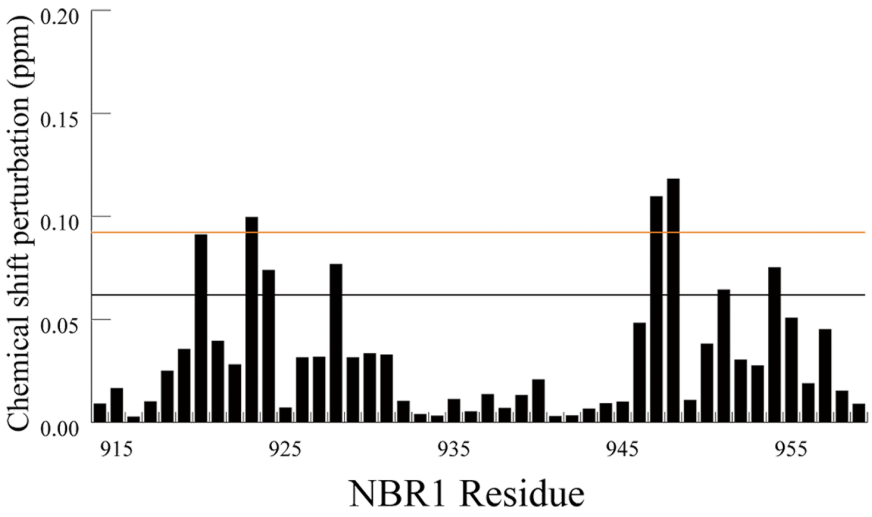

Figure 2.8: Chemical shift changes of Fig. 2.7 as a function of amino acid residue.

UBA domain	Method	K_D [μM]
NBR1	Analytical ultracentrifugation	363
	Nuclear magnetic resonance	465
p62	Analytical ultracentrifugation	10
	Nuclear magnetic resonance	3.5

Table 2.3: Dimerization affinity of the NBR1 UBA domain as determined from NMR and analytical ultracentrifugation data. For comparison, the dimerization affinity of p62 UBA is shown. These data are taken from ref. [25].

sedimentation-velocity and sedimentation-equilibrium experiments conclusively supported the hypothesis that NBR1 UBA exists in a monomer-dimer equilibrium in solution. From the chemical shift data, it was possible to derive the dissociation constant for the dimerization of NBR1 UBA by fitting the data to a self-dimerization model (eq. 2.3). Furthermore, Dr. Unzai was able to determine the dissociation constant from analytical ultracentrifugation data (Table 2.3).

The data from both NMR and analytical ultracentrifugation conclusively suggest that the dimerization of NBR1 is a relatively weak interaction with a dissociation constant of approximately 400 μM as compared with the dimer dissociation of p62 UBA (Table 2.3). This K_d value is also in fine agreement with the observed fast-exchange regime (Fig. 2.7) which in general suggests dissociation constants of the order of hundreds of micromolar. Notably, the surface used for dimerization of the UBA domains markedly differs between NBR1 and p62 (Fig. 2.9).

Both p62 and NBR1 are commonly found in autophagosomes [45]. Furthermore, they are known to bind to each other via their N-terminal PB1 domain (Fig. 2.1). Thus, I considered the possibility that the UBA domains of p62 and NBR1 may be frequently in spatial proximity. Since both of these UBA domains tend to form

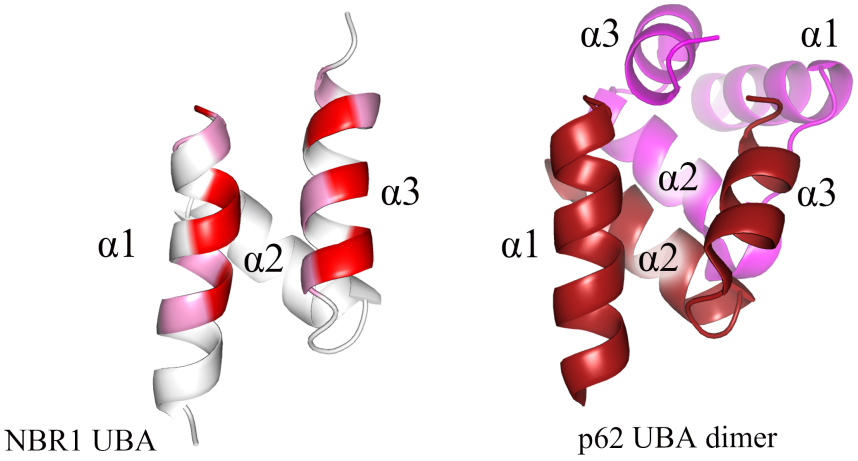

Figure 2.9: Comparison of the dimerization interface of the NBR1 UBA domain with the crystal structure of the p62 UBA dimer (PDB ID 3B0F).

dimers on their own - albeit with different affinity (Table 2.3), it seemed possible that an association between the UBA domains, i.e. the formation of a p62-NBR1 UBA heterodimer, might also occur. However, a binding experiment by NMR spectroscopy precluded this possibility (Fig. 2.10).

Taken together, these results indicate that, although NBR1 UBA has the ability to form a dimer on its own, it does so by a different surface than p62 and with much lower affinity. Therefore, the UBA domains of p62 and NBR1 differ in not only their overall structure (Fig. 2.6), but also in their dimerization properties (Table 2.3). These thoughts led to the hypothesis that NBR1 might interact with ubiquitin in a manner distinct from p62.

2.4.3 Identification of the NBR1 UBA-Ubiquitin Interface

The interaction of NBR1 and ubiquitin was investigated by NMR spectroscopy. Here, I performed NMR titration experiments using [^{15}N]-labeled NBR1 UBA and unlabeled ubiquitin. On addition of the unlabeled ubiquitin to the sample of labeled NBR1, most amide cross-peaks exhibited gradual shifts with only minor line broadening (Fig. 2.11). However, some cross-peaks, such as that of Gly^{928} and Leu^{954}, showed distinct broadening during the course of the titration. Indeed, in spectra at intermediate ligand concentrations, these peaks are broadened to an extent that makes them unresolved in the HSQC spectrum, since the broadening causes the signal intensity to decrease to the noise level of the spectrum. At the end of the titration (at high ligand concentrations), the resonance lines of these peaks reappeared and sharpened again. These features indicated that the binding of ubiquitin by NBR1 is a process of the intermediate exchange regime on the NMR timescale,

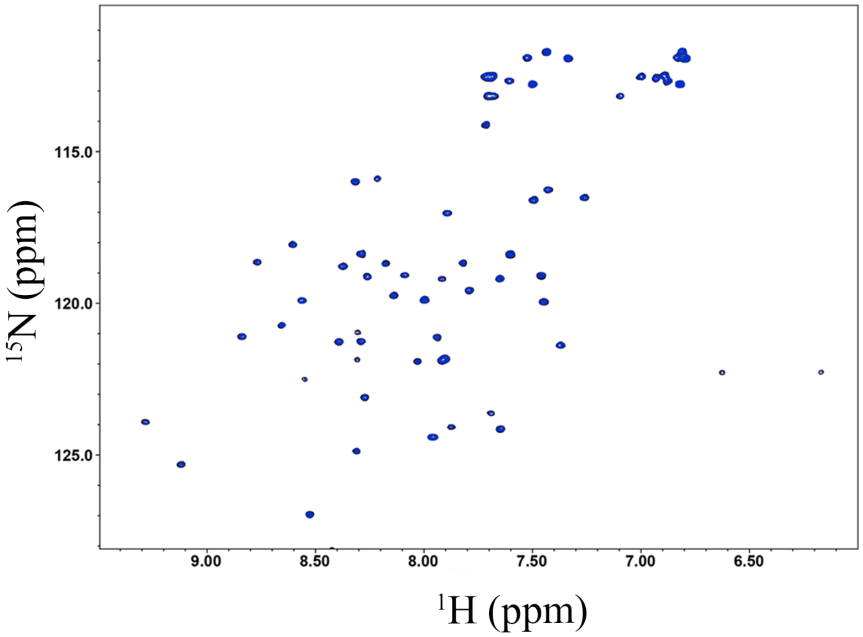

Figure 2.10: [^{1}H, ^{15}N]-HSQC spectra of 100 μM NBR1 UBA (black) and a mixture of 100 μM NBR1 UBA and 400 μM p62 UBA (blue). The spectra are identical. Thus no p62-NBR1 UBA heterodimer was detected.

although for the majority of peaks, the line broadening is not significant so that they appear to be in fast-exchange.

Two regions of NBR1 UBA displayed significant (above average) chemical shift changes upon binding to ubiquitin. The first region is centered at the loop at the end of helix α-1 around the residue Gly^{928}. The second region is formed by the residues of helix α-3, whereby the most significant chemical shift change is exhibited by amino acid Leu^{954} (Fig. 2.12).

These chemical shift perturbation data are mapped onto the cartoon representation of the NBR1 UBA domain in Fig. 2.13. The residues exhibiting the most significant chemical shift changes are highly conserved residues of regions that have previously be named "MGF motif" and "dileucine motif" (Fig. 2.16). Taken together, these observations suggest that NBR1 UBA uses a canonical, evolutionarily conserved surface to achieve binding of ubiquitin [46].

To evaluate the binding surface on the partner, ubiquitin, I performed to opposite NMR titration experiment, this time using [^{15}N]- labeled ubiquitin and the unlabeled NBR1 UBA domain. As Fig. 2.14 shows, this experiment revealed two clusters on ubiquitin showing major chemical shift perturbation due to NBR1 binding. One of these clusters is centered at residue Ile^{44}, the other one being located around Val^{70}. These data indicated that NBR1 UBA binds the so-called "hydrophobic patch" on ubiquitin which has been defined as the amino acid residues Leu^{8}, Ile^{44}, His^{68} and Val^{70} (Fig. 2.19). This further suggested that the interaction between NBR1 UBA

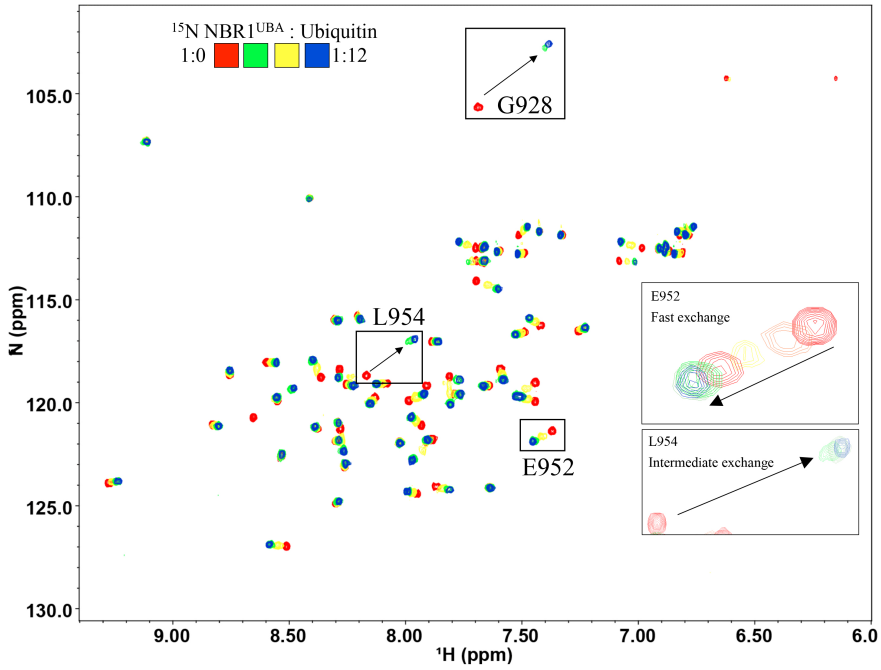

Figure 2.11: [^{1}H, ^{15}N]-HSQC spectra of 40 μM NBR1 UBA at gradually increasing concentrations of ubiquitin. Boxed regions show example peaks with significant (Gly^{928}, Leu^{954}, intermediate exchange regime) and only minor line broadening (Glu^{952}, fast-exchange regime).

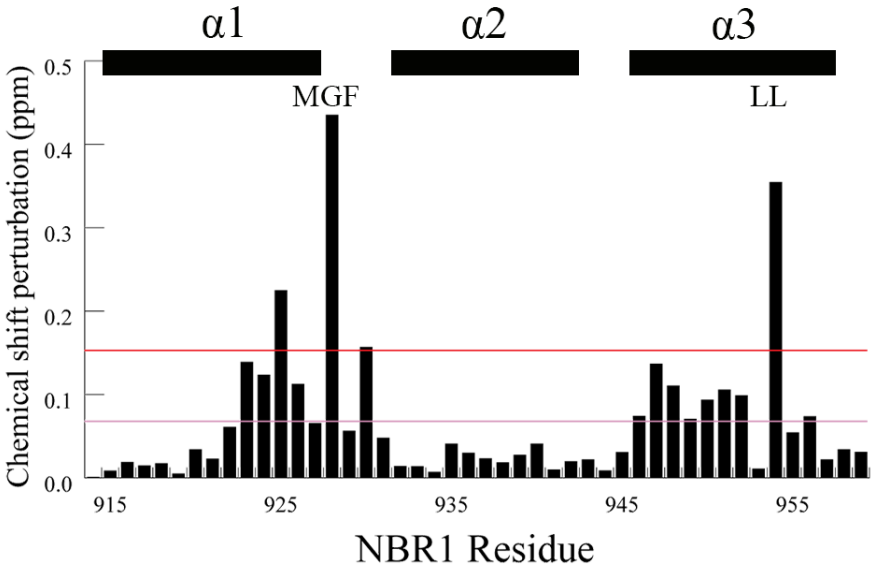

Figure 2.12: Normalized chemical shift perturbation values of NBR1 UBA on binding to ubiquitin plotted as a function of the amino acid residue. The evolutionarily conserved MGF and LL motifs and α-helices are indicated. The horizontal lines are colored as follows: magenta, $\delta_{\mathrm{av}} \leq$ chemical shift perturbation (CSP) $< \delta_{\mathrm{av}} + 1\ \sigma$; red, CSP $\geq \delta_{\mathrm{av}} + 1\ \sigma$.

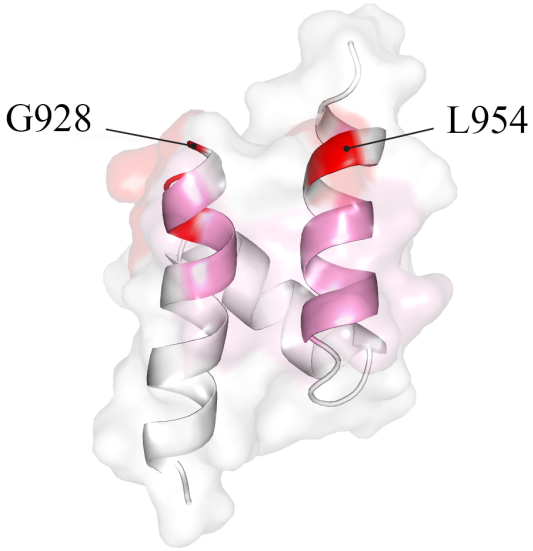

Figure 2.13: Surface representation of the chemical shift changes mapped on the solution structure of NBR1 UBA.

and ubiquitin occurs through canonical surfaces [25, 7, 47].

2.4.4 Affinity of NBR1 UBA for Ubiquitin

Next, I was interested in the affinity of NBR1 UBA for ubiquitin. To determine the dissociation constant for this interaction, I fitted the chemical shift perturbation data to a classical binding model (eq. 2.2) using GLOVE [36] under guidance of Dr. Tsuyoshi Konuma. This yielded a global K_d of 3.0 ± 0.9 μM (Fig. 2.24, A). This value was in fine agreement with the observed intermediate exchange which in general roughly indicates dissociation constants in the range of 10–100 μM [48, p. 782]. Nevertheless, this value is in stark contrast to the previously reported low affinity of the UBA domain of autophagy receptor p62 for ubiquitin of 540-750 μM [26, 49]. Therefore, in the context of autophagy receptor proteins, NBR1 UBA binds to ubiquitin with a comparatively high affinity.

I then used isothermal titration calorimetry (ITC) to evaluate the binding of NBR1 UBA to ubiquitin. Upon injection of ubiquitin into the sample cell containing NBR1 UBA, a significant amount of heat was released. The reaction remained purely exothermic until saturation at the end of the titration (Fig. 2.17).

Fitting indicated that NBR1 UBA binds to ubiquitin with a reaction enthalpy of -4.9 ± 0.1 kcal mol^{-1} and a dissociation constant of 4.3 ± 0.4 μM. This value is in line with the K_d value obtained from NMR analysis. Collectively, these data show that NBR1 UBA has a comparatively high affinity for ubiquitin.

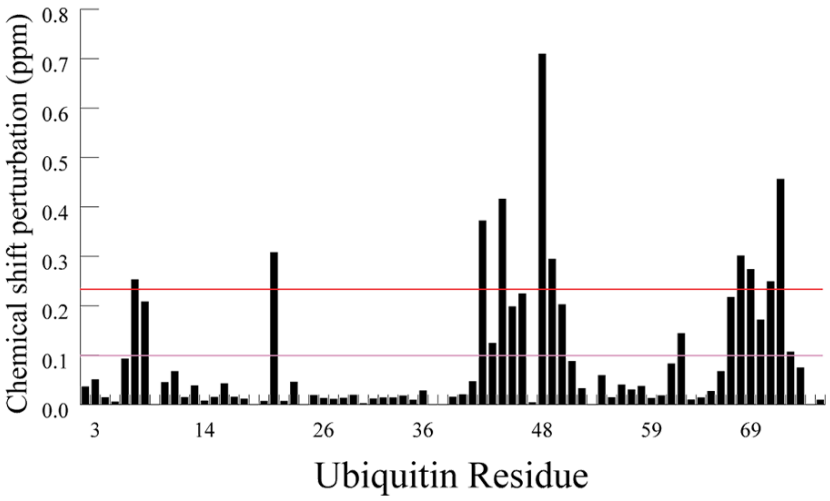

Figure 2.14: Ubiquitin backbone amide chemical shift changes on complex formation with NBR1 UBA. The horizontal lines are colored as follows: magenta, $\delta_{\mathrm{av}} \leq$ chemical shift perturbation (CSP) $< \delta_{\mathrm{av}} + 1\ \sigma$; red, CSP $\geq \delta_{\mathrm{av}} + 1\ \sigma$.

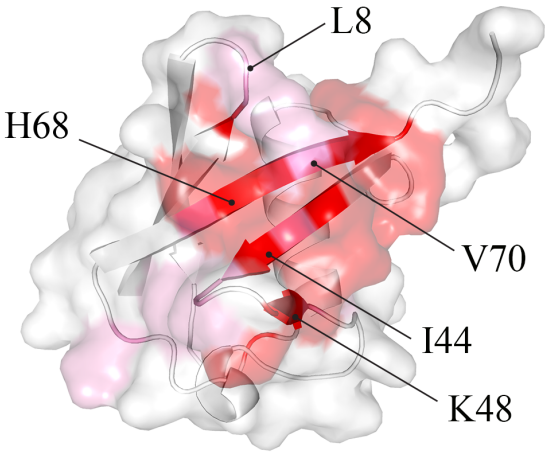

Figure 2.15: Surface representation of Fig. 2.14 mapped on the structure of ubiquitin drawn from PDB ID 1UBQ. Magenta highlights residues with an above average chemical shift perturbation, whereas red refers to a CSP of $\geq + 1\ \sigma$.

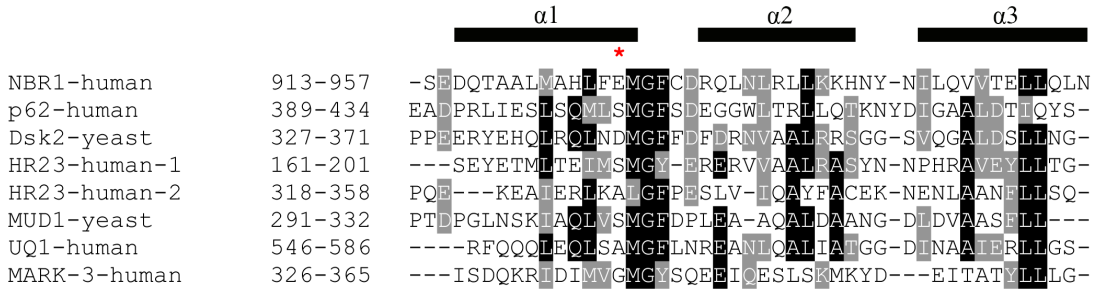

Figure 2.16: Sequence conservation in UBA domains. Identical residues are colored black, whereas homologous residues are colored gray. UBA domains at least partly contain an MGF motif and a di-leucine motif, which play an integral role in ubiquitin recognition. The red asterisk indicates a non-conserved residue with amino acid distribution shown in Fig. 2.28.

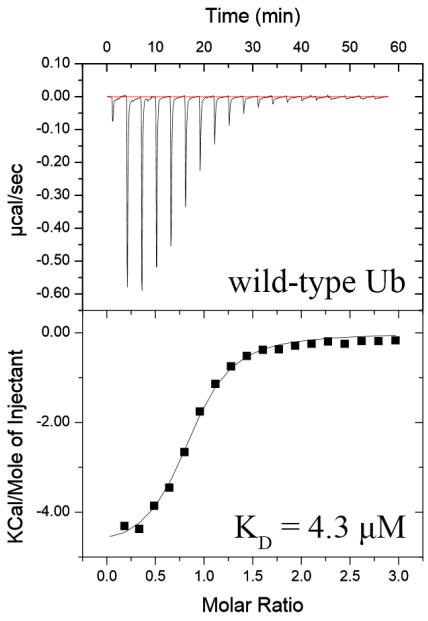

Figure 2.17: ITC thermogram for NBR1 UBA binding to monoubiquitin (Ub). The upper panel shows raw data, and the lower panels represents the integrated heat values.

	Ubiquitin	NBR1 UBA
Starting structure (PDB ID)	1D3Z	2MGW (this study)
Ambiguous restraints (CSP)	8	11
Orientational restraints (RDC)	27	29
Unambiguous restraints (NOE)	18	(18)

Table 2.4: Restraints used in the structure calculation of the UBA-ubiquitin complex by HADDOCK

HADDOCK statistics	
Clusters determined by HADDOCK	1
Structures in final cluster	199
RMSD from lowest-energy structure	0.8 ± 0.5
Van der Waals energy	-51.6 ± 2.5
Electrostatic energy	-222.2 ± 43.7
Desolvation energy	11.1 ± 6.7
Restraint violation energy	8.4 ± 12.9
Buried surface area	1144.6 ± 64.5

Table 2.5: Structure calculation of the UBA-ubiquitin complex by HADDOCK

2.4.5 Structure of the NBR1 UBA-Ubiquitin Complex

According to previous studies, both ubiquitin and UBA domains undergo only minimal structural changes on formation of the UBA-ubiquitin complex [46, 50]. For this reason, I chose the protein-protein docking approach of HADDOCK to obtain a structure of the NBR1 UBA-ubiquitin complex [42]. To drive the docking calculation, I used a combination of ambiguous restraints (obtained from chemical shift perturbation data), direct intermolecular distance restraints (obtained from a filtered NOESY experiment) and orientational restraints (obtained from residual dipolar couplings) as given in Table 2.4. The details of the docking calculation are given in Table 2.5. A representation of the resulting structural ensemble is shown in Fig. 2.18.

A close-up look on the structure shows that the main interface is composed of hydrophobic residues. On ubiquitin the side chains of Leu^{8}, Ile^{44}, His^{68} and Val^{70} appear to contribute most integrally to binding. The aliphatic side chains of Met^{927}, Ile^{946}, Leu^{947}, Val^{950}, Thr^{951} and Leu^{954} of the NBR1 UBA domain engage in hydrophobic interactions by sticking their aliphatic side chains into the hydrophobic surface on ubiquitin (Fig. 2.19).

Overall, the architecture of the NBR1 UBA-ubiquitin complex is reminiscent of the previously reported Dsk2 UBA-ubiquitin complex (Fig. 2.27 A). Indeed, the UBA helices appear to adopt a similar geometry and the complex is equally formed due to hydrophobic contacts.

Most hydrophobic interactions occur through residues that are evolutionarily

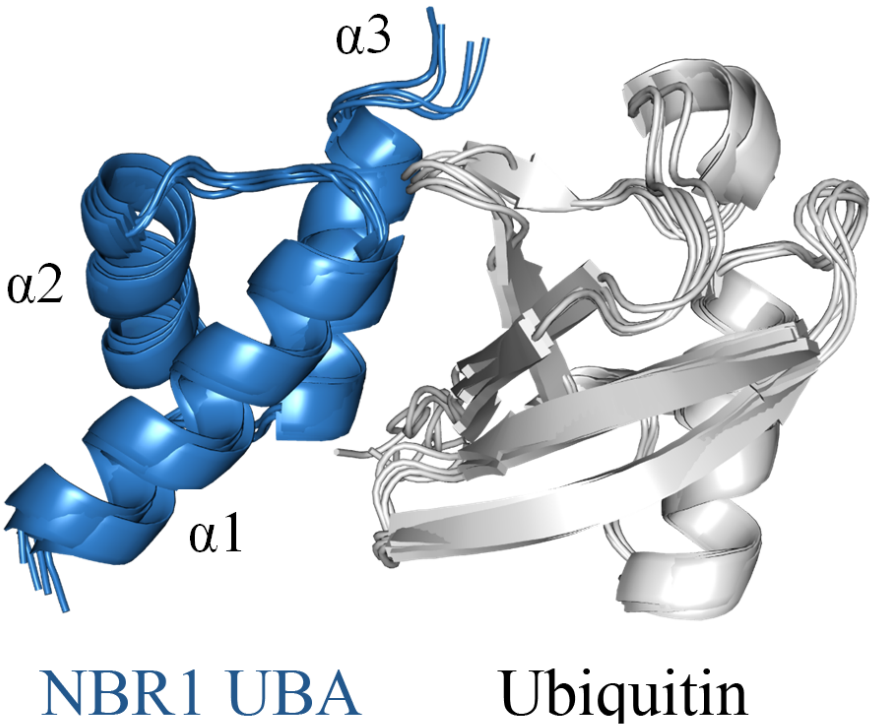

Figure 2.18: Overlay of the four lowest energy structures of NBR1 UBA in complex with ubiquitin as calculated from orientational and ambiguous and unambiguous distance restraints by HADDOCK.

conserved among UBA domains (Fig. 2.16). Therefore, I asked if any non-hydrophobic and possibly non-conserved residues may contribute to ubiquitin binding. Observation of the complex structure revealed the presence of polar contacts between the UBA domain and ubiquitin. The carboxyl group of Glu^{926} is in close proximity to both the side chain ϵ-amino group of Lys^{6}, as well as the imidazole ring of His^{68} on ubiquitin. This proximity permits the formation of both a salt bridge and a hydrogen bond, respectively (Fig. 2.27 B). Glu^{926} is not evolutionarily conserved in UBA domains; nevertheless, the UBA domain of Dsk2 has an aspartate residue at this position and this residue has indeed been shown before to contribute to ubiquitin binding via formation of similar polar contacts (Fig. 2.27 C, [50]). In stark contrast, other UBA domains, such as p62 UBA or HR23A UBA-1[12] have a serine and alanine residue, respectively, at this site (Fig. 2.16).

2.4.6 Validation of the UBA-Ubiquitin Surface by Site-directed Mutagenesis

Next, it was important to verify the structure of the UBA-ubiquitin complex to rule out the possibility of a wrong structure due to erroneous resonance assignment or

[12]This protein has two UBA domains, abbreviated as UBA-1 and UBA-2.

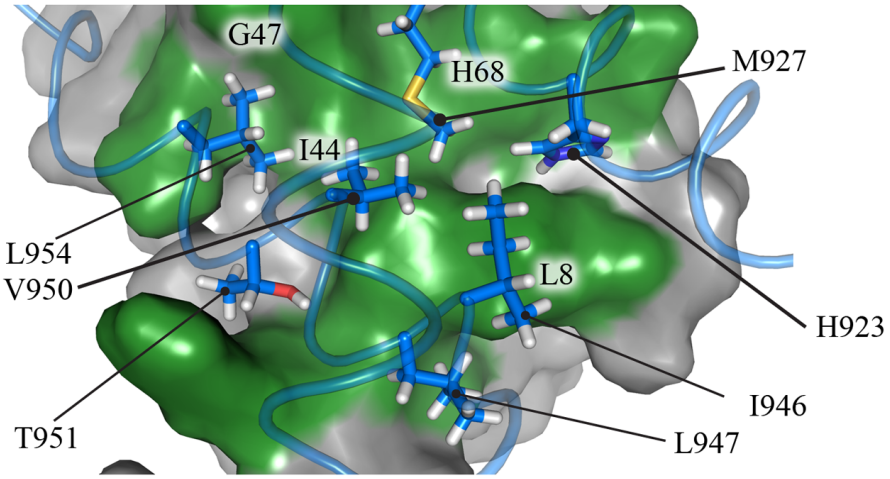

Figure 2.19: Close-up view of the UBA-ubiquitin interface. Surface residues of ubiquitin in contact with the UBA are colored green, and residues of NBR1 UBA involved in hydrophobic contacts with ubiquitin are shown as blue sticks.

misinterpretation of NOEs. For this reason, I employed site-directed mutagenesis. Using PCR, I generated four single point mutants of the NBR1 UBA domain (E926A, G928A, F929A and L954A), as well as three mutants of ubiquitin (I44A, K48A and V70A). Then, binding between wild-type and point mutant proteins was studied by ITC.

All interface point mutant proteins exhibited weaker binding as compared with the respective wild-type protein. For example, the ubiquitin V70A mutation resulted in an about 4-fold decreased binding affinity (Fig. 2.20). Moreover, replacement of Ile^{44} by alanine completely abolished binding (Table 2.6). In the binding of ubiquitin K48A to ubiquitin, a small amount of heat appeared to be released; however this binding was too weak to be quantitatively evaluated by the ITC experiment (Table 2.6).

In the case of the NBR1 UBA domain, replacement of Phe^{929} on the first inter-helix loop by alanine, as well as alanine mutation of Leu^{954} on helix α-3 entirely abolished binding (Fig. 2.21, Table 2.6). Importantly, substitution of the non-conserved residue Glu^{926} for alanine reduces the affinity of NBR1 UBA for ubiquitin about 3-fold, which highlights the important contribution of the electrostatic contacts observed in the NBR1 UBA-ubiquitin complex (Fig. 2.27 B, Table 2.6). While the G928A UBA mutant still bound to ubiquitin, this association was about 2-fold weaker than that of the wild-type protein (Table 2.6). Taken together, these mutation experiments confirm the UBA-ubiquitin complex interface determined by NMR spectroscopy.

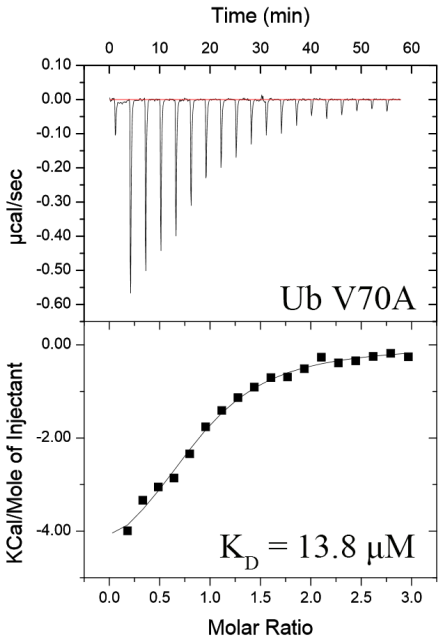

Figure 2.20: ITC thermogram for NBR1 UBA binding to the ubiquitin point mutant V70A. The upper panel shows raw data, and the lower panels represents the integrated heat values.

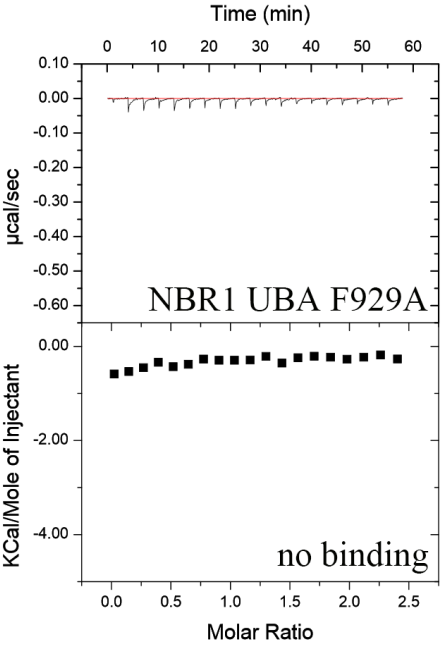

Figure 2.21: ITC thermogram for ubiquitin binding to the NBR1 UBA point mutant F929A. The upper panel shows raw data, and the lower panels represents the integrated heat values.

	N	K_d [μM]	ΔH [kcal mol^{-1}]
NBR1 UBA binding to ubiquitin			
Wild-type	0.78 ± 0.04	4.3 ± 0.4	-4.94 ± 0.16
I44A		no binding	
K48A		no binding	
V70A	0.92 ± 0.07	13.8 ± 3.8	-4.59 ± 0.67
Ubiquitin binding to NBR1 UBA			
Wild-type	0.78 ± 0.04	4.3 ± 0.4	-4.94 ± 0.16
E926A	0.94 ± 0.05	13.6 ± 3.3	-4.30 ± 0.29
G928A	0.62 ± 0.04	10.8 ± 3.4	-7.12 ± 0.73
F929A		no binding	
L954A		no binding	

Table 2.6: Thermodynamic parameters for binding of wild-type and mutant proteins. N, stoichiometry; K_d, dissociation constant; ΔH, enthalpy change. Errors represent the standard deviation from three independent experiments.

2.4.7 NBR1 UBA Does Not Discriminate between Mono- and Polyubiquitin

Up until this point, I merely considered the interaction of the UBA domain of NBR1 with monomeric ubiquitin, i.e. monoubiquitin. However, in living cells and organisms, a large fraction of ubiquitin molecules exists as polymeric ubiquitin chains, i.e. polyubiquitin (section 1.5) [51]. In polyubiquitin chains adjacent ubiquitin units are linked via isopeptide bonds between one of seven amino groups or the N-terminus of a given ubiquitin molecule and the C-terminal carboxyl group of another ubiquitin molecule. The way these chains are formed (for example, a linkage at residue Lys^{48} vs. Lys^{63}) is believed to serve as a specific signal, which can be read by ubiquitin binding domain (UBD) containing proteins. For example, it is well established that a polyubiquitin chain liked through the amino acid Lys^{48} specifically targets attached proteins for proteasomal degradation. However, it is still an open question, if such a linkage-specific polyubiquitin signal is recognized in autophagy, i.e. whether or the "ubiquitin code" contains a signal for autophagy (section 1.6).

To examine, whether or not NBR1 UBA shows a specific polyubiquitin linkage preference, I used NMR titration experiments to investigate the interaction of the UBA domain with the two types of polyubiquitin that are of highest cellular abundance: Lys^{48}- and Lys^{63}-linked polyubiquitin chains [52]. I prepared ubiquitin dimers of both linkage types and conducted titration experiments with the UBA as analyte and the ubiquitin dimers as titrant. These experiments with both Lys^{48}- and Lys^{63}-linked diubiquitin yielded chemical shift changes of NBR1 UBA that were identical to those observed in the titration experiment with monoubiquitin (Fig. 2.22, 2.23). In fine accord, the HSQC spectra of NBR1 UBA at the end of the

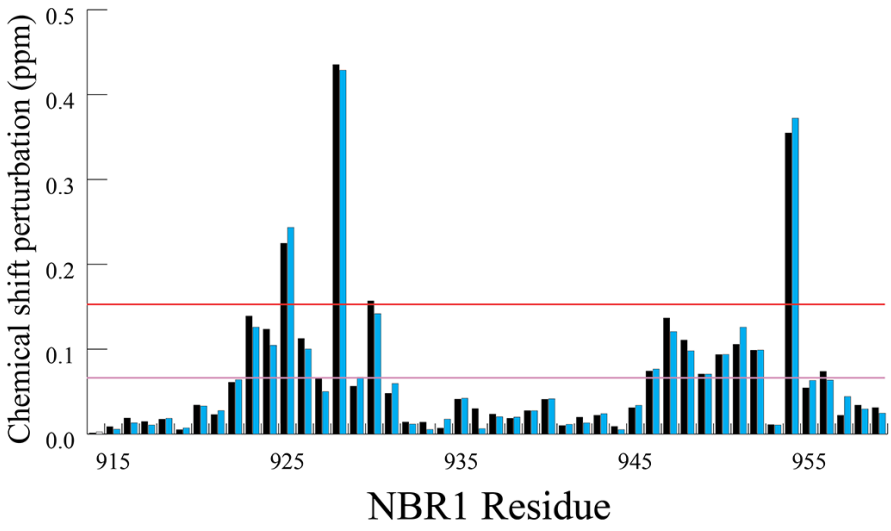

Figure 2.22: CSP values for binding of NBR1 UBA to Lys^{48}- linked diubiquitin (blue) and monoubiquitin (black). The horizontal lines are colored as follows: magenta, $\delta_{av} \leq$ chemical shift perturbation (CSP) $< \delta_{av} + 1\ \sigma$; red, CSP $\geq \delta_{av} + 1\ \sigma$.

titration displayed essentially identical chemical shifts, no matter if monoubiquitin, Lys^{48}- or Lys^{63}-linked diubiquitin was used as titrant. These data imply that NBR1 UBA uses the very same surface to bind to both types of polyubiquitin as it does to bind to monoubiquitin. Accordingly, NBR1 UBA appears to lack a specific epitope to selectively recognize Lys^{48}- or Lys^{63}-linked polyubiquitin chains.

Next, I derived the dissociation constants from the titration data as described above for the binding to monoubiquitin. These NMR spectroscopy-derived affinity values indicated that NBR1 UBA binds to Lys^{63}-linked diubiquitin essentially as strongly as to monoubiquitin ($K_d = 4.5 \pm 1.2\ \mu$M; Fig. 2.24 B). On the other hand, the Lys^{48}-linked diubiquitin was bound less tightly ($K_d = 19.0 \pm 2.0\ \mu$M; Fig. 2.24 C). The lower affinity binding of the Lys^{48}-linked dimer may be explained by the lower accessibility of the proximal ubiquitin unit of the dimer. Since this type of dimer uses Lys^{48} for isopeptide linkage [8], binding of NBR1 UBA to the close-by hydrophobic patch around Ile^{44} may be slightly weakened.

I also used ITC to assess the binding of NBR1 UBA to monoubiquitin and these two types of diubiquitin. As in the case of monoubiquitin, ITC-derived dissociation constants were in good agreement with the values obtained from NMR spectroscopy (Fig. 2.25, 2.26 and Table 2.7). Indeed, a similar amount of heat was released in the binding reactions of NBR1 UBA to both types of ubiquitin dimers and to monoubiquitin. In fine accord, the stoichiometries obtained from ITC experiments further supported that a given NBR1 UBA molecule simply binds to one ubiquitin molecule of the respective ubiquitin dimer regardless of the polymeric state or linkage

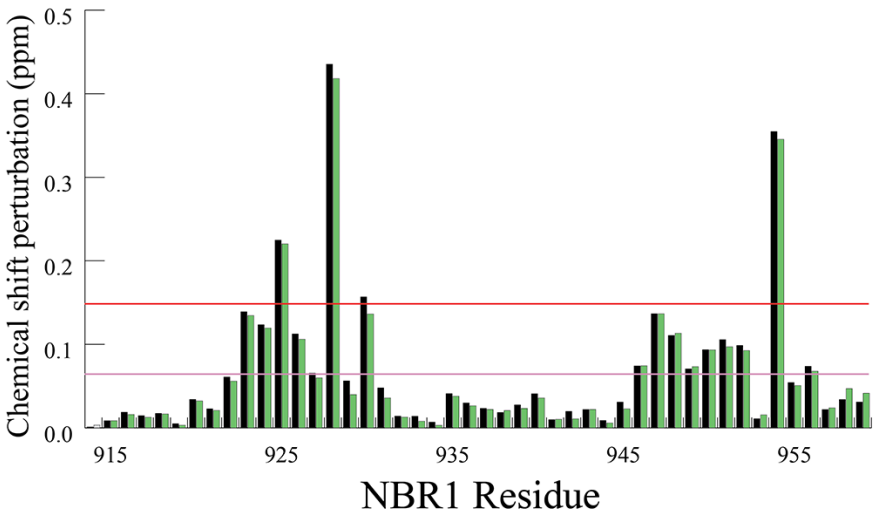

Figure 2.23: Normalized CSP values of NBR UBA on binding to Lys^{63}-linked diubiquitin (green) as compared with monoubiquitin (black). The horizontal lines are colored as follows: magenta, $\delta_{\mathrm{av}} \leq$ chemical shift perturbation (CSP) $< \delta_{\mathrm{av}} + 1\ \sigma$; red, CSP $\geq \delta_{\mathrm{av}} + 1\ \sigma$.

type[13] . Collectively, these results suggest that the UBA domain of NBR1 recognizes ubiquitin in a chain-linkage nonspecific manner, thereby binding to ubiquitin and polyubiquitin equally strongly.

2.5 Discussion

2.5.1 UBA Structure and Ubiquitin Binding

The UBA domain of human NBR1 forms a three-α-helical structure that resembles previously reported structures of UBA domains such as Dsk2 or p62. It shows a weak tendency to dimerize in solution ($K_{\mathrm{d}} \approx 400\ \mu$M). This value may indicate that the association is too weak to play a role in physiological processes. However, it should be noted that in certain cases even such weak interactions have been proven to be physiologically relevant [53].

NBR1 UBA interacts with ubiquitin primarily by evolutionarily conserved hydrophobic interactions. Nevertheless, in this study important polar contacts between the UBA and ubiquitin could be identified.

To date, all known UBA domains display a three α-helical structure and at

[13]The fact that the stoichiometry values are not perfectly 1.0 is due to difficulties in correctly assessing the protein concentrations. Both the NBR1 UBA domain and ubiquitin contain no tryptophan residue; therefore both proteins have a comparably low extinction coefficient. Estimation of protein concentrations from such proteins can cause an error of ± 10 %. This error in protein concentration directly translates into an uncertainty in the stoichiometry. Nevertheless, all types of ubiquitin show the same tendency and thus a stoichiometry of N = 1 is implied.

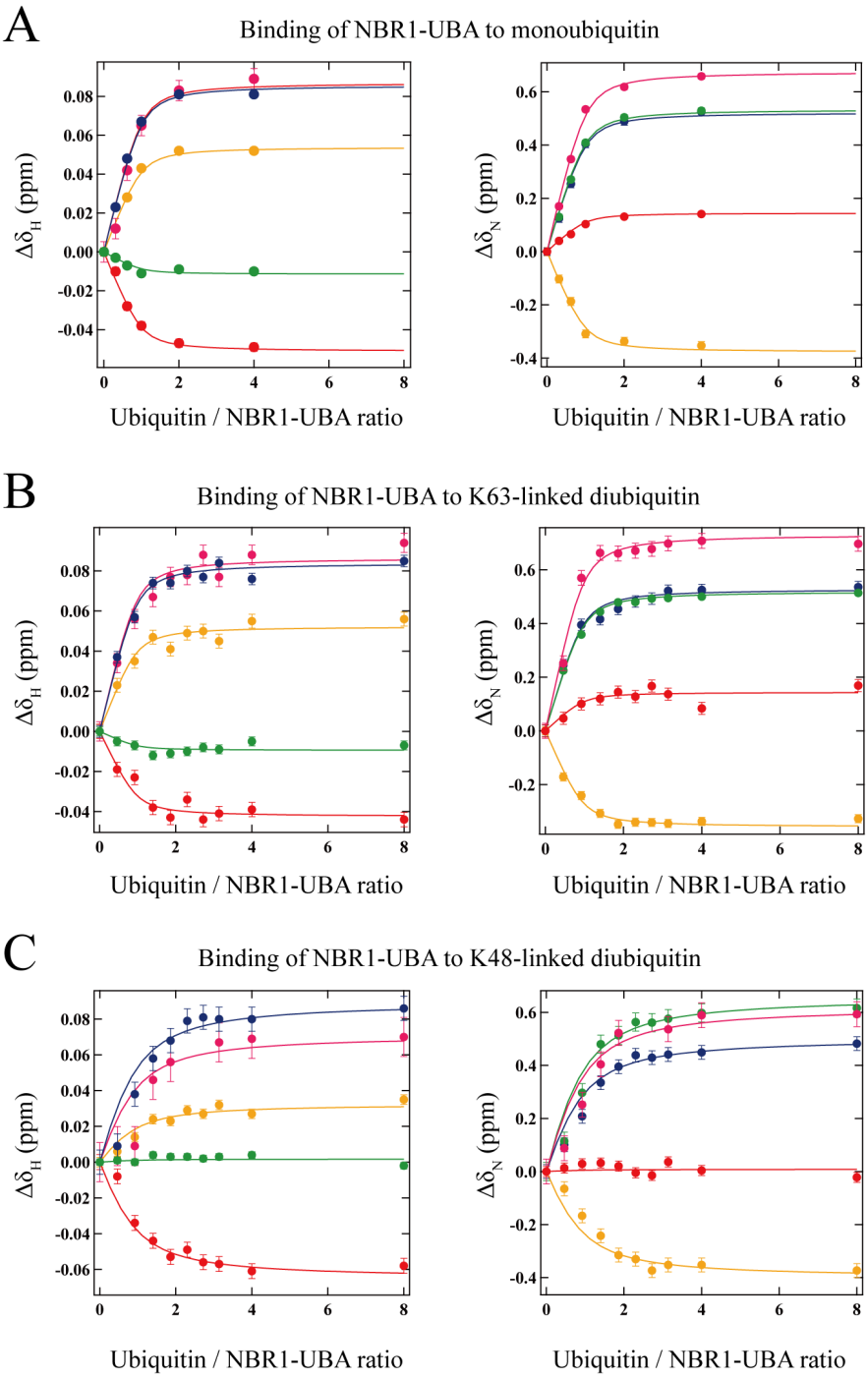

Figure 2.24: Binding of NBR1 UBA to ubiquitin and diubiquitin as determined by NMR. ^{1}H- and ^{15}N-chemical shift changes are plotted (filled circles) against the molar ratio of ubiquitin to NBR UBA and separately fit (solid lines) to eq. 2.2. A, binding of NBR1 UBA to monoubiquitin; B, binding of NBR1 UBA to Lys^{63}-linked diubiquitin; C, binding of NBR1-UBA to Lys^{48}-linked diubiquitin; In the case of diubiquitin, the data were fit assuming that one UBA domain binds to one ubiquitin unit of diubiquitin in a 1:1 stoichiometry. For comparison, data in the range of 0–8 mol eq are shown. Although for clarity only five example peaks are shown, the derived global K_{d} values are the result of fitting the chemical shift difference data of all amide resonances. Yellow, Met^{927}; red, Phe^{929}; magenta, Gln^{948}; blue, Glu^{952}; green, Gln^{955}.

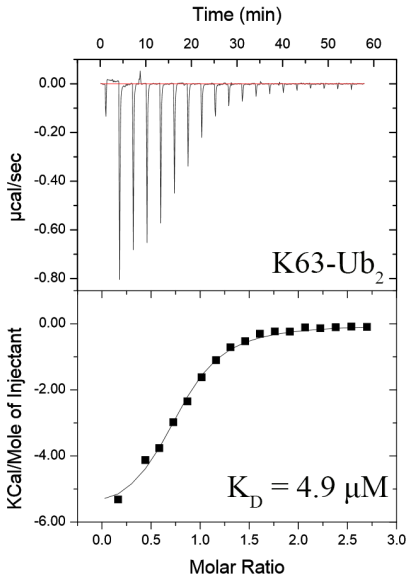

Figure 2.25: ITC thermogram for NBR1 UBA binding to Lys^{63}-linked diubiquitin, indicating an affinity similar to that for binding to monoubiquitin.

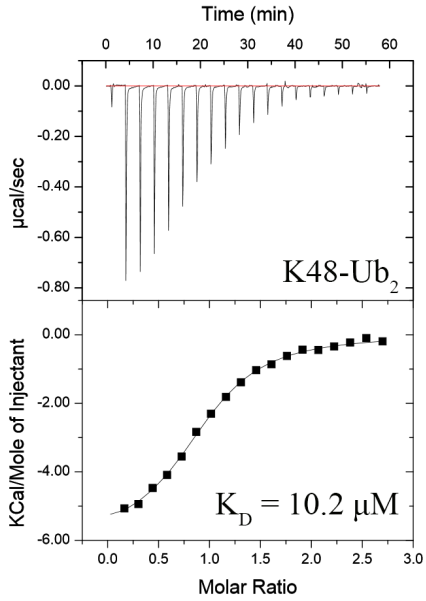

Figure 2.26: ITC thermogram for NBR1 UBA binding to Lys^{48}-linked diubiquitin, indicating a slightly weaker binding as compared with Lys^{63}-linked diubiquitin or monoubiquitin.

	N	K_d [μM]	ΔH [kcal mol^{-1}]
ITC			
Ubiquitin	0.78 ± 0.04	4.3 ± 0.4	-4.94 ± 0.16
K63-Ub_2	0.83 ± 0.09	4.9 ± 2.0	-5.36 ± 0.79
K48-Ub_2	0.90 ± 0.11	10.2 ± 1.1	-5.77 ± 0.58
NMR			
Ubiquitin		3.0 ± 0.9	
K63-Ub_2		4.5 ± 1.2	
K48-Ub_2		19.0 ± 2.0	

Table 2.7: Thermodynamic parameters for binding of NBR1 UBA to ubiquitin and diubiquitin. In the case of diubiquitin the concentration of the ligand was defined as the concentration of monomeric ubiquitin units in the dimer. Thus, a stoichiometry value of 1 indicates that one UBA domain binds to one ubiquitin unit of the respective ubiquitin dimer. Errors represent the standard deviation from three independent experiments. For comparison, the dissociation constants obtained from NMR spectroscopy (Fig. 2.24) are also shown.

least partly contain the conserved MGF and di-leucine motif. Despite this high degree of sequence conservation, the affinity of UBA domains for ubiquitin is highly variable. Generally speaking, most UBA domains have been reported to bind to monoubiquitin with a dissociation constant of the order of tens to a few hundred of micromolar. An example of a particularly weak interaction is the MARK3 UBA domain which binds to ubiquitin with a K_d of $\approx$ 2 mM [54]. Thus, among UBA domains, NBR1 UBA is a comparatively strong binder of ubiquitin ($K_d \approx 3$ μM). The affinity of NBR1 UBA for ubiquitin is particularly puzzling in the context of selective autophagy, since it is more than 2 orders of magnitude higher than the affinity of the UBA domain of autophagy receptor p62 for ubiquitin. Indeed, the low in-vitro affinity of p62 UBA for ubiquitin as determined from NMR and ITC experiments has puzzled researchers in the field, since it was not clear, how p62 could act as a ubiquitin-selective autophagy receptor given that is dissociation constant for the binding to ubiquitin is as low as 540-750 μM.

2.5.2 Possible Regulation of Ubiquitin Affinity by Phosphorylation of UBA Domains

It has been proposed that the low affinity of p62 UBA for ubiquitin stems from the strong tendency of p62 UBA to form a homodimer. This homodimer formation in turn inhibits ubiquitin binding resulting in a lower affinity. Nevertheless, it has also been reported that even specifically engineered *monomeric* point mutants of p62 UBA bind to ubiquitin with an affinity about an order of magnitude more weakly as compared with NBR1 UBA [25]. Thus, the affinity difference of both

autophagy receptor UBA domains may hint at another mechanism, in addition to the self-inhibition of p62 UBA.

For this reason, I considered the possibility of post-translational modifications to alter the ubiquitin binding affinity of the p62 UBA domain. Indeed, it has been recently reported that p62 is subject to a phosphorylation event at residue Ser^{403} on helix α-1 of its UBA domain [55]. This serine residue is located just in front of the evolutionarily conserved MGF motif. Intriguingly, phosphorylation at Ser^{403} has been shown to significantly enhance (poly-)ubiquitin binding by p62 *in vivo* [55]. To date, no dissociation constant for the binding of Ser^{403}-phosphorylated p62 UBA to ubiquitin has been reported. Nevertheless, a phosphomimetic[14] mutant of p62 (S403E) displayed a markedly increased binding to polyubiquitin in pulldown assays. This fact suggests that the binding of p62 to ubiquitin may be attenuated until the kinase CK2 [55] introduces a double negative charge in the form of a phosphate group into the ubiquitin-binding surface of p62 UBA at Ser^{403}.

The necessity of the addition of a double negative charge into the p62 UBA-ubiquitin binding interface to achieve ubiquitin binding *in vivo* was intriguing, since the corresponding residue in NBR1 is also negatively charged. As the structure of the NBR1 UBA-ubiquitin complex shows (Fig. 2.27 B), this Glu^{926} uses its negatively charged carboxylate group to engage in polar interactions with the side chains of both Lys^{6} and His^{68} on ubiquitin. These interactions contribute to the affinity of NBR1 for ubiquitin (Table 2.6, Fig. 2.27 B). Moreover, in the UBA domain of Dsk2, the residue at the corresponding position is also a negatively charged residue, Asp^{341}, which forms similar interactions (Fig. 2.27 C). Although the interaction of p62 UBA and ubiquitin is comparatively weak, published chemical shift perturbation data indicate that p62 recognizes ubiquitin via a surface similar to that of NBR1 and Dsk2 [25]. Thus, it seems reasonable that phosphorylation of p62 at residue Ser^{403} of its UBA domain raises its ubiquitin affinity by unlocking the formation of similar polar contacts between the phosphate group and Lys^{6} and His^{68} on ubiquitin (Fig. 2.27 D, E).

A comparison of the side chain charge of the amino acid in front of the evolutionarily conserved MGF motif and the ubiquitin binding affinity of the respective UBA domain is shown in Table 2.8. The UBA domain of Dsk2 contains a negatively charged (aspartate) residue at this position and is like NBR1 UBA also a comparatively strong binder of ubiquitin (K_d = 15 μM). In contrast, the weak ubiquitin binder HR23A UBA-1 (K_d = 540 μM) has an (non-charged) alanine at the corresponding position. Therefore, it seems likely that the charge of the amino acid side chain in front of the evolutionarily conserved MGF motif plays a critical auxiliary role in the interaction with ubiquitin by modulating ubiquitin affinity. Interestingly,

[14] Phosphorylation-mimicking

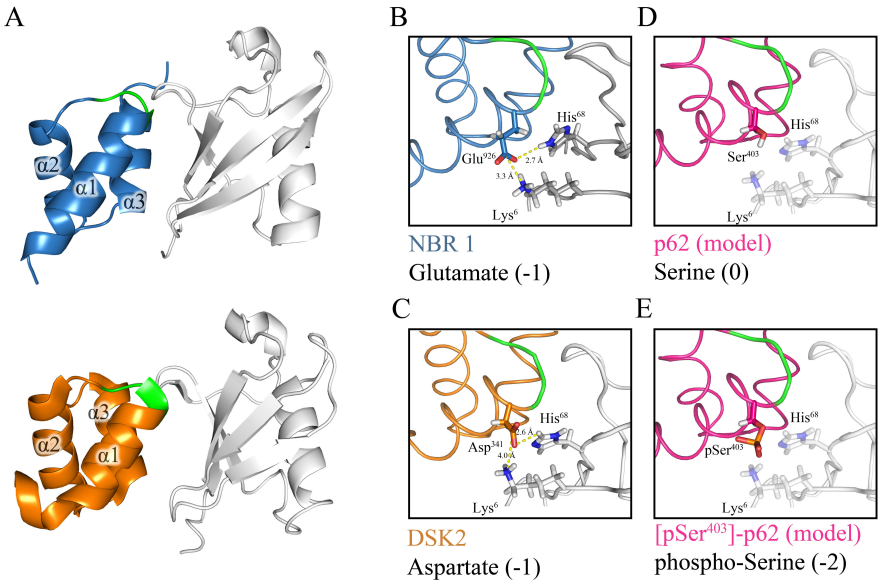

Figure 2.27: Fine-tuning of ubiquitin affinity among UBA domains. A, structural comparison of the ubiquitin complexes of NBR1 (blue, top) and Dsk2 (orange, bottom; PDB ID 1WR1) UBA domains. The conserved MGF motif is shown in green, and ubiquitin is shown in gray. The UBA helices adopt a similar geometry in both proteins. B–E, the UBA domains of NBR1 (B), Dsk2 (C), and p62 (D and E) are shown in the same orientation with the conserved MGF motif colored in green. The side chain of the residue in front of the MGF motif is displayed with its charge at physiological pH given in parentheses. B, in the NBR1 UBA-ubiquitin complex, the negative charge of Glu^{926} permits electrostatic interactions with ubiquitin. C, similar interactions are formed by Asp^{341} of Dsk2 UBA. D, model of the p62 UBA-ubiquitin complex based on the reported Dsk2 UBA-ubiquitin complex. Although p62 UBA does not possess a charged residue at this position, phosphorylation of Ser^{403} introduces a double negative charge into the p62 UBA-ubiquitin interface (E, schematic; drawn from PDB ID 2RRU). This phosphorylation event has been reported to significantly promote ubiquitin binding *in vivo*, presumably by the formation of electrostatic interactions similar to those in NBR1 (B) and Dsk2 (C).

UBA domain	Pre-MGF charge	K_D [μM]	Reference
NBR1	-1 (E)	4	this study
Dsk2	-1 (D)	15	[50]
p62	0 (S)	540-750	[25]
p62 pSer^{403}	-2 (pSer)	(significantly enhanced *in vivo*)	[55]
p62 S403E	-1 (E)	(significantly enhanced *in vitro*)	[55]
HR23A UBA-1	0 (A)	540	[49]

Table 2.8: Comparison of ubiquitin binding affinities (given in μM) of several UBA domains reported in the literature with the UBA domain of NBR1. The amino acid in front of the conserved MGF motif and its charge at physiological pH is given.

in the case of p62, the charge can be reversibly adjusted by phosphorylation and dephosphorylation according to the given physiological context.

Although the amino acid in front of the MGF motif in UBA domains appears to be evolutionarily non-conserved as judged from sequence alignment of to date studied UBA domains (Fig. 2.16), the above mentioned considerations implied some regularity in the amino acid distribution at this position. For this reason, I used the SMART database[15], in which the amino acid sequence of several hundred UBA domains[16] are listed. Plotting the amino acid distribution of the residue in front of the conserved MGF motif revealed that there is a clear preference for a negatively charged amino acid (42 %) (Fig. 2.28, red). Since in all cases this residue is in close proximity to the MGF motif that has been shown to be integral to ubiquitin binding in all cases studied so far, it is likely that the negative charge of the preceding residue may contribute to ubiquitin binding of the respective UBA domain in a similar way as observed in NBR1 (Fig. 2.27). In stark contrast, positively charged amino acids which would be expected to repel the ubiquitin side chains Lys^6 and His^{68} are virtually absent from this site (Fig. 2.28, blue). Most intriguingly, of the 1725 UBA domains checked in the SMART database, 300 (17%) UBA domains possess a serine residue at this site. This suggests the possibility that these UBA domains may be subject to regulation by phosphorylation in a fashion analogous to p62.

2.5.3 Binding of NBR1 UBA to Lys^{48}- and Lys^{63}-linked Polyubiquitin Chains

In the past, binding of UBA domains to ubiquitin and polyubiquitin has often been assessed by pull-down assays in a non-quantitative way or by surface plasmon resonance in a quantitative manner. In both cases, not the UBA domain itself, but a heterologous fusion protein of Glutathione S-transferase (GST) and the UBA domain (GST-UBA) is assessed. A recent study reported that this methodology

[15]SMART database: http://smart.embl-heidelberg.de

[16]This includes UBA domains from various species, as well as putative UBA domains.

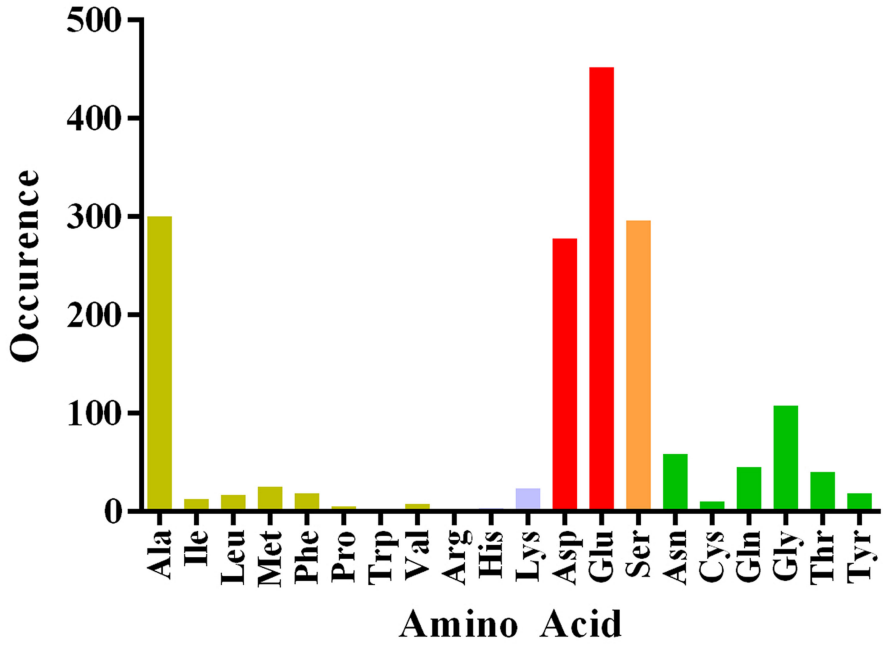

Figure 2.28: Amino acid distribution of the residue in front of the conserved MGF motif in eukaryotic UBA domains. There is a clear preference (42 %) for negatively charged residues. Moreover, 17 % of UBA domains possess a serine residue at this site that might be subject to phosphorylation as in p62. Amino acids are colored as follows: *yellow*, hydrophobic; *blue*, positively charged; *red*, negatively charged; *green*, neutral charge; *orange*, serine.

	K_d [μM]	
	UBA (this study)	GST-UBA (ref. [27])
Ubiquitin	3	119
K63-Ub_2	4	1.7
K48-Ub_2	15	2.4

Table 2.9: Comparison of the approximate dissociation constants obtained in this study (the isolated UBA) and a previous work using GST-fused NBR1 UBA. Mean values from ITC and NMR experiments are rounded for comparison.

introduces artifacts in the dissociation constants and binding stoichiometries, most likely due to the fact that the fusion partner GST forms a tight dimer [47]. In fact, different dissociation constants and thus different ubiquitin-linkage specifities were obtained from experiments with GST-fused and GST-free samples of UBA domains [47].

In the case of NBR1, a pioneering study that first identified NBR1 as an autophagy receptor reported that a GST-fused NBR1 UBA domain binds more strongly to Lys^{63}-linked diubiquitin than to monoubiquitin [27]. This observed 60-fold stronger binding in conjunction with similar experiments on GST-fused p62 UBA led to the hypothesis that Lys^{63}-linked polyubiquitin may act as a specific signal for selective autophagy in the sense of the ubiquitin code [56] (section 1.6).

Due to the aforementioned artifacts arising from using GST-fused UBA domains in binding assays, I deliberately chose to study the association of NBR1 UBA and (poly)ubiquitin by experiments in the solution state, namely ITC and NMR, to eliminate the effect of GST-dimerization. Dissociation constants obtained from these experiments markedly differed from those previously reported for the GST-fused UBA domain of NBR1 [27] (Table 2.9). Importantly, while the previous study concluded that NBR1 UBA shows a slight preference for Lys^{63}-linked polyubiquitin, no such preference for either dimer was observed in this study. In addition to the derived dissociation constants, ITC-derived parameters support that a NBR1 UBA molecule binds to one *unit* of diubiquitin in a 1:1 stoichiometry (Table 2.7). NMR titration experiments further supported this claim showing that the UBA domain of NBR1 binds to both types of diubiquitin by the same surface as used in the binding of monoubiquitin. Taken together, these results strongly argue that there is no specificity for Lys^{48}- or Lys^{63}-linked polyubiquitin in the NBR1 UBA domain.

2.5.4 Possibility of the Specificity of NBR1 UBA for Other Ubiquitin Linkages

In my experiments, I used merely monoubiquitin and Lys^{48} and Lys^{63}- linked diubiquitin. So far none of the other ubiquitin linkages have been considered. However,

Lys^{6}-, Lys^{27}-, Lys^{29}, Lys^{33}- account for ≤ 1 % of the total amount of polyubiquitin in cells [52]. From structural studies, it is well established that the linear (Met^{1}-linked) ubiquitin chain adopts an open and extended conformation which closely resembles Lys^{63}- linked diubiquitin [57]. Thus, it is equally unlikely that the small and compact NBR1 UBA domain could selectively recognize linear diubiquitin. However, the possibility that NBR1 UBA may show a higher affinity for the rather compact Lys^{11}- ubiquitin chain cannot be ruled out [58].

It should be stressed that I have been working on the isolated UBA domain of human NBR1. While this domain appears to have no polyubiquitin linkage specificity on its own, such a specificity may still arise in the context of full-length NBR1. In particular, since NBR1 is reported to bind to p62 via its PB1 domain (Fig. 2.1) and has been hypothesized to dimerize via a coiled-coiled region [22, 47], such events may bring two or more C-terminal UBA domains in spatial proximity. By such a mechanism, a particular ubiquitin chain linkage might be recognized with high specificity.

2.5.5 No Linkage Specificity and Ubiquitin-positive Inclusion Bodies

On the other hand, it is quite intriguing that the UBA domain of NBR1 does *not* show a specificity towards a particular ubiquitin-linkage if we consider recent findings in the field of autophagy cell biology. These studies surprisingly highlighted that polyubiquitin chains that are found in ubiquitin-positive inclusion bodies do not show a particular linkage specificity either [52]. Such ubiquitin-positive inclusions are generally degraded by autophagy [59]. Thus, it is very reasonable that an autophagy receptor protein such as NBR1 displays *no* polyubiquitin-linkage selectivity.

2.5.6 Conclusion

This study represents the first structural and biophysical characterization of the UBA domain of human NBR1 as well as its interaction with ubiquitin and polyubiquitin. Structural analysis revealed key differences between the autophagy receptors NBR1 and p62, explaining the significantly higher affinity of the NBR1 UBA domain for ubiquitin. Interestingly, NBR1 was identified to possess a non-selective UBA domain which is in fine accord with recent reports that ubiquitin-positive aggregates, which are substrates of autophagy, show no polyubiquitin-linkage specificity either.

Chapter 3

Dynamics of Autophagy Receptor p62 Studied by Automated $R_{1\rho}$-Relaxation Dispersion

3.1 Abstract

p62 was the first protein identified as a receptor for ubiquitylated proteins in selective autophagy. In analogy to NBR1 (chapter 2), p62 is thought to use its ubiquitin-associated (UBA) domain to bind to ubiquitin, which marks autophagic cargo for degradation. However, in contrast to NBR1, which simply binds to ubiquitin, p62 shows a more complex binding mode: the UBA domain of p62 also shows a strong tendency to bind to itself (homo-dimerization). Accordingly, the crystal structure of the UBA domain of p62 was reported as a homodimer. To attain a better understanding of this complex system, it is necessary to characterize the conformational dynamics of the p62 UBA domain. Accordingly, I employed $R_{1\rho}$-relaxation dispersion NMR spectroscopy to investigate the UBA domain of p62. Implementation of this measurement spawned development of an efficient processing methodology, which I named "Amaterasu". As a result, $R_{1\rho}$-relaxation dispersion could be successfully detected in p62 UBA: two distinct kinds of motion on the millisecond timescale were observed. Residues exhibiting $R_{1\rho}$-relaxation dispersion consist of residues at the UBA:UBA interface of the homodimer and residues that participate in ubiquitin binding. I anticipate that the methodology introduced in this section will facilitate the study of biologically relevant motion in many proteins. Indeed, Amaterasu is already being applied in the study of the dynamics of three proteins in addition to p62.

3.2 Introduction

3.2.1 Autophagy Receptor p62

The protein p62[1], was first described as a receptor protein in selective autophagy by *in vivo* studies, which demonstrated that p62 colocalizes with ubiquitylated protein aggregates in neurodegenerative diseases [60, 61]. This protein is believed to link ubiquitylated proteins to the autophagy system to achieve their degradation by the lysosome (Fig. 1.6), in a manner analagous to NBR1 (chapter 2).

However, as mentioned already in chapter 2, it is still somewhat elusive, how p62 actually does achieve binding to autophagic substrates, that is to ubiquitylated proteins. Whereas it was expected that p62 uses its ubiquitin-associated (UBA) domain (section 1.7) to bind to ubiquitin on autophagic substrates, *in vitro* experiments showed that the affinity of the UBA domain of p62 for ubiquitin was unexpectedly low ($K_{\mathrm{d}} \approx 500\ \mu$M). For comparison, the UBA domain of autophagy receptor NBR1 binds ubiquitin with a $K_{\mathrm{d}} \approx 4\ \mu$M (Fig. 2.17). Because NMR can detect even weak interactions, it was possible to establish that p62 uses the evolutionarily conserved MGF motif to interact with ubiquitin in a manner similar to Dsk2 and NBR1 [25, 50]. However, in contrast to both Dsk2 and NBR1, the UBA domain of p62 was found to form a stable homodimer [25]. To date, this intriguing property among UBA domains has only been described for the case of p62.

Interestingly, the dimerization interface of p62 UBA does apparently not completely overlap with the predicted ubiquitin binding site. For example, the conserved MGF motif that plays the most integral role in ubiquitin binding, is partially exposed to solvent, even in the dimeric state. Therefore, it appears that *ubiquitin could bind* much more strongly to p62 UBA, perhaps on a similar order as NBR1 and Dsk2 (Fig. 3.1). Disruption of the p62 UBA dimer by site-directed mutagenesis converts the UBA domain to a monomeric molecule. In the monomeric form, p62 UBA shows a markedly enhanced ($K_{\mathrm{d}} \approx$ 20-30 μM, [25]) affinity toward ubiquitin. This increase in ubiquitin binding affinity cannot be exclusively explained by the structure of the UBA domain since, it appears that ubiquitin could bind to the dimeric form as well as to the monomeric form. Therefore, the goal of this study was to establish a dynamical view of the p62 UBA domain in addition to the static view represented by the p62 UBA crystal structure to attain a better understanding of the role of the dimerization of p62 and its effect on ubiquitin binding.

[1] Also called sequestosome 1 (SQSTM1)

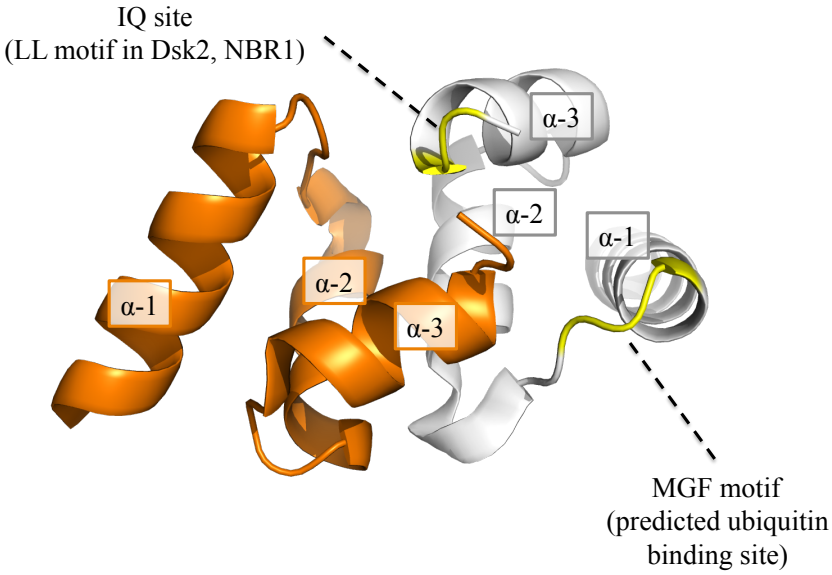

Figure 3.1: Crystal structure of p62 UBA drawn from PDB ID 3B0F (ref. [25]). The two monomeric parts of the UBA dimer are shown in orange and white. The α-helices of the canonical three-helix UBA fold are indicated. The evolutionarily conserved MGF motif, which is predicted to function in ubiquitin binding, is highlighted in yellow. The "LL" sequence, which is another conserved motif in UBA domains, is only partially similar in p62: residues "IQ" at the C-terminus of the molecule, yellow. (Fig. 2.16).

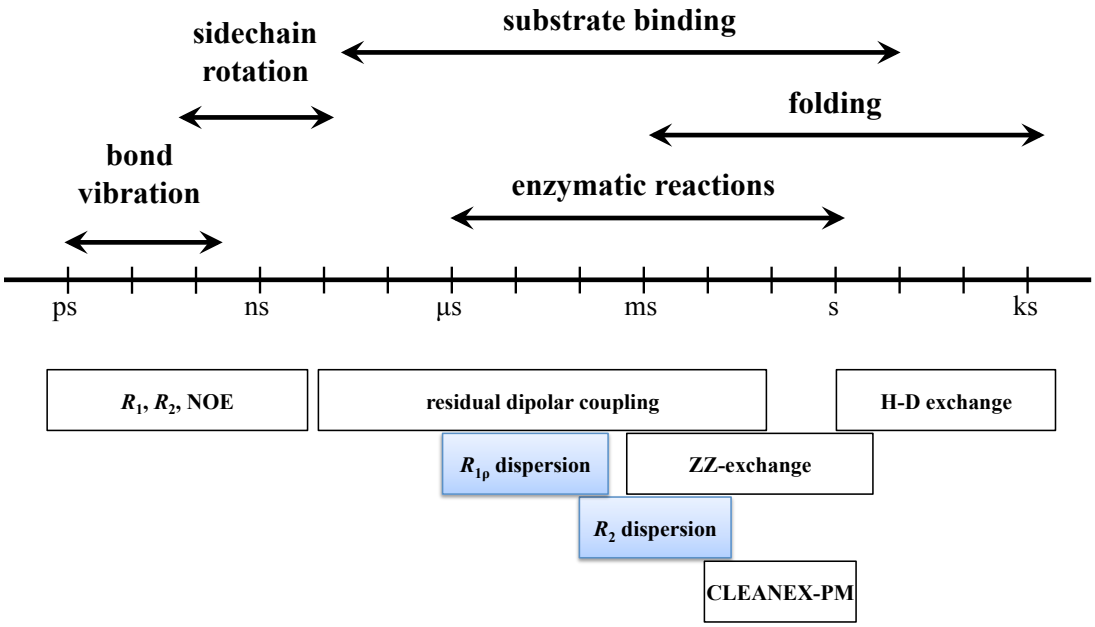

Figure 3.2: Timescales accessible to study protein motion by NMR spectroscopy.

3.2.2 Relaxation Dispersion

NMR spectroscopy is uniquely suited to measure protein dynamics on a variety of timescales. Picosecond to nanosecond dynamics are detected by measurement of R_1 and R_2 relaxation constants and heteronuclear NOE experiments [62]. Examples of motion on this timescale include bond vibrations and sidechain rotations. Residual dipolar couplings can detect a wide range of timescales from sub-microsecond to milliseconds. Slower dynamics on the order of seconds to days can be assessed by hydrogen-deuterium (H-D) exchange (Fig. 3.2).

Relaxation dispersion (RD) is a particularly useful NMR technique for detecting motion on a *biological relevant timescale* in proteins such as the conformational rearrangements encompassing substrate binding and enzymatic reactions. Such motion occurs on the microsecond to millisecond timescale. In particular, the existence of minor conformers (excited states; population $\approx 1\%$) in a protein ensemble can be detected by relaxation dispersion, whereas a usual NMR spectrum would merely report the ensemble *average* in the solution. This average is close to the major conformer (population $\approx 99\%$), that is the "ground state" of the protein. From RD-NMR measurements structural information (chemical shift), kinetic (lifetime) and thermodynamic information (population) about this "invisible" minor state can

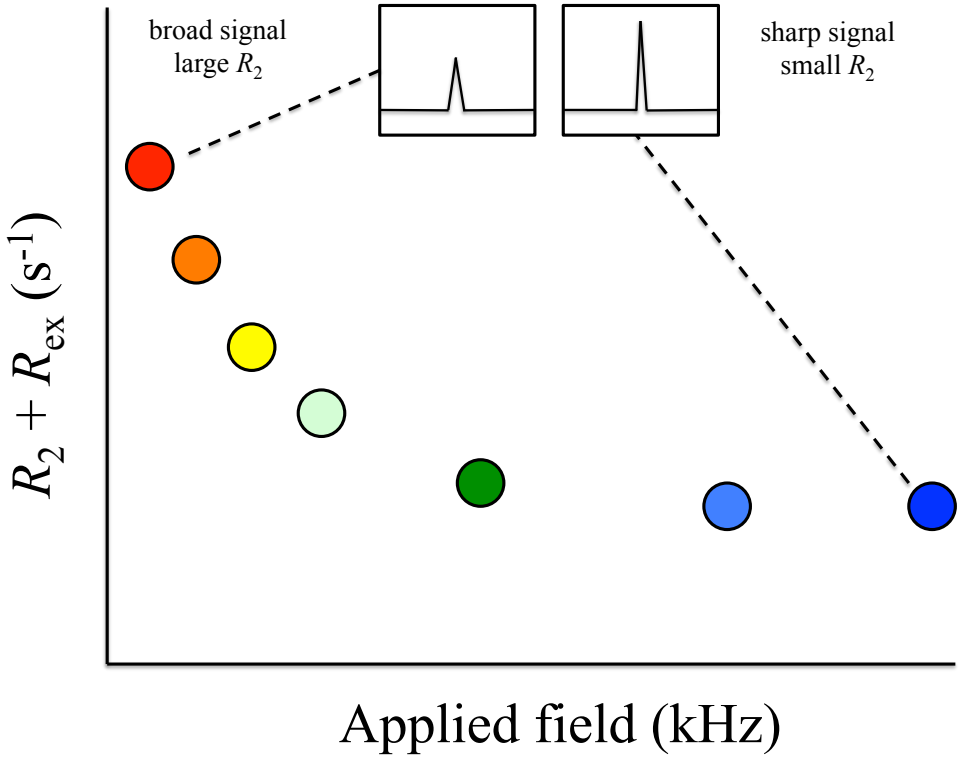

Figure 3.3: Schematic relaxation dispersion curve. Each data point represents the transverse relaxation rate, which is estimated from the peak intensity of the respective resonance.

be extracted. The method reports information at atomic resolution and does not require drastic measures to expose the minor state such as the use of detergents, high temperature or high pressure. It is a highly sensitive technique, as minor states of a population $\approx 1\%$ have been successfully detected.

To probe protein motion, relaxation dispersion quantifies the broadening of resonance lines that it caused by the chemical exchange process (R_{ex}, Fig. 3.3). At strong applied fields, chemical exchange is almost completely refocused and sharp signal lines (high peak intensities) are obtained (Fig. 3.3, right end of the curve), because there is no exchange contribution R_{ex} to the linewidth R_2. Conversely, application of weak fields leads to broad resonance lines (low peak intensities), if the resonance under study shows chemical exchange (Fig. 3.3, left end of the curve). If no chemical exchange is present, $R_{ex} = 0$) and a simple flat line is obtained in this experiment.

RD is measured by one of two methods: R_2-relaxation dispersion or $R_{1\rho}$-relaxation dispersion. The R_2-relaxation dispersion experiment measures the apparent transverse relaxation rate R_2^{eff} as a function of the delay between refocusing pulses in a CPMG-type pulse sequence [63, 64]. In contrast, the $R_{1\rho}$-relaxation dispersion ex-

periment derives the rotating frame relaxation rate $R_{1\rho}^{eff}$ as a function of the strength of an applied spin-lock radio-frequency field.

3.2.3 $R_{1\rho}$-Relaxation Dispersion.

Whether or not an exchange process can be resolved to be studied by RD-NMR depends on its timescale and the experimentally applicable effective field strengths [65]. It has been shown before that CPMG-type R_2-relaxation dispersion experiments can resolve exchange processes on the millisecond timescale ($k_{ex} \approx 2\pi\nu_{\mathrm{CPMG}}$, Fig. 3.2)[2].

Unlike CPMG-type R_2-relaxation dispersion experiments, $R_{1\rho}$-relaxation dispersion can detect chemical exchange processes with exchange rates of the order of $k_{ex} \approx \omega_e = \sqrt{\omega_1^2 + \delta^2}$ in which ω_e is the amplitude of the effective RF field and δ is the offset (the distance of a given resonance from the applied spin-lock field). In these experiments comparably, strong fields $\omega_1/2\pi \approx$ 2-3 kHz are used and pulses can be applied on-resonance ($\delta = 0$) or off-resonance ($\delta \neq 0$). In this thesis, only the simpler on-resonance experiment is discussed. Thus, $\omega_e = \omega_1$.

Strong spin-lock pulses $\geq$ 2 kHz are required to sufficiently refocus the chemical exchange (yielding a sharp signal) to define the right end of the relaxation dispersion curve (Fig. 3.3). On the other hand, weak spin-lock pulses (leading a broad signal) $\ll$ 500 Hz are required to resolve fast exchange processes. Otherwise, $\omega_e^2 \gg k_{ex}^2$ and the exchange remains unresolved.

3.2.4 On-resonance $R_{1\rho}$-Relaxation Dispersion

Because it is difficult to align *multiple* spins along their effective field, when weak spin-lock powers are used, the $R_{1\rho}$-relaxation dispersion experiment needs to be recorded in a way, such that only 1 spin is probed at a time. Thus, the experiment is not performed as a 2D experiment in which all amide resonances of the protein would be probed simultanously (such as in case of the CPMG-type R_2-relaxation dispersion), but as a series of 1D experiments with only a single nucleus probed at a time.

This is achieved by a Hartmann-Hahn polarization transfer scheme, as shown in the pulse sequence shown in Fig. 3.4. The experiment is described in detail in ref. [65]. In brief, the experiment measures the ^{15}N $R_{1\rho}$-relaxation rate for one amide group at a time. The initial ^{15}N pulse in combination with the gradient G1 dephases initial ^{15}N equilibirium magnetization. Next, transverse proton magnetization is created and transferred to nitrogen magnetization by cross-polarization under the strong coupling Hamiltonian using matched on-resonance CW fields ("CP" in

[2] ν_{CPMG} is of the order of 50 Hz – 2 kHz.

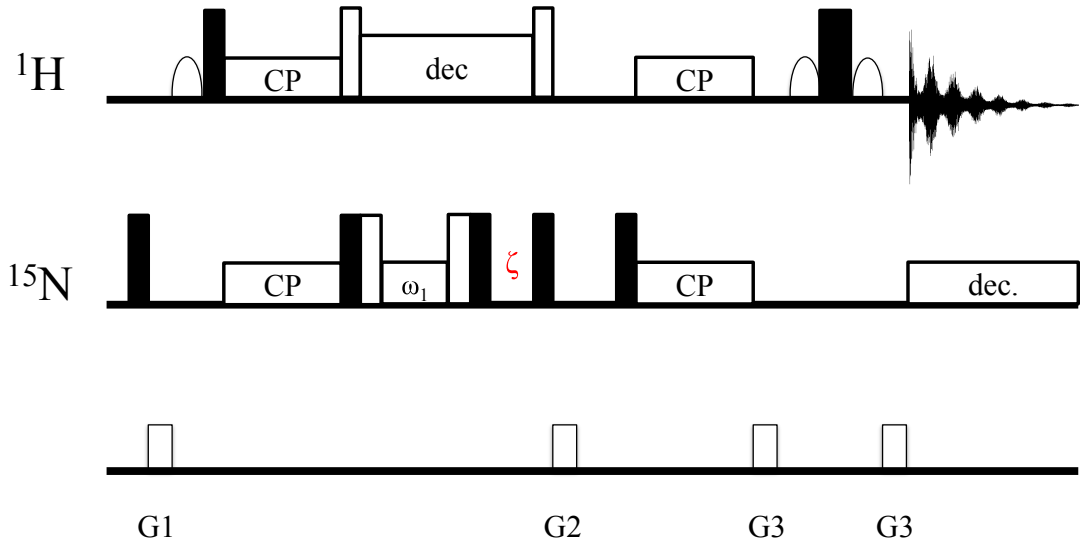

Figure 3.4: Pulse sequence for the $R_{1\rho}$-relaxation dispersion experiment adapted from ref. [65]. A single resonance of interest is excited at a time by cross-polarization (CP) mediated magnetization transfer and spin-locked by application of the ω_1 field. The delay ζ serves to discriminate the on-resonance signal of interest and unwanted near-by resonances (see text). Decoupling is achieved by a WALTZ-16 scheme in the ^{15}N- dimension and by a on-resonance CW field in the proton dimension. For simplicity, pulse phases are omitted from this simplified description.

Fig. 3.4). The ^{15}N spin-lock field ω_1 is applied on-resonance, while a strong on-resonance CW field is applied on the proton channel to suppress J-coupling and cross-correlation effects. Spin-lock powers are varied between experiments to obtain the relaxation rates as a function of applied field, thereby yielding a complete relaxation dispersion profile (Fig. 3.3). Subsequently, magnetization is transferred back to ^{1}H by Hartmann-Hahn matching and passed through a WATERGATE element before detection [48].

It is important to selectively excite the resonance of interest without perturbing nearby resonances, which would lead to an incorrect estimation of the $R_{1\rho}$-relaxation rates. In the case that two resonances have similar ^{1}H, but different ^{15}N frequencies, the delay ζ (red in Fig. 3.4), can be employed to purge undesired resonances and retain only the resonance of interest for acquisition. Herein, the coherence of interest is on-resonance ($\Omega = 0$) and thus remains on the x-axis during the period ζ, whereas the unwanted coherence ($\Omega \neq 0$) rotates 90 degrees (onto the y-axis). After a y-pulse, the undesirable resonance remains in the transverse plane and is purged by the gradient G2. Conversely, the resonance of interest is aligned along the z-axis and is not dephased by the action of G2.

3.2.5 Effective Processing of $R_{1\rho}$-Relaxation Dispersion Data

The use of Hartmann-Hahn polarization transfer in the $R_{1\rho}$-relaxation dispersion experiment has many advantages, the most crucial of which is that very low spin-lock powers can be used to resolve fast exchange processes. However, the switch from a 2D acquisition scheme to a series of 1D experiments also presents challenges. For example, a 2D experiment (typical acquisition for R_2-relaxation dispersion data) would require the acquisition of $\approx$ 20 two-dimensional spectra, each of which will already contain signals of all observable amide protons in the protein. Thus, in total only 20 spectra have to be processed. In stark contrast, the 1D acquisition scheme discussed here requires the measurement of 1 resonance at a time. As a result, a large number of spectra has to be acquired and processed. For example, even for a small protein such as ubiquitin, 75 residues $*$ 20 data points = 1500 spectra have to be acquired. In addition, each dataset has to be apodized, Fourier-transformed, baseline corrected, phase-corrected, peak-picked and subjected to intensity extraction, followed by calculation of the $R_{1\rho}$-relaxation rates and preparation for fitting of the relaxation dispersion curve to extract the dynamical parameters of the exchange process. Manual processing of such large datasets is not practical. Accordingly, I decided to take the chance to develop an automated method for acquisition and processing of $R_{1\rho}$-relaxation dispersion data.

3.3 Experimental Procedures

3.3.1 Sample Preparation

The UBA domain of p62 (residues 391-438) was expressed and purified as described before [25]. Prior to the NMR experiments, the protein was buffer exchanged into NMR buffer (20 mM potassium phosphate pH 6.8, 5 mM potassium chloride, 1 mM EDTA and 1 mM DTT) by dialysis.

3.3.2 NMR Spectroscopy

All NMR experiments were performed at a protein concentration of 1 mM on a Bruker Avance 700 MHz NMR spectrometer in 5 mm Shigemi tubes.

$R_{1\rho}$-relaxation dispersion was measured using a pulse sequence first described by Kay and coworkers [65] (Fig. 3.4). The pulse sequence was programmed as a pseudo-2D experiment in Bruker TopSpin 3.1. Pulse powers were pre-calculated and read into the pulse sequence from a single column file (file *valist*). Reference data (no spin lock to apply) and spin-lock applied data are discriminated by a 0 (reference) or 1 (spin-lock applied) flag in a single column file (file *vclist*). The

delays ζ are calculated by the Python script *ama_zeta.py* (see text) and read into the pulse program from a single column file (file *vdlist*).

Since $R_{1\rho}$-relaxation dispersion experiments give the best temporal resolution if a wide range of ^{15}N spin-lock pulses are applied in the experiments, pulse strengths were optimized. To find ideal spin-lock power conditions achievable on the amplifier at our 700 MHz NMR spectrometer, spin-lock pulse powers of up to 4 kHz were tested. Spin lock pulse powers of $\geq$ 3.5 kHz were not correctly performed by the spectrometer hardware. For this reason, spin-lock powers in the range of 50 Hz – 3 kHz were applied with a ^{15}N spin-lock pulse length of 40 μs.

Peak intensities were converted into $R_{1\rho}$-relaxation rates using the equation:

$$R_{1\rho} = -\frac{1}{T} ln(\frac{I_1}{I_0}) \tag{3.1}$$

in which I_1 is the intensity of a peak with spin-lock applied, whereas I_0 is the reference intensity (no spin-lock applied) and T is the constant time relaxation period (30 - 40 ms).

3.3.3 Statistical Evaluation

For statistical evaluation of the relaxation dispersion data, the extra sum-of-squares F test was employed [66] to decide, whether it is acceptable to fit the $R_{1\rho}$-dispersion data to a relaxation dispersion curve.

The data for each residue were fit to two distinct models:

1. The residue does not show relaxation dispersion (null hypothesis). Accordingly, the rotating frame relaxation rate is constant in the experiment.

$$R_{1\rho} = const. \tag{3.2}$$

2. The residue shows relaxation dispersion (alternative hypothesis). Accordingly, the $R_{1\rho}$-relaxation rate can be described by a two-site fast-exchange model [48, p. 708]:

$$R_{1\rho} = R_2^0 + \frac{k_{ex} p_a p_b \Delta\omega^2}{k_{ex}^2 + \omega_e^2} \tag{3.3}$$

The significance level α was set to 0.999 and the null hypothesis – the respective residue does not show relaxation dispersion – was accepted if p value $\leq \alpha$ was obtained. Moreover, an R_{ex} value of ≤ 2 s^{-1} was considered to be a not significant.

For estimation of the p-value, first the F-ratio was calculated [66]:

$$F = \frac{(SS_{null} - SS_{alt})/SS_{alt}}{(DF_{null} - DF_{alt})/DF_{alt}} \tag{3.4}$$

in which SS are the sum-of-squares (χ^2) of the null and alternative hypotheses and DF are the degrees of freedom of the respective model. Since the distribution of the F ratio is known, this information was used to calculate a p-value from the F ratio and the degrees of freedom of the two models in SciPy[3].

3.4 Results

3.4.1 Development of an Automated Processing Protocol for $R_{1\rho}$-Relaxation Dispersion Data

Overview

For efficient acquisition and processing of $R_{1\rho}$-relaxation dispersion data, I developed an acquisition and processing protocol, which I named Automated $R_{1\rho}$-analysis utility, abbreviated "Amaterasu". This protocol entails the following steps:

1. Acquisition of a ^{1}H-^{15}N-correlation spectrum (HSQC).
2. Peak picking of this spectrum to locate the ^{1}H and ^{15}N chemical shifts of all residues (Bruker TopSpin).
3. Calculation of the ζ-delay values for all residues (Amaterasu).
4. Acquisition of $R_{1\rho}$-relaxation constants for all amino acid residues of the protein at two distinct ^{15}N-spin-lock power values (50 Hz and 3 kHz).
5. Evaluation if the respective residue may show $R_1\rho$-dispersion. (Amaterasu)
6. If step 5 suggested that the respective residue may show $R_{1\rho}$-relaxation dispersion, acquisition of $R_{1\rho}$-relaxation constants at 23 different spin-lock powers. For error estimation, 3 spin-lock power experiments are acquired twice. A reference dataset without application of a spin-lock is also recorded in this step.
7. Fourier transform, automatic phase correction and intensity extraction of the whole dataset[4] (Amaterasu).

[3] SciPy function scipy.stats.f.cdf. The equivalent function in Microsoft Excel is called FDIST.
[4] N residues x 26 spin-lock powers

8. Fitting of the $R_{1\rho}$-data to a relaxation dispersion model and interpretation of the results using the fitting program GLOVE.

9. Visualization of the 1D raw data of the processed spectra to rule out possible artifacts induced by overlapping peaks or distorted baselines (Amaterasu).

10. Statistical evaluation of the results by calculation of the p-value (Amaterasu).

All functions of this protocol have been combined into a graphical user interface (GUI) written in PyQt4 which runs on MacOS and Linux machines[5].

Calculation of ζ-delays

After acquisition of a HSQC spectrum, cross-peaks are quickly picked in Bruker TopSpin by the tab "peak picking". After initial picking, the peak list is inspected and accidentally picked noise peaks and side chain NH_2 resonances are removed. The peak list is exported as a tabular file and saved. This file is used as the input file for the Amaterasu script *ama_zeta.py* which automatically calculates optimal ζ-delays for all residues in the peak list.

As explained in section 3.2.4, the delay ζ serves to selectively record only one single resonance of interest at a time, while not recording nearby resonances that were also excited by the cross-polarization step, since such signals are only slightly off-resonance. A distinct value of ζ has to be calculated for each amide resonance according to [65]:

$$\zeta = \frac{\pi}{2\delta} \tag{3.5}$$

in which δ is the chemical shift difference in the nitrogen-dimension in Hz between the resonance of interest and a nearby undesired resonance.

To record $R_{1\rho}$-relaxation constants for the maximal possible amount of residues, it is important to efficiently calculate the ζ-delays for all residues to suppress undesired neighbour resonances that may overlap and/or interfere with interpretation of the desired on-resonance signal. The ζ calculation in Amaterasu uses two loops to locate neighbour resonances:

1. Read in the peak list of the HSQC spectrum.

2. For each given peak i, check if the peak has a "neighbours" in the proton dimension. For this cause, the difference of the chemical shifts of peak i and all other peaks j, $\Delta\omega_{ij}^{H}$, is calculated.

[5]Amaterasu uses a number of common packages which have to be installed on the system: Python (version 2.7), NumPy, PyQt, Qt, nmrPipe, Tcl/Tk. Moreover, nmrglue [67] is required for plotting of the 1D spectra, if visualization of the spectra to rule out artifacts is desired.

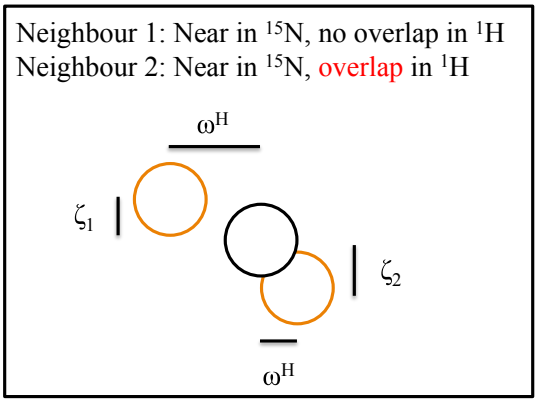

Figure 3.5: Multiple neighbour peaks (orange) for a given cross-peak (black) would lead to different values of ζ, depending on which peak is to be cancelled. In such a case, Amaterasu takes priority to cancel the peak that is closer in the proton dimension.

3. If this difference $\Delta\omega_{ij}^H$ is $\leq$ 0.19 ppm[6], check if i and j are also near in the nitrogen dimension by calculation of $\Delta\omega_{ij}^N$.

4. If this difference $\Delta\omega_{ij}^N$ is $\leq$ 1.9 ppm[7], then the peaks i and j are near in both the proton and the nitrogen dimension and ζ is calculated according to eq. 3.5.

5. For completely isolated resonances, no calculation of ζ is necessary and the value of 0 is assigned.

6. Output of a file containing the ζ delays and a log file containing a report about the calculation.

Multiple Neighbour Peaks Often times HSQC cross-peaks have not only one, but multiple neighbour peaks. In this case, Amaterasu chooses the ζ-value that is most beneficial from a practical point of view. Here, is is more important to cancel acquisition of a peak that overlaps in the proton dimension than a peak that may be near in both dimensions, but is still somewhat separated in the proton dimension (Fig. 3.5).

Irresolvable Values of ζ In some cases eq. 3.5 returns the value infinity since the denominator is 0. This happens if the neighbour peaks have completely identical chemical shifts in the ^{15}N-dimension. In such cases, it is impossible to distinguish the desired and undesired spins using the pulse sequence shown in Fig. 3.4. A

[6]Default value at a static magnetic field of 700 MHz.
[7]Default value at a static magnetic field of 700 MHz.

simple remedy in small proteins is to exploit the fact that chemical shifts and therefore chemical shift dispersion depends on temperature. Therefore, it is sometimes possible to achieve better chemical shift dispersion by slightly shifting the experimental temperature, because resonances that have identical chemical shifts at a given temperature T_1 do not necessarily have identical chemical shifts at a different temperature T_2.

However, for larger proteins one can expect that the chemical shift dispersion is similarly problematic at all temperatures and it may be difficult to resolve the resonances by the simple change in temperature. Moreover, the temperature change directly affects the dynamics of the protein itself, as well as the sensitivity of the NMR experiment, and therefore a change in temperature may not always be desirable.

Screening for $R_{1\rho}$-Relaxation Dispersion

Generally speaking the experimental time accessible to the investigator does not allow the acquisition of a full $R_{1\rho}$-relaxation dispersion dataset for all residues of a protein[8]. Even if sufficient experimental time is available, it should not be used on residues that show no $R_{1\rho}$-relaxation dispersion, but rather used to increase the signal-to-noise ratio of the signals showing relaxation dispersion by acquisition of additional transients. Therefore, it is desirable to include a screening step to only record relaxation dispersion profiles for residues that actually do show $R_{1\rho}$-relaxation dispersion.

For this reason, in the Amaterasu protocol, an initial pilot experiment records data for all residues of the protein using only two spin-lock power values – a weak (50 Hz) and a strong (3 kHz) ^{15}N-spin-lock pulse. This corresponds to recording only the two data points at the left and right ends of a relaxation dispersion curve (Fig. 3.3) The resulting screening dataset will contain $2n$ spectra which are automatically evaluated by Amaterasu. The program reports the intensity ratio of the signal acquired with a weak / signal acquired with a strong spin-lock. A strong spin-lock prevents dephasing of the resonance, whereas a weak spin-lock permits relaxation of the coherence, thereby resulting in a lower peak intensity. Thus, a residue showing $R_{1\rho}$-relaxation dispersion has an intensity ratio $\ll 1$ in this pilot experiment. In the Amaterasu protocol, an arbitrary threshold value (≈ 0.95) is chosen and peaks showing an intensity ratio $\leq$ this threshold value are considered valuable for acquisition of a full dataset. Moreover, problematic residues for example resulting from overlapping resonances are often detected at this stage since they show atypical intensity ratios (≥ 1.2, Table 3.1).

[8] N residues x 23 spin-lock values

Peak	data file (weak/strong spinlock)	$\frac{I_1}{I_0}$	Comment
1	test001.dat / test002.dat	0.80	R1rho!
2	test003.dat / test004.dat	1.01	
3	test005.dat / test006.dat	0.97	
...			

Table 3.1: Screening for $R_{1\rho}$-relaxation dispersion of p62 UBA by the Amaterasu script *ama_screening.py*. If the intensity ratio weak spin-lock to strong spin-lock is below a given threshold, the software judges that for this resonance, a full-data set should be recorded.

After conversion of the "screening" dataset from spectrometer format to nmrPipe format[9] and splitting[10] of the pseudo-2D data file into $2n$ FIDs, where n is the number of resonances recorded with 2 spin-lock powers each, the evaluation by Amaterasu (*ama_screening.py*) works as follows:

1. Enter the directory containing the split data set. Read in all filenames. After splitting, the filenames are test001.dat, test002.dat for the first resonance, test003.dat, test004.dat for the second resonance and so on.

2. Discriminate odd and even numbers in the filename. Odd numbers correspond to weak spin-lock power values, whereas even number correspond to strong spin-lock powers.

3. Perform Fourier transform, apodization and baseline correction for all spectra.

4. Perform automatic first order (Φ_0) phase correction for each pair of spectra.

5. Pick peaks in all 1D-spectra. Although ideally there should be only one peak in each spectrum, some noise peaks or peaks from insufficient ζ-filtering may be accidentally picked. Therefore, filter the peak list to extract only the peak with the highest peak intensity. This should be the desired on-resonance peak.

6. Store the peak intensities of all residues and calculate the intensity ratios $\frac{I_1}{I_0}$ (weak spin-lock spectrum to strong spin-lock spectrum).

7. Write output file. In the case that the ratio is below the set threshold value, add the comment that $R_{1\rho}$-relaxation dispersion is likely to be observed for this residue. An example output file is shown in Table 3.1.

[9] fid.com script in nmrPipe
[10] split2D.com script in nmrPipe

Automated Phase Correction in Amaterasu A problem in the development of this protocol was the fact that by using the CP-based pulse sequence (Fig. 3.4), each resonance requires a different first order phase correction[11]. This required the development of an automatic phase correction module. A very simple algorithm was used to achieve this goal. The algorithm works as follows by maximizing the peak integral of the desired resonance:

1. Perform Fourier transform, apodization and baseline correction to the data.
2. Apply a phase correction of $\Phi_0 = 0$ degrees to the dataset.
3. Extract the on-resonance peak and its intensity from the spectrum. Store this intensity.
4. Apply different phase corrections of $\Phi_0 = 30$, 60, 90, 120, 150, 180, 210, 240, 270, 300 and 330 degrees to the dataset. Store and compare the peak intensities obtained from each phase correction.
5. The phase correction Φ_0 corresponding to the peak of maximal intensity in this set is close to the correct value of Φ_0.
6. Repeat steps 4 and 5 for Φ_0 values around the value of Φ_0 found in step 4. For example, if step 4 yields a value of $\Phi_0 = 30$, screen Φ_0 values from 15 to 45 degrees.
7. The phase correction Φ_0 corresponding to the peak of maximal intensity in this set is the correct value of Φ_0.

The algorithm is admittedly unsophisticated, requiring 42 steps to find the correct phase correction. Nevertheless, it is stable and given today's computers speed fast enough for our purposes taking less than 1 minute to finish for resonances of an entire protein on a normal Desktop PC.

Processing of a Full Dataset in Amaterasu

Processing of a full dataset is similar to the procedure of the screening step, except that now not only two data points, but all $\approx$ 23 data points of the relaxation dispersion curve are actually sampled. In addition, one reference dataset (*no spin lock applied*) is required to convert peak intensities into $R_{1\rho}$-relaxation constants (eq. 3.1). In Amaterasu, the script *ama_processing.py* carries out this task and reports peak intensities in form of a table, in which the leftmost column contains

[11] Since there is only one peak of interest in each spectrum, second order phase corrections are not required.

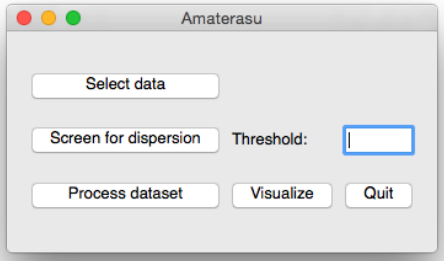

Figure 3.6: Graphical user interface of Amaterasu.

the reference intensities and all other columns contain the peak intensities for various spin-lock strengths applied. This table is easily converted to GLOVE format using a perl script called *r1rho2glove*, which is included in the GLOVE package. Data can subsequently fitted to a $R_{1\rho}$-relaxation dispersion model by GLOVE and plotted for convenient analysis. Data fitting allows for extraction of the dynamical parameters of the exchange process, which concludes the analysis.

Graphical User Interface in Amaterasu

All functions of Amaterasu have been combined into a simple graphical user interface (Fig. 3.6). The correct folder containing the dataset to be processed is selected ("Select data") and subjected to dispersion screening or full dataset processing. For the screening step, an intensity ratio threshold is set by the user (section 3.4.1). The automatically processed spectra can be visualized ("visualize") to check if spectra were correctly phased and whether artifacts from overlapping peaks are present.

3.4.2 $R_{1\rho}$-Relaxation Dispersion of p62 UBA

After finishing the development of Amaterasu, the UBA domain of the human autophagy receptor protein p62 was subjected to $R_{1\rho}$-relaxation dispersion experiments. First, the HSQC spectrum of p62 was acquired (Fig. 3.7). Resonance assignments for the p62 UBA domains were previously reported and could be easily transferred [25].

Cross-peaks were picked and the resulting chemical shifts were used to calculate the ζ-delay for each amide group by the Amaterasu script *ama_zeta.py*.

Resonance	ω_H	ω_N	ζ [s]
1	8.6594	123.012	0.00000000
2	7.4619	118.441	0.00000000
3	7.8552	106.019	0.00000000

4	8.1545	115.512	0.00734099
5	7.2271	110.707	0.00197959
6	7.7612	115.394	0.00734099
7	8.3952	107.425	0.00640668
8	6.8455	128.168	0.00187430
9	8.3483	119.496	0.00187430
10	8.1604	124.535	0.00000000
11	8.9646	124.418	0.00000000
12	7.6145	118.793	0.02072751
13	7.9491	116.332	0.00000000
14	8.2015	123.129	0.00607530
15	7.3680	118.793	0.00607530
16	8.3659	126.997	0.00355927
17	8.2426	106.839	0.00652533
18	8.3541	118.324	0.02072751
19	7.3973	114.574	0.00320334
20	8.7063	117.855	0.05033824
21	8.9059	112.582	0.00800836
22	7.8493	116.215	0.01468199
23	8.0958	107.660	0.00927283
24	8.4304	114.926	0.00463642
25	8.5537	116.801	0.01136670
26	10.0799	132.036	0.05033824
27	7.3327	116.918	0.01136670
28	8.7650	121.020	0.01136670
29	8.1663	122.895	0.00607530
30	7.3151	117.269	0.00000000
31	7.7495	116.449	0.00000000
32	7.2271	120.199	0.01468199
33	8.8120	123.949	0.01136670
34	7.7965	119.145	0.00332422
35	7.5969	117.973	0.00489400
36	8.4128	117.973	0.00542104
37	7.9491	120.785	0.00489400
38	7.0099	119.965	0.00568335
39	7.8728	119.613	0.00409730
40	8.4363	110.238	0.00542104
41	8.1722	123.598	0.01258456
42	8.9705	124.653	0.00194678

43	7.9726	118.676	0.00568335
44	8.6535	116.918	0.00000000
45	8.5420	122.660	0.02710520
46	7.7495	120.902	0.01136670
47	7.8552	113.636	0.01174559
48	7.9608	124.067	0.01258456
49	8.4950	122.426	0.00640668
50	8.0254	124.067	0.00859433
51	7.7965	123.715	0.02710520
52	7.9550	125.473	0.01174559

Table 3.2: Peak picking and ζ-calculation by the Amaterasu script *ama_zeta.py* for the p62 UBA domain.

Next, the screening step was used to separate residues showing $R_{1\rho}$-relaxation dispersion from such residues that do not show relaxation dispersion. The resulting dataset was processed by the script *ama_screening.py* from the Amaterasu GUI. The majority of the residues did not appear to exhibit $R_{1\rho}$-relaxation dispersion. However, several residues showed promising intensity ratios in the screening step. Accordingly, these residues were selected for acquisition of a full dataset.

After acquisition of a full dataset for these resonances, the dataset was processed by *ama_processing.py* called from the GUI of Amaterasu. Peak intensity data were converted into $R_{1\rho}$-relaxation rates by the perl script r1rho2glove and fitted to the $R_{1\rho}$-relaxation dispersion model after Meiboom [48]. The resulting relaxation dispersion profiles were plotted automatically by the GLOVE program. Although the exchange contribution to the rotating frame relaxation constant $R_{1\rho}$ was low in most cases, several promising relaxation dispersion profiles were obtained (Fig. 3.8).

To judge, if it is correct to fit these data to a relaxation dispersion model, rather than to a constant function, which would be the case if the residue did not show chemical exchange, the statistical F-test was applied automatically by the Amaterasu script *ama_ftest.py* (section 3.3.3).

The statistical test convincingly showed that it very likely that these residues are indeed exhibiting chemical exchange. Several residues, however, had to be excluded from the analysis, because of insufficiently resolved relaxation dispersion profiles.

Residues in chemical exchange are mapped onto the crystal structure of the p62 UBA domain in Fig. 3.9. Several residues that exhibit conformational fluctuation are located at or near the dimer interface. Conversely, a mobile residue far from the dimer interface is intriguingly the glycine residue of the conserved MGF motif, which functions in recognition of ubiquitin. This analysis indicates that there are at least two types of conformational exchange present in the p62 UBA domain.

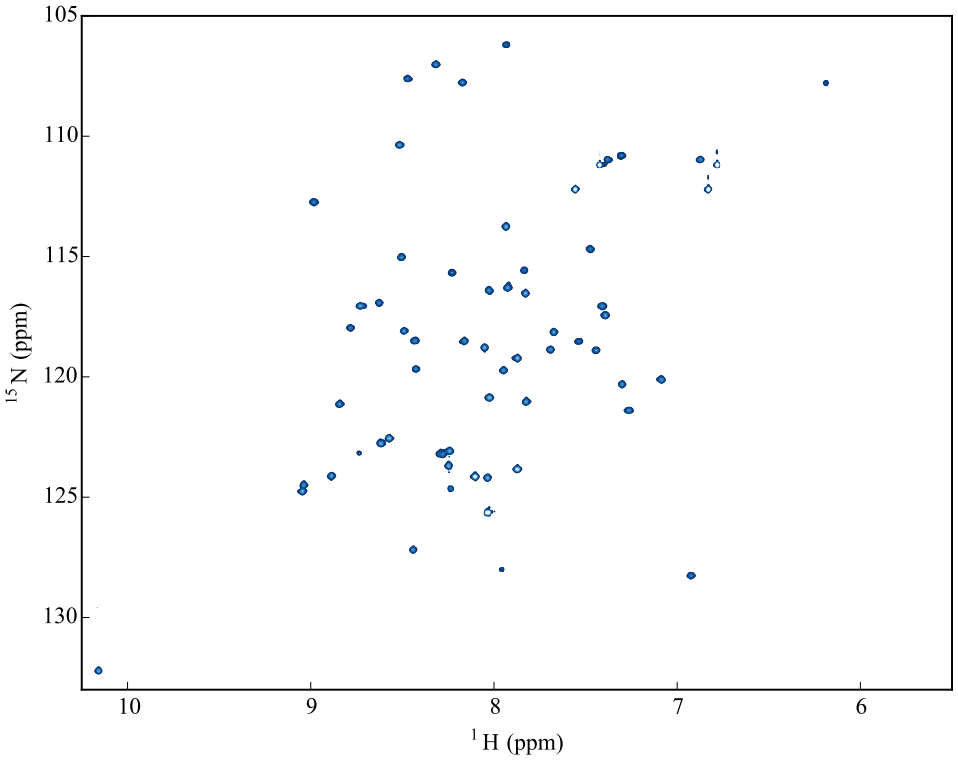

Figure 3.7: ^{1}H-^{15}N-HSQC spectrum of p62 UBA.

Resonance	Assignment	p-value
2	Gly-407	0.9999999871
3	Tyr-424	0.9982175473
5	Asn-423	0.9999999893
6	Thr-421	1.0000000000
7	Asp-393	0.9999995686
9	Ile-426	0.9999985167
10	Asp-410	0.9999999996
12	Gly-412	1.0000000000
15	Ile-397	0.9999996589
18	Ser-436	1.0000000000
19	Asp-425	0.9993225139
20	Thr-432	1.0000000000
22	Ala-429	1.0000000000
23	Asp-431	0.9989773267
24	Gln-402	0.9998620105
27	His-438	0.9999979317

Table 3.3: P-values obtained from the statistical F-test comparing the null hypothesis (no conformational exchange) and the alternative hypothesis ($R_{1\rho}$-relaxation dispersion owing to conformational exchange).

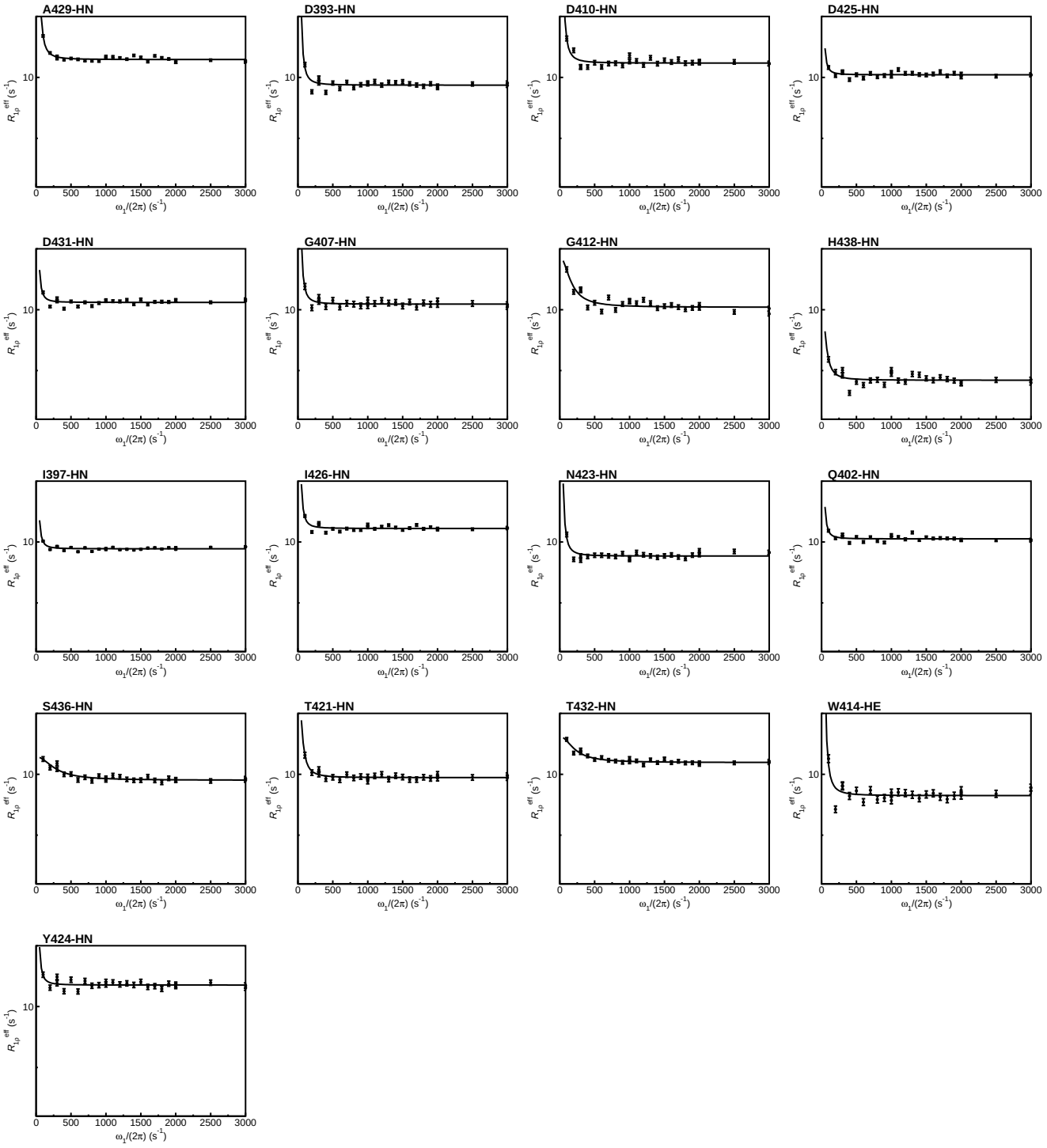

Figure 3.8: $R_{1\rho}$-relaxation dispersion profiles of the p62 UBA domain.

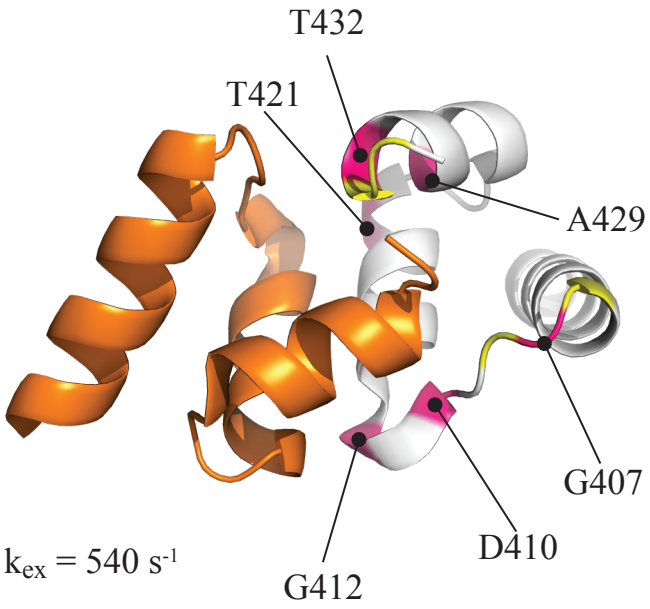

Figure 3.9: $R_{1\rho}$-relaxation dispersion observed in the UBA domain of autophagy receptor protein p62. Residues predicted to function in ubiquitin recognition are colored yellow, whereas residues showing conformational exchange as detected by $R_{1\rho}$-relaxation dispersion are colored magenta.

3.5 Discussion

In this section I showed that the UBA domain of p62 is not a static molecule, but rather exhibits conformational exchange. Although in most cases the exchange contribution to the linewidth R_{ex} was of the order of 2-10 s^{-1}, which is a rather small value, it was possible to unambigously detect the exchange process, as reflected by the high p-values obtained in the statistical F-test (Table 3.3). However, the magnitude of the R_{ex} was too low to extract all dynamic parameters of the exchange process, such as the chemical shift difference $\Delta\omega$ of major and minor conformation. Nevertheless, the fact that $R_{1\rho}$-relaxation *was observed* suggests that in addition to the p62 UBA dimer represented by the crystal structure, there exists a minor conformer in solution with unequal chemical shift.

In these experiments, the p62 UBA domain showed two distinct kinds of dynamics. One type of exchange is found at the interface of the homodimer (residues Asp^{410}, Gly^{412}, Thr^{421}, Ala^{429}, Thr^{432}; Fig. 3.9). Because the residues at the homodimer interface also are the residues, which show chemical exchange in the $R_{1\rho}$-relaxation dispersion experiments, it is tempting to speculate that the observed dispersion reflects the monomer-dimer exchange process. Indeed, Isogai et al. reported that in solution, the p62 UBA exists in a monomer-dimer equilibrium, which is in slow-exchange on the NMR timescale [25]. This observation is in fine agreement with the exchange rate obtained from $R_{1\rho}$-relaxation dispersion experiments ($k_{ex} \approx 540s^{-1}$) of this study. However, to prove the hypothesis that the observed conformational exchange is indeed the monomer-dimer association, it would be necessary to extract the chemical shift of the minor popuation from $R_{1\rho}$-relaxation dispersion measurements to check, if this chemical shift correllates with the chemical shift of the p62 UBA monomer.

The second region exhibiting $R_{1\rho}$-relaxation dispersion is intriguingly the glycine residue Gly^{407} of the conserved MGF motif that functions in ubiquitin recognition. Here, it is tempting to speculate that this motion reflects the fact that the amide proton of Gly^{407} is found in different chemical environments, which would be the case if this loop adopted various conformations to accomodate ubiquitin binding, perhaps by screening various orientations. Nevertheless, since the dimeric form of p62 UBA shows a very low affinity toward ubiquitin, it would be interesting to compare $R_{1\rho}$-relaxation dispersion data of a monomeric form of p62 UBA to see if the motion of Gly^{407} is enhanced in absence of homodimerization.

In summary, conformational dynamics of the UBA domain of p62 were successfully detected. In addition, I anticipate that the methodology introduced in this section will help the characterization of conformational exchange of many proteins the chemical exchange of which is not detectable by CPMG-type R_2-relaxation dis-

persion experiments.

Chapter 4

Activation of Mitophagy E3-Ligase Parkin by Phosphoubiquitin

4.1 Abstract

Mutations in the gene for parkin are the most common cause of autosomal-recessive juvenile Parkinsonism. Parkin is an auto-inhibited E3 ubiquitin ligase and its activation required for initiation of mitophagy, an event which is poorly understood. Here, I describe that binding of phosphoubiquitin induces structural rearrangements in the catalytic core of parkin, thereby priming parkin for catalysis. Notably, the action of ubiquitin is counterbalanced by the action of the parkin UBL domain *in cis*. Thus, I propose a novel competition-based mechanism for regulation of parkin activity.

4.2 Introduction

Mutations in the PARK2 gene which encodes the protein *parkin* have been found to cause a familial form of Parkinson's Disease termed autosomal-recessive juvenile parkinsonism (AR-JP) [15]. Parkin is an E3-ubiquitin ligase of the RBR family [20] (Fig. 4.1 A) which resides in the cytosol under basal conditions [17]. Cytosolic parkin is thought to be catalytically inactive. Accordingly, multiple crystal structures have recently been reported showing parkin in an auto-inhibited conformation: both the interaction platform with the incoming E2 enzyme and the catalytic residue Cys^{431} are occluded due to intramolecular interactions [20] (Fig. 4.1 B). Since parkin functions in the ubiquitylation of mitochondrial proteins, an event which initiates degradation of dysfunctional mitochondria by mitophagy, activation of parkin by external factors is implied [17].

In living cells, at least two events are required to release the latent E3-activity of parkin. First, upon loss of the electrochemical potential across the inner mitochondrial membrane the mitochondrial protein kinase PINK1, sensing mitochondrial dysfunction, phosphorylates parkin at residue Ser^{65} of its UBL domain [68]. Second, PINK1 also phosphorylates ubiquitin intriguingly at the same residue Ser^{65} which is conserved between ubiquitin and the parkin UBL domain [17] (Fig. 4.1 C).

While these two phosphorylation events are not on their own sufficient to induce translocation of parkin to mitochondria, they are required for translocation and are sufficient to release the latent E3-ubiquitin ligase activity of parkin *in vivo* [17, 69]. How these two phosphorylation events convert parkin from an inactive into a catalytically active state is elusive. A recent molecular dynamics simulation reported that phosphorylation of the UBL domain of parkin might alter the intramolecular dynamics of parkin [70]. However, experimental evidence is still scarce. In particular, the role of phosphorylated ubiquitin in parkin activation is remains completely unexplained. The leading hypothesis is that phosphorylated ubiquitin activates parkin in an allosteric manner [17]. Therefore, I set out to investigate the possibility of a direct modulation of parkin activity by phosphorylated ubiquitin.

4.3 Experimental Procedures

4.3.1 Protein Expression and Purification

Ubiquitin was expressed and purified as described previously [28]. Full-length parkin (residues 1-465) and its fragment lacking the N-terminal UBL domain (parkin ΔUBL; residues 141-465) were expressed and purified as reported before [20]. In brief, BL21(de3) *Escherichia coli* cultures harboring the parkin expression vector were

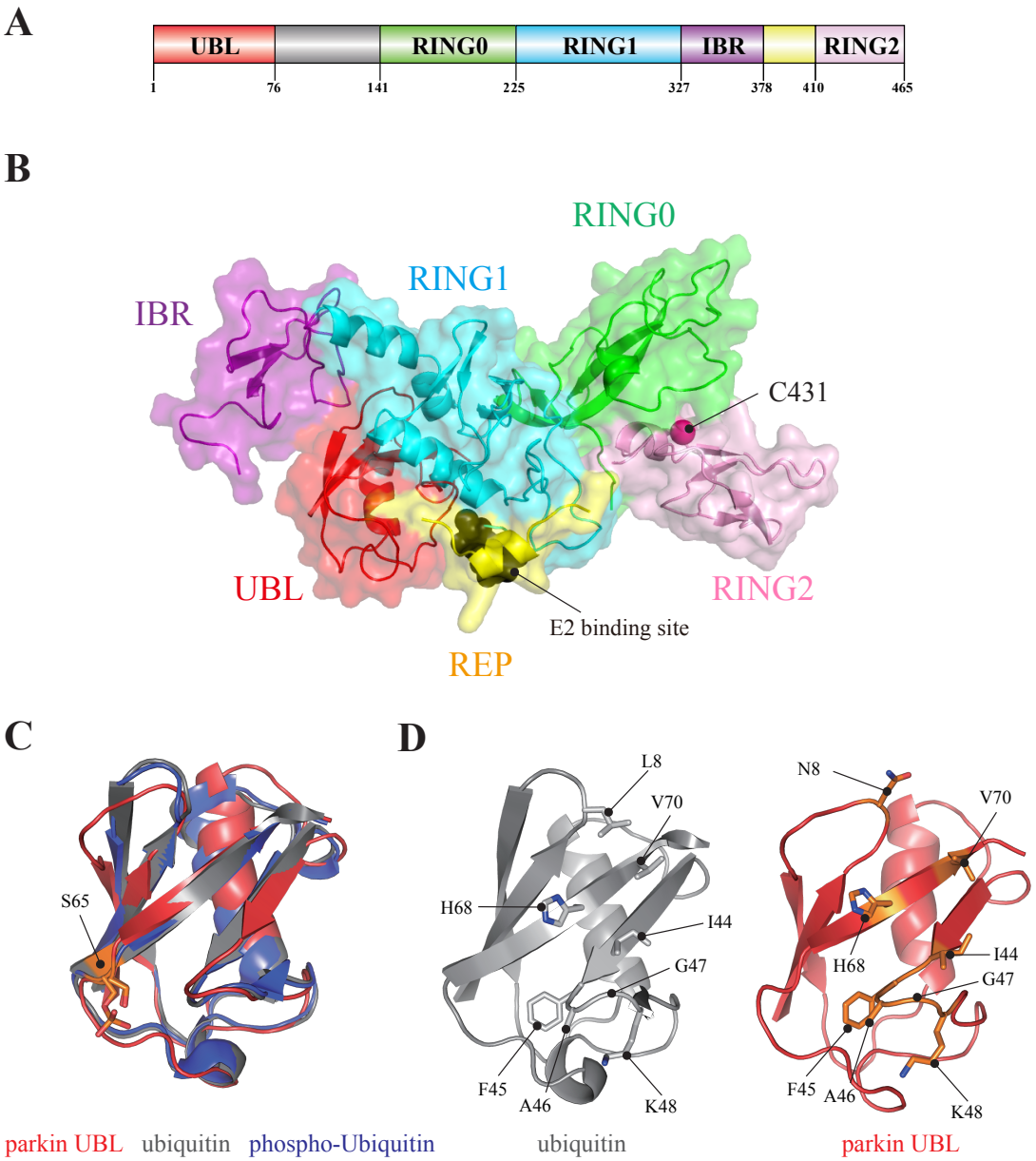

Figure 4.1: Parkin is an auto-inhibited E3 enzyme that requires structural rearrangements for activation. A, Domain architecture of rat parkin. B, Quaternary domain arrangement in full-length parkin (PDB ID 4K95 [20]). Key sites are indicated. C, Structural resemblance of ubiquitin, phosphorylated ubiquitin and the UBL domain of parkin (PDB IDs 1UBQ, 4WZP, 4K95 [20, 71, 72]). The conserved phosphorylation site Ser^{65} is indicated in orange. D, Amino-acid conservation in the hydrophobic patch used to recognize the ubiquitin fold by the majority of biomolecules between ubiquitin and parkin UBL [8, 73].

grown to an OD_{600} = 0.5 at 37°C. The temperature was decreased to 16°Cand expression was induced by addition of 25 μM IPTG overnight. Cells were harvested, lysed by sonication and GST-parkin was purified by glutathione-Sepharose 4FF (GE Healthcare) column chromatography. After cleavage of the GST affinity tag by PreScission protease, parkin was further purified by size-exclusion chromatography on a HiLoad 16/60 Superdex 75pg column (GE Healthcare). The parkin UBL domain (residues 1-76) was expressed and purified as described previously [74]. Point mutations of parkin and ubiquitin were generated by PCR. Representations of protein structures were created using PyMol.

4.3.2 Isothermal Titration Calorimetry

Isothermal titration calorimetry (ITC) experiments were conducted at room temperature (298 K) on a MicroCal ITC200 system. All protein samples were dialyzed against ITC buffer (20 mM HEPES, pH 7.5, 100 mM sodium chloride, 0.1 mM TCEP[1]) overnight and thoroughly degassed before each experiment. Samples were incubated at room temperature before each experiment.

Syringe protein concentrations (ubiquitin, phosphomimetic ubiquitin and parkin UBL) ranged from 0.8 to 1.5 mM, whereas the cell (parkin or parkin ΔUBL) contained 87-100 μM protein. Syringe protein was injected into the cell at 3-min intervals. The resulting data were processed using Origin 7 (MicroCal Software, Inc.). Errors in the derived data represent the standard error of the mean of three independent experiments.

4.3.3 NMR Spectroscopy

All NMR spectra were acquired on a Bruker Avance 700 MHz spectrometer equipped with a 5 mm $^{15}N/^{13}C/^{1}H$ z-gradient triple resonance cryoprobe. The FIDs were apodized, zero-filled and Fourier-transformed using NMRPipe [29]. The resolution in the indirect dimension was improved by linear prediction. Spectra were analyzed in CCPN [32]. Due to the large difference in chemical shift between ubiquitin and phosphomimetic (S65D) ubiquitin, resonance assignments could not be unambiguously transferred. Accordingly, resonances of S65D ubiquitin were newly assigned from HNCACB, CBCA(CO)NH, HNCO and HN(CA)CO triple-resonance experiments [48].

Normalized chemical shift differences between wild-type and mutant ubiquitin were calculated according to eq. 2.1. Figures were created by the Python modules nmrglue [67] and matplotlib [75].

[1] Tris(2-carboxyethyl)phosphine

4.3.4 Fluorescence Spectroscopy

Fluorescence was measured using a FluoroMax4 fluorescence spectrometer (HORIBA). Tryptophan fluorescence was selectively excited at 300 nm and emission spectra were acquired over wavelengths of 310 to 400 nm with the slit width set to 5 nm. Before the experiments, proteins were dialyzed into 20 mM HEPES, pH 7.5, 100 mM sodium chloride, 0.1 mM TCEP. Parkin was diluted to a final protein concentration of 7 μM and measured at a volume of 500 μl. The spectral contribution of the buffer was subtracted from the acquired spectra using the software supplied by the manufacturer (FluorEssence v3). In each measurement, 10 transients were acquired were acquired and averaged. The redshift of the fluorescence spectrum upon ligand addition was evaluated by calculation of the barycentric mean of the spectrum. The barycentric mean was calculated as:

$$\lambda_{bcm} = \sum \frac{F(\lambda)\lambda}{F(\lambda)} \tag{4.1}$$

in which $F(\lambda)$ is the tryptophan fluorescence emission intensity at λ nm. Fluorescence redshifts are reported as differences from the initial barycentric mean:

$$\Delta\lambda_{bcm} = \lambda_i - \lambda_0 \tag{4.2}$$

in which λ_i is the barycentric mean of the i-th spectrum (i-th addition of ligand) and λ_0 is the barycentric mean of the fluorescence spectrum before addition of ligand.

4.4 Results

4.4.1 Physical Interaction of Parkin and Ubiquitin

In the reaction mechanism of RBR-type E3 ligases, ubiquitin is – after E1-mediated activation and subsequent transfer to E2 – covalently transferred from the E2 to the catalytically active cysteine residue on the E3 by transthiolation [73]. Thus, in general, isolated ubiquitin is not expected to bind to the E3 enzyme in a non-covalent manner. Therefore, I was surprised to find a physical, non-covalent interaction between ubiquitin and parkin by NMR spectroscopy. Addition of 1 mol. eq. full-length parkin to a solution of ^{15}N-labeled ubiquitin drastically changed its TROSY-HSQC spectrum (Fig. 4.2A). Several resonances such as Ile^{36} shifted in an intermediate to fast-exchange manner upon addition of parkin while displaying a small degree of line broadening. Conversely, other resonances such as Leu^{8} or Val^{70} were most severely broadened and were no longer detectable after addition of parkin. This observation suggested that ubiquitin does bind to parkin and that it does so by a specific interaction.

Line broadening occurs because during the NMR experiment ubiquitin is in exchange between a free state and at least one parkin-bound state leading to an exchange contribution to the transverse relaxation constant R_2. Not all residues, however, experience the same degree of broadening. Therefore, the signal intensity ratio ubiquitin sample with / without parkin varies in a residue-specific manner (Fig. 4.2B). The complete loss of the signals Leu^8 and Val^{70} suggested that parkin binds to the Leu^8-Ile^{44}-His^{68}-Val^{70} hydrophobic patch on ubiquitin which is a canonical site for recognition of the ubiquitin fold [73]. Thus, intriguingly, parkin appears to bind to the same hydrophobic patch on ubiquitin as it does in binding to its own UBL domain *in cis* [20, 74].

Where on parkin does ubiquitin bind? Although the present analysis cannot directly answer this question, the simplest hypothesis is that ubiquitin binds to the same site as does the UBL domain in full-length parkin. Indeed, both structure and sequence are highly conserved between ubiquitin and the UBL domain of parkin (Fig. 4.1C). Although the global sequence conservation is only 30% as determined by ClustalW, residues on ubiquitin / parkin UBL that interact with core parkin show a much higher sequence identity between the two molecules (Fig. 4.1D). Accordingly, superimposing ubiquitin onto the UBL binding site in parkin brings residues that show severe line-broadening in the TROSY spectrum in close proximity to the surface of parkin (Fig. 4.2C), thereby explaining the selective broadening as influence from interaction with the surface of parkin. Thus, I adopted the working hypothesis that ubiquitin and parkin UBL interact with parkin via the same binding site.

4.4.2 Ubiquitin Binding to Parkin is Entropy-driven

The line broadening of the NMR signals was difficult to interpret in a quantitative manner. To evaluate the binding of ubiquitin and parkin more accurately, I employed isothermal titration calorimetry (ITC). Indeed, ITC experiments confirmed the binding of ubiquitin to parkin. However, to my surprise, addition of ubiquitin to parkin resulted in an uptake of heat from the environment (endothermic reaction, Fig. 4.3A). Data fitting yielded a dissociation constant of $\approx 70\ \mu M$, (Fig. 4.3C), which is in line with the intermediate to fast-exchange behavior observed by NMR spectroscopy [48].

Since not ubiquitin per se, but Ser^{65}-phosphorylated ubiquitin is thought to activate parkin, one would expect a stronger affinity of phosphorylated ubiquitin for parkin as compared to wild-type ubiquitin. To probe this possibility, I employed the S65D ubiquitin variant which has been successfully used as a mimic for phosphorylated ubiquitin before [17]. I further confirmed that the chemical shift perturbation induced by the S65D mutation agreed with previously reported structural data of

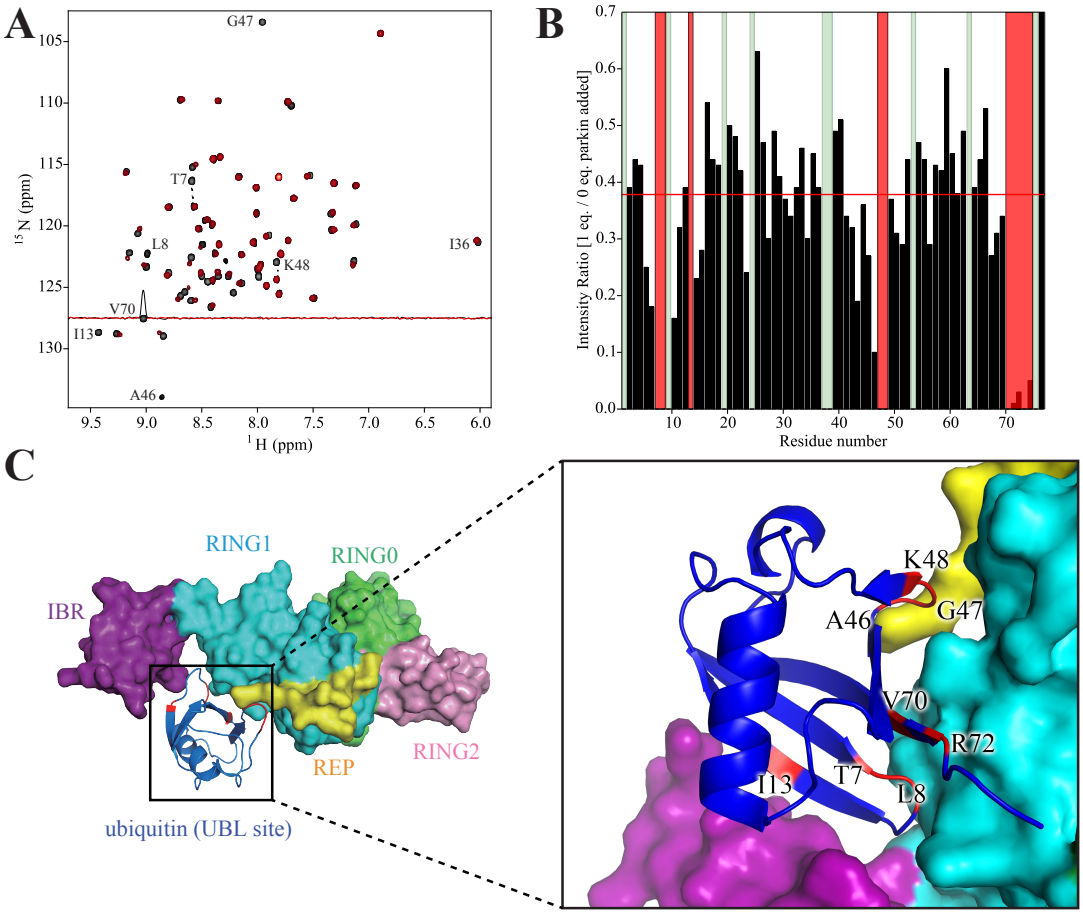

Figure 4.2: Physical interaction of parkin and ubiquitin. A, TROSY-HSQC spectrum of 100 μM ^{15}N-labeled ubiquitin before (black) and after (red) addition of 1 mol. eq. parkin acquired at 310 K. The spectrum of ubiquitin shows selective line broadening after addition of parkin. B, Residue specific NMR signal loss in ubiquitin upon addition of parkin. Green bars represent residues that are not observed in the reference spectrum (e.g. proline or residues intrinsically broadened due to conformational exchange), whereas red bars indicate residues where near-complete signal loss occurred due to parkin addition. Red line: average intensity ratio. C, Hypothetical model of ubiquitin binding to parkin. Ubiquitin (PDB ID 1UBQ, ref. [72]) is superimposed on the UBL binding site in the crystal structure of parkin (r.m.s.d. 1.6Å; PDB ID: 4K95, [20]). Residues of ubiquitin that show significant line broadening in the NMR experiment (A, B) are colored red. If bound to the UBL site in parkin, most of these ubiquitin residues come in close contact to the RING1 (cyan) and IBR (purple) domains of parkin.

phosphorylated ubiquitin ([71], Fig. 4.4). Strikingly, ITC experiments with phosphomimetic ubiquitin (Fig. 4.3B) resulted in a dissociation constant of $\approx$ 35 μM which is about 2-fold stronger than the binding of wild-type ubiquitin to parkin (4.3C). Moreover, the entropic contribution to the binding reaction was enhanced in the case of phosphomimetic ubiquitin (Fig. 4.3D). Taken together, these results strongly argue that parkin has a specific, entropy-driven recognition mode for phosphorylated ubiquitin.

4.4.3 Competition of Ubiquitin and Parkin UBL for Core Parkin Binding

I was surprised by the entropy-driven nature of (phosphomimetic) ubiquitin binding to parkin. Therefore, I considered the possibility that the entropy change may reflect the conversion of the compact multi-domain arrangement of parkin into a more flexible form with higher conformational entropy. In light of the observation that ubiquitin uses a similar interface as the UBL domain to bind to core parkin (Fig. 4.2), I asked if ubiquitin and the UBL domain could compete for core parkin binding resulting in formation of a less compact molecule with higher conformational entropy. To probe this possibility, I deleted the N-terminal UBL domain in parkin (parkin ΔUBL; residues 141-465) and employed ITC.

If competition between parkin UBL and ubiquitin occurred, deletion of the UBL domain in parkin should favor ubiquitin binding. Indeed, parkin ΔUBL showed an increased affinity for wild-type ubiquitin (Fig. 4.5A; K_{d} 18 μM) as compared to the full-length protein (Fig. 4.3A; $K_{\mathrm{d}} \approx 70$ μM). This observation supported the hypothesis that parkin UBL and ubiquitin compete for core parkin binding. Nevertheless, deletion of the UBL domain did not change the endothermic nature of the reaction, indicating that the major entropic driving-force behind the reaction is independent of the UBL-ubiquitin competition event.

Since phosphomimetic ubiquitin bound to full-length parkin about 2-fold more tightly than did ubiquitin (Fig. 4.3C), I asked if this preference for the phosphomimetic form of ubiquitin is preserved in absence of the UBL domain of parkin. ITC experiments showed that phosphomimetic ubiquitin bound to parkin ΔUBL with a dissociation constant of $\approx$ 1.5 μM (Fig. 4.5B, D). Thus, the specificity for phosphomimetic ubiquitin is even more pronounced in parkin ΔUBL ($\approx$ 10-fold increase between wild-type and phosphomimetic ubiquitin) as compared to full-length parkin ($\approx$ 2-fold increase). As in all cases discussed so far, the reaction was entropy-driven and the entropic contribution was larger for phosphomimetic ubiquitin as compared to the wild-type protein (Fig. 4.5D).

If the UBL domain of parkin and (phosphorylated) ubiquitin compete for core

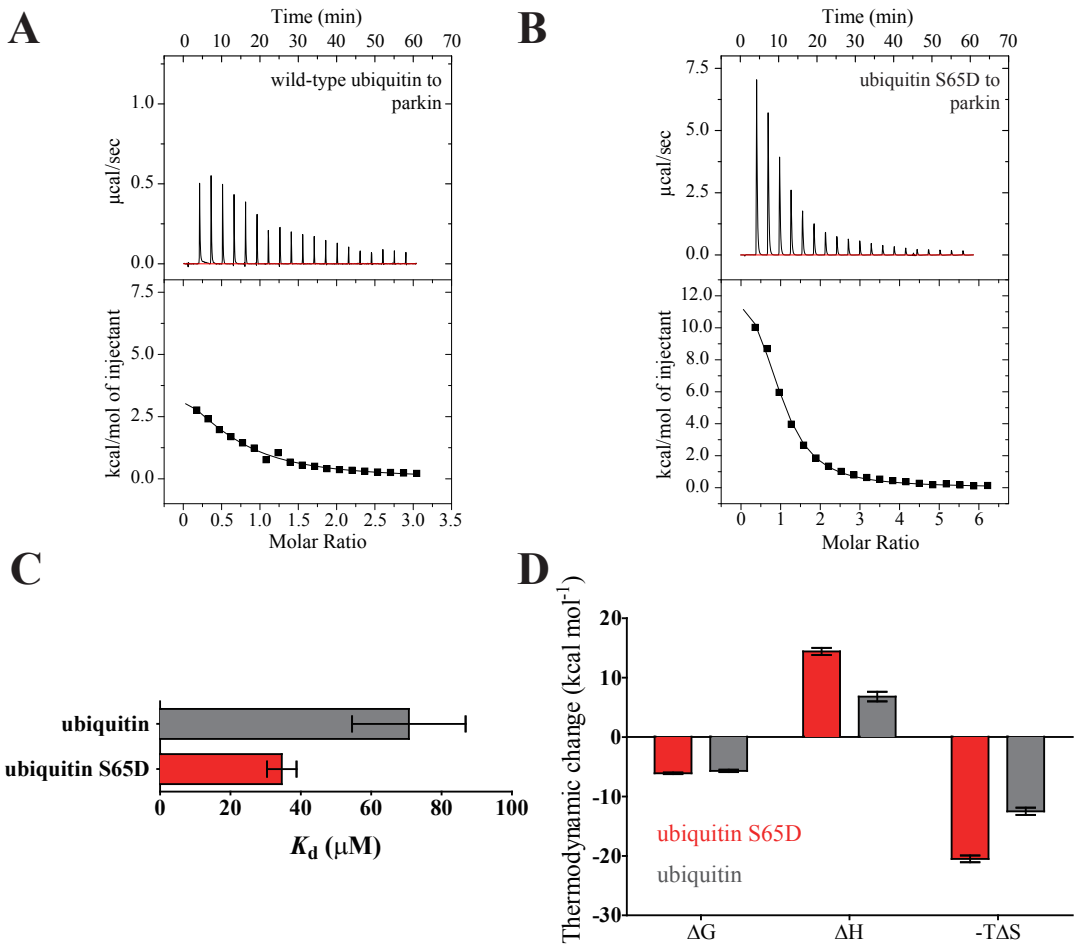

Figure 4.3: Ubiquitin binding to parkin is entropy-driven. A, B, Isothermal titration calorimetry thermogram for ubiquitin (A) and phosphomimetic ubiquitin (B) binding to full-length parkin showing an endothermic reaction. The upper panels show raw data, and the lower panels represent the integrated heat values. C, Dissociation constants determined from the ITC experiments. Phosphomimetic ubiquitin binds about 2-fold more strongly to full-length parkin as compared to wild-type ubiquitin. D, Entropy drives the binding of both ubiquitin and phosphomimetic ubiquitin to parkin. Although enthalpically the binding process is disfavored, this is compensated by a large favorable entropic term. Error bars in C, D: standard error of the mean.

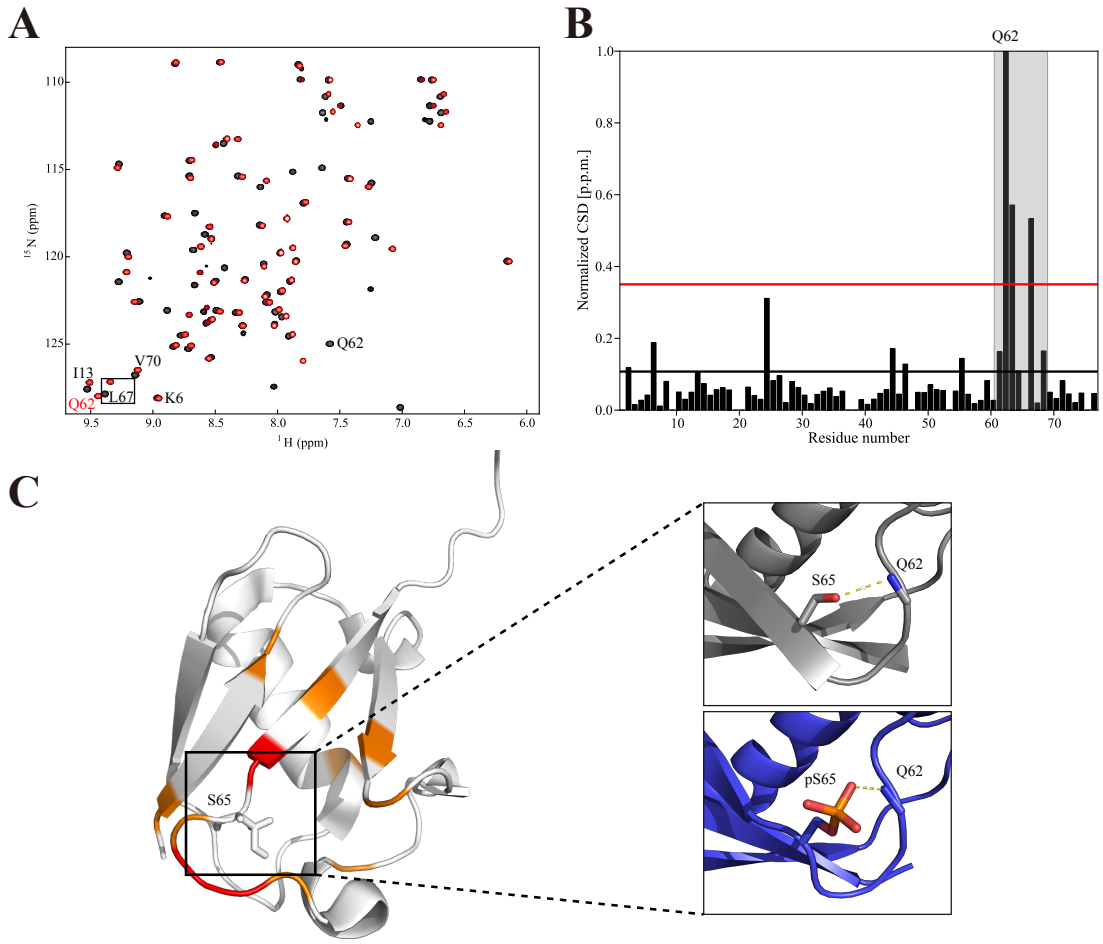

Figure 4.4: Characterization of phosphomimetic ubiquitin. A, ^{1}H-^{15}N HSQC spectra of ubiquitin (black) and phosphomimetic (S65D) ubiquitin. Large differences in chemical shift precluded unambiguous transfer of the assignments. For example, there are only 4 cross-peaks in the lower left corner (around 9.25 ppm) HSQC of wild-type ubiquitin, whereas 5 resonances appear in the HSQC of the S65D ubiquitin. Accordingly, phosphomimetic Ub was assigned anew using triple-resonance experiments. B, Normalized chemical shift difference (CSD) between wild-type and S65D ubiquitin. Black line: average CSD; red line: average CSD + 1 σ. The region with the most significant CSD is centered at the modified Ser^{65} residue (grey box). Gln^{62} has a CSD value of 1.9. For comparison, the CSD-axis range is truncated at a CSD value of 1.0. C, Chemical shift difference values mapped onto the crystal structure of ubiquitin (PDB ID 1UBQ). The largest CSD is observed for the amide group of Gln^{62}. As pointed out before [71], in Ser^{65}-phosphorylated ubiquitin the phosphate group at Ser^{65} forms a hydrogen bond with the backbone amide of Gln^{62} (PDBID 4WZP; ref. [71]), whereas this hydrogen bond is donated by the hydroxyl group of Ser^{65} in wild-type ubiquitin. Accordingly, S65D ubiquitin appears to mimic key structural features of Ser^{65}-phosphorylated ubiquitin which agrees with reports that S65D ubiquitin is a good mimic of phosphorylated ubiquitin *in vivo* [17].

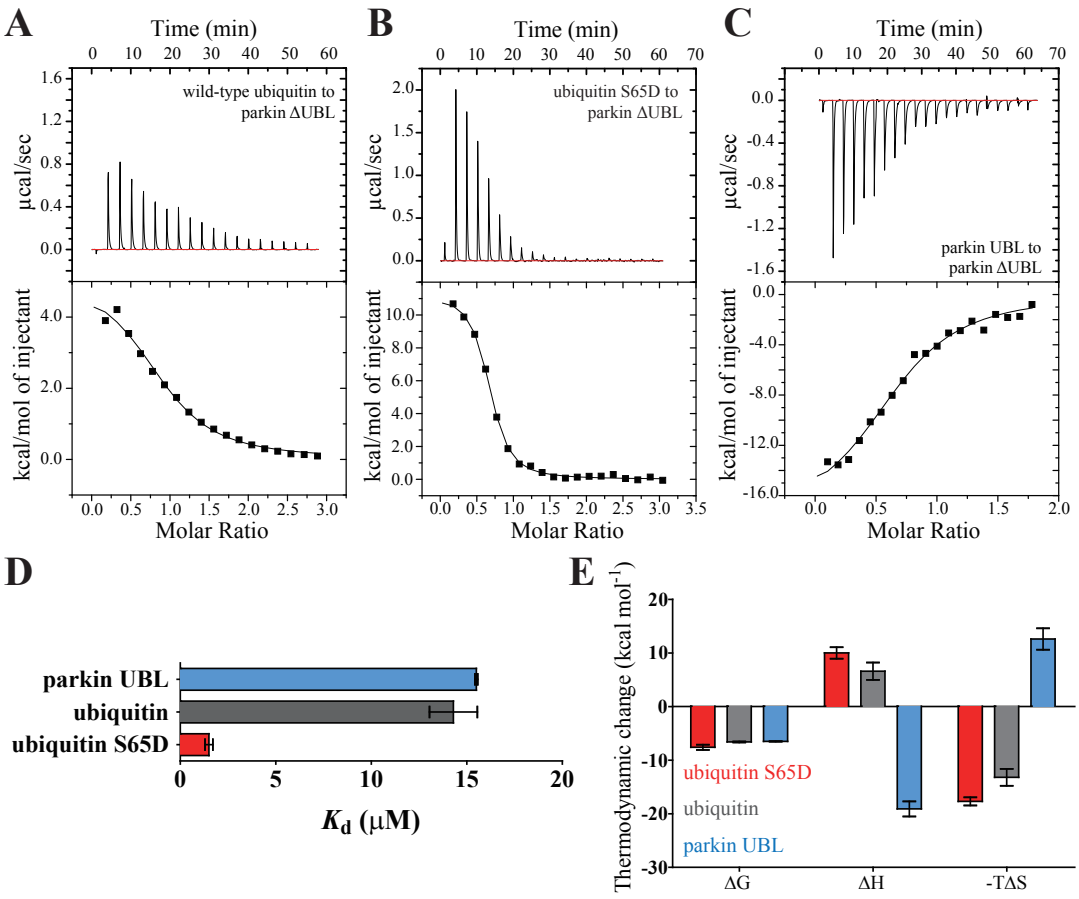

Figure 4.5: Competition of ubiquitin and parkin UBL for core parkin binding. A-C, Isothermal titration calorimetry thermograms for ubiquitin (A), phosphomimetic ubiquitin (B) and parkin UBL (C) binding to parkin ΔUBL. As in the binding to full-length parkin, binding of ubiquitin and phosphomimetic ubiquitin to parkin ΔUBL are endothermic processes. Strikingly, the ITC profile of parkin UBL binding to parkin ΔUBL shows an exothermic reaction. The upper panels show raw data, and lower panels represent the integrated heat values. D, Dissociation constants determined from the ITC experiments. Parkin ΔUBL shows a markedly increased affinity for ubiquitin and phosphomimetic ubiquitin as compared to full-length parkin (Fig. 4.3C). Notably, the phosphomimetic form binds about 10-fold more strongly to parkin ΔUBL than wild-type ubiquitin. E, Thermodynamic parameters for the binding of ubiquitin, phosphomimetic ubiquitin and parkin UBL to parkin ΔUBL. While the binding of both ubiquitin and phosphomimetic ubiquitin to parkin ΔUBL are still entropy-driven, binding of parkin UBL to parkin ΔUBL is an enthalpy-driven reaction. The entropy change upon binding is most pronounced for phosphomimetic ubiquitin. Error bars in D, E: standard error of the mean.

parkin binding, the UBL domain of parkin should also be capable of binding to parkin ΔUBL *in trans*. I investigated this possibility by ITC and found that parkin UBL binds parkin ΔUBL with a K_d of ≈ 16 μM (Fig. 4.5C). Accordingly, parkin ΔUBL shows essentially the same affinity towards wild-type ubiquitin and its UBL domain *in trans*, while phosphomimetic ubiquitin is bound approximately 10-fold more tightly (Fig. 4.5D). Strikingly, binding of the UBL domain to parkin ΔUBL is an exothermic reaction (Fig. 4.5C). Thus, although ubiquitin and UBL bind to parkin ΔUBL with essentially the same affinity, the driving force behind the binding is fundamentally reversed. In conclusion, these results strongly suggest that the UBL domain of parkin and ubiquitin compete for binding to core parkin, which in turn shows a strong preference for the phosphomimetic form of ubiquitin.

4.4.4 Structural Changes in the Remote Catalytic Core of Parkin.

The entropy-driven nature of (phosphomimetic) ubiquitin binding to parkin suggested that conformational changes accompany the binding. To investigate this possibility, I exploited the fact that ubiquitin has no tryptophan residues, whereas rat parkin has seven tryptophan residues, which are – with the single exception of Trp^{97} – conveniently located in the vicinity of the catalytic center of parkin (Fig. 4.6A, upper). Thus, tryptophan fluorescence emission spectra of parkin will specifically reflect the chemical environment in the catalytic core of parkin.

Addition of ubiquitin to full-length parkin resulted in a measurable redshift of its tryptophan emission spectrum (Fig. 4.6B), suggesting that tryptophan residues became increasingly solvent-exposed upon ubiquitin binding. Importantly, the phosphomimetic form of ubiquitin had a comparably larger effect on the emission spectrum of parkin (Fig. 4.6B, C). This indicated that phosphorylated ubiquitin may be capable of inducing larger overall structural changes in parkin as compared to ubiquitin.

Equivalent experiments with parkin ΔUBL confirmed that (phosphomimetic) ubiquitin induces structural changes in the catalytic core of parkin as indicated by the redshift of the fluorescence emission spectrum (Fig. 4.6D). Once again, phosphomimetic ubiquitin appeared to induce a larger opening in the structure of parkin than ubiquitin (Fig. 4.6E). This observation also provided reassurance that the presence of Trp^{97} did not affect the conclusion of the experiments with full-length parkin (Fig. 4.6B, C). Taken together, these results strongly argue that binding of phosphorylated ubiquitin to parkin induces remote structural changes in the catalytic center of parkin (Fig. 4.6A, lower).

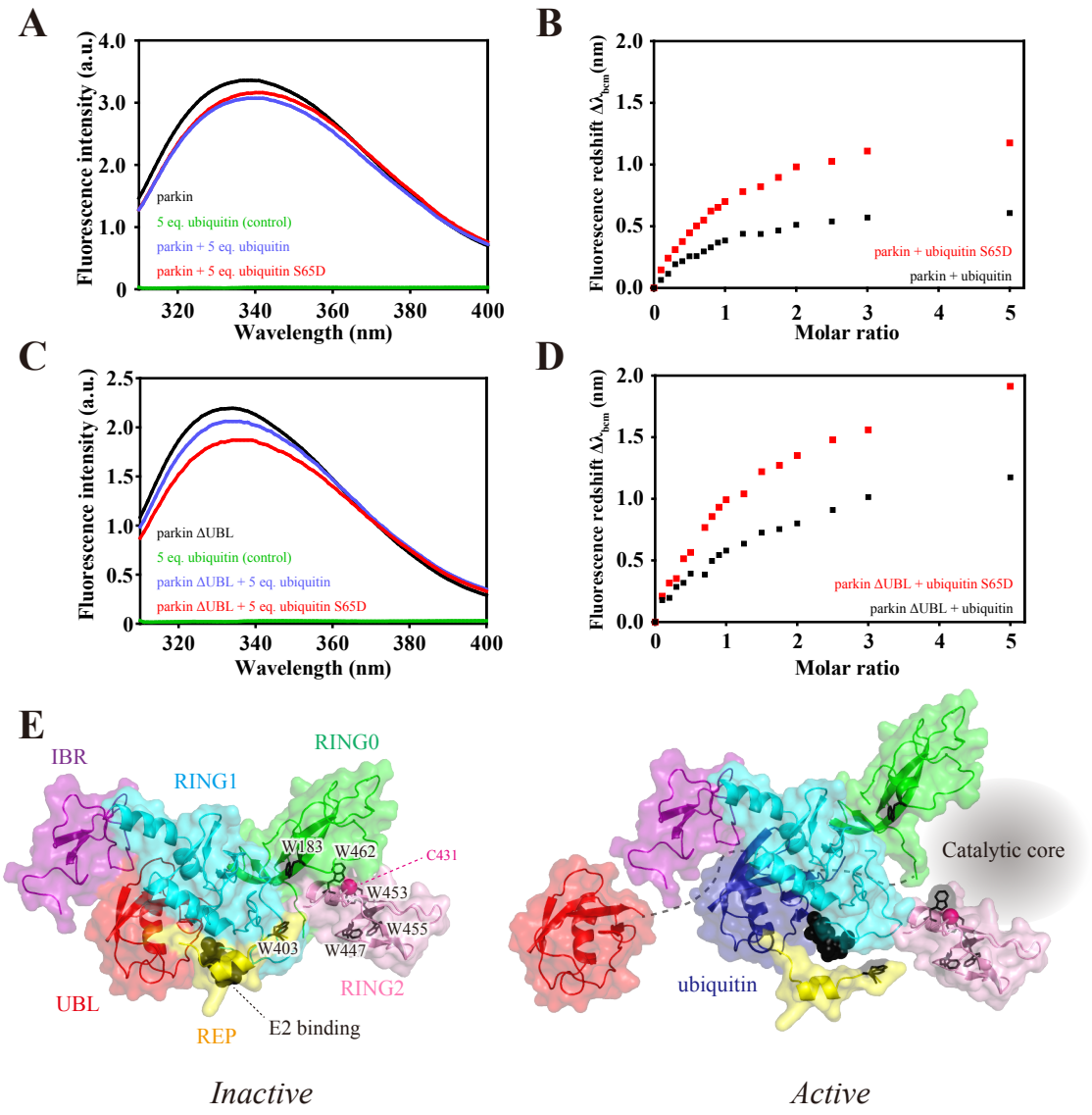

Figure 4.6: Phosphomimetic ubiquitin binding induces structural changes in the remote catalytic core of parkin. A, Changes in the tryptophan fluorescence emission spectrum of full-length parkin upon binding to (phosphomimetic) ubiquitin. B, Redshift of the barycentric mean λ_{bcm} of the fluorescence emission spectrum as a function of molar ratio (phosphomimetic) ubiquitin to parkin. C, Changes in the tryptophan emission spectrum of parkin ΔUBL upon binding to (phosphomimetic) ubiquitin. D, Redshift of the barycentric mean λ_{bcm} of the fluorescence emission spectrum as a function of molar ratio (phosphomimetic) ubiquitin to parkin ΔUBL. E, Left: distribution of tryptophan residues in rat parkin (auto-inhibited form; PDB ID 4K95; ref. [20]). With the exception of Trp^{97} (not observed in the crystal structure), all tryptophan residues of parkin are located in the vicinity of the catalytic core of parkin (Cys^{431} in the RING2 domain; pink sphere). Right: model for parkin activation by phosphorylated ubiquitin. Binding of phosphorylated ubiquitin to core parkin (blue) converts the closed structure of the remote catalytic center of parkin in a more open form as reflected by the partial solvent exposure of tryptophan residues.

4.5 Discussion

Two events are required to convert auto-inhibited parkin into an active enzyme *in vivo*: phosphorylation of Ser^{65} of the parkin UBL domain and phosphorylation of ubiquitin at Ser^{65} [17]. To date, the effect of neither phosphorylation effect is understood in molecular detail. On the one hand, crystal structures of parkin convincingly demonstrated that large structural rearrangements are necessary to allow the buried catalytic residue Cys^{431} to participate in the ubiquitylation reaction [20]. On the other hand, biochemical experiments have shown that the presence of phosphorylated ubiquitin is sufficient to release the latent E3 activity of parkin *in vitro* [71]. Nevertheless, how phosphorylated ubiquitin converts parkin in an active enzyme remains unexplained.

In this chapter, I showed that phosphomimetic ubiquitin undergoes competition with the parkin UBL domain for binding to core parkin. Strikingly, binding of phosphomimetic ubiquitin to parkin is enthalpically disfavored, but is rendered favorable by an increase in total entropy. In recent years, it has been elegantly demonstrated that the total entropy of binding (as determined by ITC; $-T\Delta S$) is dominated by the change in conformational entropy upon binding ($-T\Delta S_{conf}$) [76, 77, 78, 79, 80]. Thus, I interpret the binding of phosphomimetic ubiquitin to parkin as an event that raises the conformational entropy of parkin, which is in fine agreement with the idea that the auto-inhibited, rigid parkin is converted into a more dynamic conformation. Accordingly, I directly observed an opening in the catalytic core of parkin by monitoring the fluorescence redshift accompanying the solvent exposure of tryptophan residues in the vicinity of the catalytic core.

On the basis of these results, I propose a model in which phosphorylated ubiquitin is an allosteric activator of parkin *in trans*, whereas the N-terminal UBL domain of parkin adopts an antagonistic role *in cis* (Fig. 4.6A, lower). Similarly sophisticated auto-regulation mechanisms are typical for RBR ligases, as has been elegantly demonstrated for HOIP and HHARI [81, 82]. The crucial importance of RBR regulation has been stressed before [83]. A parkin-specific reason for the need for such a tight regulation is likely the fact that it recognizes an enormous number of different substrates marking these proteins for degradation [18]. Accordingly, parkin dysregulation is particularly dramatic as highlighted by the effect of the pathogenic mutations associated with autosomal-recessive parkinsonism [84]. I anticipate that this report will aid the understanding of parkin regulation, which is a necessary prerequisite for the development of therapeutic means to modulate parkin activity.

Chapter 5

Summary and General Conclusions

In the course of this thesis, specific examples of proteins involved in the biochemical pathway *autophagy* have been discussed: NBR1 (chapter 2), p62 (chapter 3) and parkin (chapter 4). These proteins have been studied with methods of structural biology and biophysics, such as nuclear magnetic resonance, isothermal titration calorimetry and fluorescence spectroscopy.

It has emerged that the function of all of these proteins is influenced by the intracellular signaling protein *ubiquitin*. Nevertheless, a closer look revealed that the role that ubiquitin plays in the regulation of these steps in the pathway of autophagy is intriguingly versatile.

Chapter 2 presented a rather classical example of ubiquitin binding. The autophagy receptor protein NBR1 simply binds to ubiquitin, which fulfills the role of a *signaling tag* to mark molecules that are destined for degradation by autophagy. A similar mechanism is used by p62 to recognize autophagic cargo via the ubiquitin tag. In the case of p62, however, ubiquitin binding is complicated owing to tight regulation by p62 phosphorylation and p62 homo-dimerization (chapter 3). Finally, chapter 4 described a completely different role of ubiquitin as an *allosteric activator* of an auto-inhibited E3-ubiquitin ligase.

Although these roles are truly complementary, ubiquitin uses a very similar surface to bind to its partners (NBR1, p62 and parkin) in these interactions: the Ile^{44} hydrophobic patch (section 1.3). Accordingly, a similar free energy term ΔG of ubiquitin binding is used by nature for quite different purposes: tight binding to achieve recognition of an autophagic substrate (chapters 2-3) or conversion of a rigid, inactive structure to active enzyme (chapter 4). The former process is driven by enthalpy, whereas the latter process is entropy-driven.

Ubiquitin is evolutionarily conserved to an unchallenged degree. For example, only 3 of the 76 amino acids vary from yeast to human [6]. In a sense, one may

think of ubiquitin as a protein whose *evolution has been completed* and its amino acid sequence is now fixed. Nevertheless, ubiquitin can still be varied by post-translational modifications such as phosphorylation to achieve exceptional functions (chapter 4).

We can therefore anticipate that the combination of ubiquitylation and phosphorylation and other post-translational modifications will further impact not only autophagy, but other biochemical pathways as well.

Appendix

Bibliography

[1] M. S. Forman, J. Q. Trojanowski, and V. M. Lee, "Neurodegenerative diseases: a decade of discoveries paves the way for therapeutic breakthroughs," *Nature Medicine*, vol. 10, no. 10, pp. 1055–1063, 2004.

[2] J. M. Berg, J. L. Tymoczko, and L. Stryer, *Biochemistry*. New York: W.H. Freeman, 2006.

[3] F. Crick, "Central dogma of molecular biology," *Nature*, vol. 227, no. 5258, pp. 561–563, 1970. 10.1038/227561a0.

[4] C. B. Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

[5] E. L. Greer, "Histone methylation: a dynamic mark in health, disease and inheritance," *Nat Rev Genet*, vol. 13, no. 5, pp. 343–357, 2012.

[6] M. Hochstrasser, "Origin and function of ubiquitin-like proteins," *Nature*, vol. 458, no. 7237, pp. 422–429, 2009. 10.1038/nature07958.

[7] D. Komander and M. Rape, "The ubiquitin code," *Annu Rev Biochem*, pp. 203–29, 2012.

[8] I. Dikic, S. Wakatsuki, and W. K. J., "Ubiquitin-binding domains: from structures to functions.," *Nat Rev Mol Cell Biol*, vol. 10, pp. 659–671, 2009.

[9] R. A. Nixon, "The role of autophagy in neurodegenerative disease," *Nat Med*, pp. 983–997, 2013.

[10] F. Chiti and C. M. Dobson, "Protein misfolding, functional amyloid, and human disease," *Annu Rev Biochem*, pp. 333–366, 2006.

[11] C. A. Ross and M. A. Poirier, "Protein aggregation and neurodegenerative disease," *Nat Med*, pp. S10–S17, 2004.

[12] J. M. Shulman, P. L. De Jager, and M. B. Feany, "Parkinson's disease: Genetics and pathogenesis," *Annu. Rev. Pathol. Mech. Dis.*, vol. 6, pp. 193–222, 2011.

[13] T. Hara, K. Nakamura, M. Matsui, A. Yamamoto, Y. Nakahara, R. Suzuki-Migishima, M. Yokoyama, K. Mishima, I. Saito, H. Okano, and N. Mizushima, "Suppression of basal autophagy in neural cells causes neurodegenerative disease in mice," *Nature*, vol. 441, pp. 885–889, 2006.

[14] M. Komatsu, S. Waguru, T. Chiba, S. Murata, J.-i. Iwata, I. Tanida, T. Ueno, M. Koike, Y. Uchiyama, E. Kominami, and K. Tanaka, "Loss of autophagy in the central nervous system causes neurodegeneration in mice," *Nature*, vol. 441, pp. 880–884, 2006.

[15] T. Kitada, S. Asakawa, N. Hattori, H. Matsumine, Y. Yamamura, S. Minoshima, M. Yokochi, Y. Mizuno, and N. Shimizu, "Mutations in the parkin gene cause autosomal recessive juvenile parkinsonism," *Nature*, vol. 392, pp. 605–608, 1998.

[16] I. E. Clark, M. W. Dodson, C. Jiang, J. H. Cao, J. R. Huh, J. H. Seol, S. J. Yoo, B. A. Hay, and M. Guo, "Drosophila pink1 is required for mitochondrial function and interacts genetically with parkin," *Nature*, vol. 441, pp. 1162–1166, 2006.

[17] F. Koyano, K. Okatsu, H. Kosako, Y. Tamura, E. Go, M. Kimura, Y. Kimura, H. Tsuchiya, H. Yoshihara, T. Hirokawa, T. Endo, E. A. Fon, J.-F. Trempe, Y. Saeki, K. Tanaka, and N. Matsuda, "Ubiquitin is phosphorylated by pink1 to activate parkin," *Nature*, 2014.

[18] S. A. Sarraf, M. Raman, V. Guarani-Pereira, M. E. Sowa, E. L. Huttlin, S. P. Gygi, and J. W. Harper, "Landscape of the parkin-dependent ubiquitylome in response to mitochondrial depolarization," *Nature*, vol. 496, pp. 372–376, 2013.

[19] J. N. Guzman, J. Sanchez-Padilla, D. Wokosin, J. Kondapalli, E. Ilijic, P. T. Schumacker, and D. J. Surmeir, "Oxidant stress evoked by pacemaking in dopamineric neurons is attenuated by dj-1," *Nature*, vol. 468, pp. 696–700, 2010.

[20] J.-F. Trempe, V. Sauve, K. Grenier, M. Seirafi, M. Y. Tang, M. Menade, S. Al-Abdul-Wahid, J. Krett, K. Wong, G. Kozlov, B. Nagar, E. A. Fon, and K. Gehring, "Structure of parkin reveals mechanisms for ubiquitin ligase activation," *Science*, vol. 340, pp. 1451–1455, 2013.

[21] D. Finley, "Recognition and processing of ubiquitin-protein conjugates by the proteasome," *Annu Rev Biochem*, pp. 477–513, 2009.

[22] T. Johansen and T. Lamark, "Selective autophagy mediated by autophagic adapter proteins.," *Autophagy*, vol. 7, pp. 279–296, 2011.

[23] C. Kraft, M. Peter, and K. Hofmann, "Selective autophagy: ubiquitin-mediated recognition and beyond," *Nat Cell Biol*, vol. 12, pp. 836–841, 2010.

[24] J. Long, T. P. Garner, M. J. Pandya, C. J. Craven, P. Chen, B. Shaw, M. P. Williamson, R. Layfield, and M. S. Searle, "Dimerisation of the UBA domain of p62 inhibits ubiquitin binding and regulates NF- κb signalling.," *J Mol Biol*, vol. 396, pp. 178–194, 2010.

[25] S. Isogai, D. Morimoto, K. Arita, S. Unzai, T. Tenno, J. Hasegawa, Y.-s. Sou, M. Komatsu, K. Tanaka, M. Shirakawa, and H. Tochio, "Crystal structure of the ubiquitin-associated (uba) domain of p62 and its interaction with ubiquitin.," *J Biol Chem*, vol. 286, pp. 31864–31874, 2011.

[26] J. Long, T. R. A. Gallagher, J. R. Cavey, P. W. Sheppard, S. H. Ralston, R. Layfield, and M. S. Searle, "Ubiquitin recognition by the ubiquitin-associated domain of p62 involves a novel conformational switch.," *J Biol Chem*, vol. 283, pp. 5427–5540, 2008.

[27] V. Kirkin, T. Lamark, Y.-S. Sou, G. Bjørkøy, J. L. Nunn, J.-A. Bruun, E. Shvets, D. G. McEwan, T. H. Clausen, P. Wild, I. Bilusic, J.-P. Theurillat, A. Øvervatn, T. Ishii, Z. Elazar, M. Komatsu, I. Dikic, and T. Johansen, "A role for nbr1 in autophagosomal degradation of ubiquitinated substrates," *Mol Cell*, vol. 33, pp. 505–516, 2009.

[28] T. Tenno, K. Fujiwara, H. Tochio, K. Iwai, E. H. Morita, H. Hayashi, S. Murata, H. Hiroaki, M. Sato, K. Tanaka, and M. Shirakawa, "Structural basis for distinct roles of lys63- and lys48-linked polyubiquitin chains," *Genes to Cells*, vol. 9, no. 10, pp. 865–875, 2004.

[29] F. Delagio, S. Grzesiek, G. W. Vuister, G. Zhu, J. Pfeifer, and A. Bax, "Nmrpipe: a multidimensional spectral processing system based on unix pipes," *J Biomol NMR*, vol. 6, pp. 277–293, 1995.

[30] N. Kobayashi, Y. Harano, N. Tochio, E. Nakatani, T. Kigawa, S. Yokoyama, S. Mading, E. L. Ulrich, J. L. Markley, H. Akutsu, and T. Fujiwara, "An automated system designed for large scale nmrdata deposition and annotation: application to over 600 assigned chemical shift data entries to the biomagresbank from the riken structural genomics/proteomics initiative internal database," *J Biomol NMR*, vol. 53, pp. 311–320, 2012.

[31] N. Kobayashi, J. Iwahara, S. Koshiba, T. Tomizawa, N. Tochio, P. Guentert, T. Kigawa, and S. Yokoyama, "Kujira, a package of integrated modules for

systematic and interactive analysis of nmr data directed to high-throughput nmr structure studies," *J Biomol NMR*, vol. 39, pp. 31–52, 2007.

[32] W. F. Vranken, W. Boucher, T. J. Stevens, R. H. Fogh, A. Pajon, M. Llinas, E. L. Ulrich, J. L. Markley, J. Ionides, and E. D. Laue, "The ccpn data model for nmr spectroscopy: development of a software pipeline," *Proteins*, vol. 59, pp. 687–696, 2005.

[33] Y.-S. Jung and M. Zweckstaetter, "Mars: robust automatic backbone assignment of proteins.," *J Biomol NMR*, vol. 30, pp. 11–23, 2004.

[34] P. Guentert, "Automated nmr structure calculation with cyana," *Methods Mol Biol*, vol. 278, pp. 353–378, 2004.

[35] Y. Shen, F. Delagio, G. Cornilescu, and A. Bax, "Talos+: a hybrid method for predicting protein backbone torsion angles from nmr chemical shifts," *J Biomol NMR*, vol. 44, pp. 213–223, 2009.

[36] K. Sugase, T. Konuma, J. C. Lansing, and P. E. Wright, "Fast and accurate fitting of relaxation dispersion data using the flexible software package glove," *J Biomol NMR*, vol. 56, pp. 275–283, 2013.

[37] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007.

[38] H. Valafar and J. H. Prestegard, "Redcat: a residual dipolar coupling analysis tool," *J Magn Reson*, vol. 167, pp. 228–241, 2004.

[39] Y.-X. Wang, J. L. Marquardt, P. Wingfield, S. J. Stahl, S. Lee-Huang, D. Torchia, and A. Bax, "Simultaneous measurement of ^{1}h,^{15}n, ^{1}h-^{13}c, and ^{15}n-^{13}c dipolar couplings in a perdeuterated 30-kda protein dissolved in a dilute liquid crystalline phase," *J Am Chem Soc*, vol. 120, pp. 7385–7386, 1998.

[40] M. Zweckstaetter, "Nmr: prediction of molecular alignment from structure using the pales software," *Nat Protoc*, vol. 3, pp. 679–690, 2008.

[41] C. Zwahlen, P. Legault, S. J. F. V. Vincent, J. Greenblatt, R. Konrat, and L. E. Kay, "Methods for measurement of intermolecular noes by multinuclear nmr spectroscopy: application to a bacteriophage λ n-peptide/boxb rna complex," *J Am Chem Soc*, vol. 119, pp. 6711–6721, 1997.

[42] S. J. de Vries, M. van Dijk, and A. M. J. J. Bonvin, "The haddock web server for data-driven biomolecular docking," *Nat Protoc*, vol. 5, pp. 883–897, 2010.

[43] S. J. Hubbard and J. M. Thornton, "Naccess," *University College, London*, 1993.

[44] L. Holm and P. Rosenstroem, "Dali server: conservation mapping in 3d," *Nucl Acids Res*, vol. 38, pp. 545–549, 2010.

[45] I. Eisuke and N. Mizushima, "p62 targeting to the autophagosome formation site requires self-oligomerization but not lc3 binding," *J Cell Biol*, vol. 192, pp. 17–27, 2011.

[46] D. Zhang, S. Raasi, and D. Fushman, "Affinity makes the difference: nonselective interaction of the uba domain of ubiquilin-1 with monomeric ubiquitin and polyubiquitin chains.," *J Mol Biol*, vol. 377, pp. 162–180, 2008.

[47] J. J. Sims, A. Haririnia, B. C. Dickinson, D. Fushman, and R. E. Cohen, "Avid interactions underlie the lys^{63}-linked polyubiquitin binding specificities observed for uba domains," *Nat Struct Mol Biol*, vol. 16, pp. 883–889, 2009.

[48] J. Cavanagh, W. J. Fairbrother, A. G. Palmer, M. Rance, and N. J. Skelton, *Protein NMR Spectroscopy*. Burlington, MA, USA: Elsevier Academic Press, 2007.

[49] S. Raasi, R. Varadan, D. Fushman, and C. M. Pickart, "Diverse polyubiquitin interaction properties of ubiquitin-associated domains," *Nat Struct Mol Biol*, vol. 12, pp. 708–714, 2005.

[50] A. Ohno, J.-G. Jee, K. Fujiwara, T. Tenno, N. Goda, H. Tochio, H. Kobayashi, H. Hiroaki, and M. Shirakawa, "Structure of the uba domain of dsk2p in complex with ubiquitin molecular determinants for ubiquitin recognition," *Structure*, vol. 13, pp. 521–532, 2005.

[51] S. E. Kaiser, B. E. Riley, T. A. Shaler, R. S. Trevino, C. H. Becker, H. Schulman, and R. R. Kopito, "Protein standard absolute quantification (psaq) method for the measurement of cellular ubiquitin pools," *Nat Methods*, vol. 8, pp. 691–696, 2011.

[52] B. E. Riley, S. E. Kaiser, T. A. Shaler, A. C. Y. Ng, T. Hara, M. S. Hipp, K. Lage, R. J. Xavier, K.-Y. Ryu, K. Taguchi, M. Yamamoto, K. Tanaka, M. Noboru, M. Komatsu, and R. R. Kopito, "Ubiquitin accumulation in autophagy-deficient mice is dependent on the nrf2-mediated stress response pathway: a potential role for protein aggregation in autophagic substrate selection," *J Cell Biol*, vol. 191, pp. 537–552, 2010.

[53] P. Peschard, G. Kozlov, T. Lin, I. A. Mirza, A. M. Berghuis, S. Lipkowitz, M. Park, and K. Gehring, "Structural basis for ubiquitin-mediated dimerization and activation of the ubiquitin protein ligase cbl-b," *Mol Cell*, vol. 27, pp. 474–485, 2007.

[54] J. M. Murphy, D. M. Korzhnev, D. F. Ceccarelli, D. J. Briant, A. Zarrine-Afsar, F. Sicheri, L. E. Kay, and T. Pawson, "Conformational instability of the mark3 uba domain compromises ubiquitin recognition and promotes interaction with the adjacent kinase domain," *Proc Natl Acad Sci U.S.A.*, vol. 104, pp. 14336–14341, 2007.

[55] G. Matsumoto, K. Wada, M. Okuno, M. Kurosawa, and N. Nukina, "Serine 403 phosphorylation of p62/sqstm1 regulates selective autophagic clearance of ubiquitinated proteins," *Mol Cell*, vol. 44, pp. 279–289, 2011.

[56] V. Kirkin, D. G. McEwan, I. Novak, and I. Dikic, "A role for ubiquitin in selective autophagy," *Mol Cell*, vol. 34, pp. 259–269, 2009.

[57] D. Komander, F. Reyes-Turcu, J. D. F. Licchesi, P. Odenwaelder, K. D. Willkinson, and D. Barford, "Molecular discrimination of structurally equivalent lys 63-linked and linear polyubiquitin chains," *EMBO Rep*, vol. 10, pp. 466–473, 2009.

[58] C. A. Castaneda, T. R. Kashyap, M. A. Nakasone, S. Krueger, and D. Fushman, "Unique structural, dynamical, and functional properties of k11-linked polyubiquitin chains," *Structure*, vol. 21, pp. 1168–1181, 2013.

[59] T. H. Clausen, T. Lamark, P. Isakson, K. D. Finley, K. B. Larsen, A. Brech, A. Øvervatn, H. Stenmark, G. Bjørkøy, A. Simonsen, and T. Johansen, "p62/sqstm1 and alfy interact to facilitate the formation of p62 bodies/alis and their degradation by autophagy," *Autophagy*, vol. 6, pp. 330–344, 2010.

[60] E. Kuusisto, A. Salminen, and I. Alafuzoff, "Ubiquitin-binding protein p62 is present in neuronal and glial inclusions in human tauopathies and synucleinopathies," *Neuroreport*, vol. 12, no. 10, pp. 2085–2090, 2001.

[61] K. Zatloukal, C. Stumptner, A. Fuchsbichler, H. Heid, M. Schnoelzer, L. Kenner, R. Kleinert, M. Prinz, A. Aguzzi, and H. Denk, "p62 is a common component of cytoplasmic inclusions in protein aggregation diseases," *The American journal of pathology*, vol. 160, no. 1, pp. 255–263, 2002.

[62] R. J. Bothe, E. N. Nikolova, C. Eichhorn, J. Chugh, A. L. Hansen, and H. M. Al-Hashimi, "Characterizing rna dynamics at atomic resolution using solution-state nmr spectroscopy," *Nature Methods*, vol. 8, p. 919–931, 2011.

[63] H. Y. Carr and E. M. Purcell, "Effects of Diffusion on Free Precession in Nuclear Magnetic Resonance Experiments," *Physical Review*, vol. 94, pp. 630–638, May 1954.

[64] S. Meiboom and D. Gill, "Modified Spin-Echo Method for Measuring Nuclear Relaxation Times," *Review of Scientific Instruments*, vol. 29, pp. 688–691, Aug. 1958.

[65] D. M. Korzhnev, V. Y. Orekhov, and L. E. K. Kay, "Off-resonance $r_1\rho$ nmr studies of exchange dynamics in proteins with low spin-lock fields: An application to a fyn sh3 domain," *J. Am. Chem. Soc.*, vol. 127, pp. 713–721, 2005.

[66] H. Motulsky and A. Christopoulos, *Fitting Models to Biological Data Using Linear and Nonlinear Regression.* New York: Oxford University Press, 2004.

[67] J. J. Helmus and C. P. Jaroniec, "Nmrglue: An open source python package for the analysis of multidimensional nmr data," *J Biomol NMR*, vol. 55, pp. 355–367, 2013.

[68] M. Iguchi, Y. Kujuro, K. Okatsu, F. Koyano, H. Kosako, M. Kimura, N. Suzuki, S. Uchiyama, K. Tanaka, and N. Matsuda, "Parkin-catalyzed ubiquitin-ester transfer is triggered by pink1-dependent phosphorylation," *J Biol Chem*, vol. 288, pp. 22019–22032, 2013.

[69] L. A. Kane, M. Lazarou, A. I. Fogel, Y. Li, K. Yamano, S. A. S. Sarraf, S. Banerjee, and R. J. Youle, "Pink1 phosphorylates ubiquitin to activate parkin e3 ubiquitin ligase activity," *Journal of Cell Biology*, vol. 205, pp. 143–153, 2014.

[70] T. R. Caulfield, F. C. Fiesel, E. L. Moussaud-Lamodiere, D. F. Dourado, S. C. Flores, and W. Springer, "Phosphorylation by pink1 releases the ubl domain and initializes the conformational opening of the e3 ubiquitin ligase parkin," 2014.

[71] T. Wauer, K. N. Swatek, J. L. Wagstaff, C. Gladkova, J. N. Pruneda, M. A. Michel, M. Gersch, C. M. Johnson, S. M. Freund, and D. Komander, "Ubiquitin ser65 phosphorylation affects ubiquitin structure, chain assembly and hydrolysis," *The EMBO journal*, vol. 34, no. 3, pp. 307–325, 2015.

[72] S. Vijay-Kumar, C. E. Bugg, and W. J. Cook, "Structure of ubiquitin refined at 1.8 åresolution," *Journal of molecular biology*, vol. 194, no. 3, pp. 531–544, 1987.

[73] D. Komander and M. Rape, "The ubiquitin code," *Annu Rev Biochem*, pp. 203–29, 2012.

[74] V. K. Chaugule, L. Burchell, K. R. Barber, A. Sidhu, S. J. Leslie, G. S. Shaw, and H. Walden, "Autoregulation of parkin activity through its ubiquitin-like domain," *EMBO Journal*, vol. 30, pp. 2853–2867, 2011.

[75] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[76] S.-R. Tzeng and C. G. Kalodimos, "Protein activity regulation by conformational entropy," *Nature*, vol. 488, no. 7410, pp. 236–240, 2012.

[77] K. K. Frederick, M. S. Marlow, K. G. Valentine, and A. J. Wand, "Conformational entropy in molecular recognition by proteins," *Nature*, vol. 448, no. 7151, pp. 325–329, 2007.

[78] M. S. Marlow, J. Dogan, K. K. Frederick, K. G. Valentine, and A. J. Wand, "The role of conformational entropy in molecular recognition by calmodulin," *Nature chemical biology*, vol. 6, no. 5, pp. 352–358, 2010.

[79] A. J. Wand, "Dynamic activation of protein function: a view emerging from nmr spectroscopy," *Nature Structural & Molecular Biology*, vol. 8, no. 11, pp. 926–931, 2001.

[80] S.-R. Tzeng and C. G. Kalodimos, "Dynamic activation of an allosteric regulatory protein," *Nature*, vol. 462, no. 7271, pp. 368–372, 2009.

[81] J. J. Smit, D. Monteferrario, S. M. Noordermeer, W. J. van Dijk, B. A. van der Reijden, and T. K. Sixma, "The e3 ligase hoip specifies linear ubiquitin chain assembly through its ring-ibr-ring domain and the unique ldd extension," *The EMBO journal*, vol. 31, no. 19, pp. 3833–3844, 2012.

[82] D. M. Duda, J. L. Olszewski, J. P. Schuermann, I. Kurinov, D. J. Miller, A. Nourse, A. F. Alpi, and B. A. Schulman, "Structure of hhari, a ring-ibr-ring ubiquitin ligase: autoinhibition of an ariadne-family e3 and insights into ligation mechanism," *Structure*, vol. 21, no. 6, pp. 1030–1041, 2013.

[83] K. K. Dove and R. E. Klevit, "Ring-between-rings—keeping the safety on loaded guns," *The EMBO journal*, vol. 31, no. 19, pp. 3792–3794, 2012.

[84] C. B. Lücking, A. Dürr, V. Bonifati, J. Vaughan, G. De Michele, T. Gasser, B. S. Harhangi, G. Meco, P. Denèfle, N. W. Wood, *et al.*, "Association between early-onset parkinson's disease and mutations in the parkin gene," *New England Journal of Medicine*, vol. 342, no. 21, pp. 1560–1567, 2000.

Amaterasu Source Code

Amaterasu GUI: *ama_gui.py*

```
import os
import sys
from PyQt4 import QtGui, QtCore

import ama_init
import ama_screening
import ama_visualize



class Window(QtGui.QWidget):

    def __init__(self):
        super(Window, self).__init__()
        self.initUI()

    def initUI(self):

        # __ Get data location [folder] __

        select_data_button = QtGui.QPushButton('Select data', self)
        select_data_button.setToolTip('Select the data
            __directory__ that you want to process. This is the
            directory containing the NMR data files (ser, acqus,
            ...). It is not the dat subdirectory inside the data
            directory.')
        select_data_button.resize(select_data_button.sizeHint())
        select_data_button.clicked.connect(self.
            on_clicked_select_data_directory)

        # __ Screen for dispersion __

        screen_for_dispersion_button = QtGui.QPushButton('Screen
```

```
    for dispersion', self)
screen_for_dispersion_button.setToolTip('Perform a
    screening run on the protein to do a quick check on
    which residues are likely to show relaxation dispersion
    and thus are worthy of acquisition of a whole dataset.')
screen_for_dispersion_button.clicked.connect(self.
    start_screening)

# __ User-set intensity ratio threshold __

threshold_label = QtGui.QLabel('Threshold: ')
threshold_qle = QtGui.QLineEdit(self)
threshold_qle.textChanged[str].connect(self.
    on_threshold_changed)
threshold_qle.setToolTip('Define threshold for the
    screening step. This setting is not required for
    functions other than the [Screen for dispersion].')

# __ User-defined p0 / p1 values __

p0_label = QtGui.QLabel('User defined p0 and p1 [optional]
    ')
p0_qle = QtGui.QLineEdit(self)
p0_qle.textChanged[str].connect(self.on_p0_changed)
p0_qle.setToolTip('User-defined p0 phase correction [
    optional]. You may supply pre-determined p0 and p1 phase
     correction values. This will make Amaterasu skip the
    automatic phase correction step and make the run faster.
    ')

# p1_label = QtGui.QLabel('p1: ')
p1_qle = QtGui.QLineEdit(self)
p1_qle.textChanged[str].connect(self.on_p1_changed)
p1_qle.setToolTip('User-defined p1 phase correction [
    optional]. You may supply pre-determined p0 and p1 phase
     correction values. This will make Amaterasu skip the
    automatic phase correction step and make the run faster.
    ')

# __ Process dataset __

process_all_data_button = QtGui.QPushButton('Process
    dataset', self)
process_all_data_button.setToolTip('Peform automatic FT,
    phasing, baseline correction and intensity extraction.
    This will create an intensity report file that can be
    subjected to curve fitting to a relaxation dispersion
```

```
    model by the GLOVE software package.')
process_all_data_button.resize(process_all_data_button.
    sizeHint())
process_all_data_button.clicked.connect(self.proc_all)

# __ Peak visualization (requires nmrglue) __

visualize_dispersion_button = QtGui.QPushButton('Visualize'
    , self)
visualize_dispersion_button.clicked.connect(self.visualize)
visualize_dispersion_button.setToolTip('Make plots of all
    of the 1D spectra of the experiment grouped by residue.
    Requires prior execution of the [process dataset]
    function.')

# __ Quit button __

quit_button = QtGui.QPushButton('Quit', self)
quit_button.resize(quit_button.sizeHint())
quit_button.clicked.connect(QtCore.QCoreApplication.
    instance().quit)

# __ Grid to align the buttons in the window __

grid = QtGui.QGridLayout()
grid.setSpacing(10)

grid.addWidget(select_data_button, 2, 0)
grid.addWidget(screen_for_dispersion_button, 3, 0)
grid.addWidget(threshold_label, 3, 1)
grid.addWidget(threshold_qle, 3, 2)
grid.addWidget(process_all_data_button, 4, 0)
grid.addWidget(visualize_dispersion_button, 4, 1)
grid.addWidget(quit_button, 4, 2)

grid.addWidget(p0_label, 5, 0)
grid.addWidget(p0_qle, 5, 1)
grid.addWidget(p1_qle, 5, 2)


self.setLayout(grid)
self.setGeometry(200, 400, 350, 180)
self.setWindowTitle('Amaterasu')

self.show()
```

```
def on_threshold_changed(self, text):
    global threshold
    try:
            threshold = float(text)
    except ValueError:
            pass

def on_p0_changed(self, text):
    global p0
    try:
            p0 = float(text)
    except ValueError:
            pass

def on_p1_changed(self, text):
    global p1
    try:
            p1 = float(text)
    except ValueError:
            pass

def on_clicked_select_data_directory(self):
    global data_directory
    data_directory = QtGui.QFileDialog.getExistingDirectory(
       None, 'Select dataset folder', './', QtGui.QFileDialog.
       ShowDirsOnly)
    print "\nData directory selected: ", data_directory

def start_screening(self):
            ama_bin = os.getcwd() + '/bin'
            p0_exists = 'p0' in locals() or 'p0' in globals()

            if p0_exists == True:
                    ama_screening.go_screening(str(
                       data_directory), str(threshold), ama_bin
                       , 1, p0, p1)
            else:
                    ama_screening.go_screening(str(
                       data_directory), str(threshold), ama_bin
                       , 0, 0, 0)

            os.chdir(initial_dir)

def visualize(self):
            ama_visualize.visualize(str(data_directory))
            os.chdir(initial_dir)
```

```
    def proc_all(self):
                p0_exists = 'p0' in locals() or 'p0' in globals()

                # __ User set p0 / p1 values __

                if p0_exists == True:
                        ama_init.fox_go(str(data_directory), 1, p0,
                            p1) # user defined p0, p1
                else:
                        ama_init.fox_go(str(data_directory), 0, 0,
                           0)   # automatic phasing

                os.chdir(initial_dir)

def main():

    global initial_dir
    initial_dir = os.getcwd()

    app = QtGui.QApplication(sys.argv)
    ex = Window()
    sys.exit(app.exec_())


if __name__ == '__main__':
    main()
```

Amaterasu Screening for Dispersion: *ama_screening.py*

```
import os

import ama_findphase
import ama_intensity
import ama_processing
import ama_ft

def go_screening(data_location, threshold, ama_bin,
    user_defined_p0_p1, p0, p1):

        # __ Preparation  __
        ver = '1.0'
        threshold = float(threshold)

        print "----------------------------"
        print "Amaterasu version ", ver
        print "----------------------------"
        print "Location of Amaterasu files: ", ama_bin
        print "Location of data to process: ", data_location
        print "Intensity ratio threshold  : ", threshold
        print "Phasing (0: auto / 1: user): ", user_defined_p0_p1
        print "User-specified           p0: ", p0
        print "                         p1: ", p1
        print "----------------------------"
        print "Phasing datasets ..."

        datadir = data_location + "/dat"
        os.chdir(datadir)

        # __ Create arrays for data filenames __
        datasets = []    # strong spin-lock
        partners = []    # weak spin-lock

        for i in os.listdir(os.getcwd()):
                if i.endswith(".dat"):
                        aa = i.replace("test", "")
                        bb = aa.replace("dat", "")
                        cc = float(bb)
                        dd = int(cc)

                        # __ Discriminate odd and even numbers in
                            filenames __
                        if (dd % 2) == 0:
                                datasets.append(i)
                        else:
```

```
                        partners.append(i)
                        continue
                continue
        else:
                continue
os.chdir("../..")


# __ Phase correction for filenames with even numbers (
    strong spin lock) __
final_p0 = []
final_p1 = []

progress_counter = 0
progress_end = float(len(datasets))

for i in datasets:
        if user_defined_p0_p1 == 0:
                # __ Find p0 for strong spin-lock dataset
                    __
                this_p0 = ama_findphase.find_phase(i,
                    data_location, ama_bin)
                this_p1 = 0
                final_p0.append(this_p0)
                final_p1.append(0)
        elif user_defined_p0_p1 == 1:
                # __ Use user-specificed phase corrections
                    __
                this_p0 = p0
                this_p1 = p1
                final_p0.append(this_p0)
                final_p1.append(this_p1)
        else:
                pass

        # __ Calculate progress for command line progress
            bar __
        progress_counter = progress_counter + 1
        percent = float(progress_counter / progress_end)
        ama_processing.update_progress(percent)

# __ Prepare output __
output_filename = '../output/dispersion_screening_output_'
    + (data_location.rsplit('/', 1)[1]) + '.txt'
f = open(output_filename,'w')
header = 'Resonance \tweak spin-lock \t\tstrong spin-lock \
    t ratio \t\t p0\t\t p1\t\tcomment\n'
```

```
f.write(header)


# __ Acquire and compare the intensities of weak and strong
    spin-lock data __
for i in range(len(datasets)):
        # __ Strong spin-lock / high intensity __
        strong_spin_lock_intensity, X_ppm = ama_intensity.
           get_intensity_ppm(final_p0[i], final_p1[i],
           datasets[i], data_location, ama_bin)

        # __ Weak spin-lock / low intensity __
        weak_spin_lock_intensity, Y_ppm = ama_intensity.
           get_intensity_ppm(final_p0[i], final_p1[i],
           partners[i], data_location, ama_bin)

        # __ Retain FT data for analysis __
        ama_ft.apply_phasing(final_p0[i], final_p1[i],
           datasets[i], data_location, ama_bin)
        ama_ft.apply_phasing(final_p0[i], final_p1[i],
           partners[i], data_location, ama_bin)

        if q > 0:
                ratio = weak_spin_lock_intensity /
                   strong_spin_lock_intensity   # intensity
                    ratio weak / strong spin-lock
        else:
                ratio = 999             # debug to avoid
                   division by 0

        # __ Output  __

        if ratio > 0:
                if ratio < threshold:
                        reportstring = "Resonance " +   str
                           (i + 1) + "\t" + \
                                   partners[i] + "\t /
                                       \t" + \
                                   datasets[i] + "\t\t
                                       " + \
                                   str('% 3.2f'  %
                                      float(ratio)) +
                                      "\t\t" + \
                                   str('% 3.1f'  %
                                      float(final_p0[i
                                      ])) + "\t\t" + \
                                   str('% 3.1f'  %
```

```
                                    float(final_p1[i
                                    ])) + "\t\t" + \
                                "*** R1rho-
                                    dispersion! ***\
                                    n"
            elif ratio > threshold:
                    if ratio < 1.2:
                            reportstring = "Resonance "
                                +   str(i + 1) + "\t" +
                                \
                                        partners[i]
                                            + "\t /
                                            \t" + \
                                        datasets[i]
                                            + "\t\t
                                           " + \
                                        str('% 3.2f
                                           '  %
                                           float(
                                           ratio))
                                           + "\t\t"
                                            + \
                                        str('% 3.1f
                                           '  %
                                           float(
                                           final_p0
                                           [i])) +
                                           "\t\t" +
                                            \
                                        str('% 3.1f
                                           '  %
                                           float(
                                           final_p1
                                           [i])) +
                                           "\t\t" +
                                            \
                                        "\n"
                    if ratio > 1.2:
                            reportstring = "Resonance "
                                +   str(i + 1) + "\t" +
                                \
                                        partners[i]
                                            + "\t /
                                            \t" + \
                                        datasets[i]
                                            + "\t\t
                                           " + \
```

```
                                                    str('% 3.2f
                                                        '  %
                                                        float(
                                                        ratio))
                                                        + "\t\t"
                                                         + \
                                                    str('% 3.1f
                                                        '  %
                                                        float(
                                                        final_p0
                                                        [i])) +
                                                        "\t\t" +
                                                         \
                                                    str('% 3.1f
                                                        '  %
                                                        float(
                                                        final_p1
                                                        [i])) +
                                                        "\t\t" +
                                                         \
                                                    "__ Unusual
                                                         ratio.
                                                        Please
                                                        check
                                                        spectra.
                                                         __\n"
                        else:
                                continue
                else:
                        reportstring = "Resonance " +   str(i + 1)
                            + "\t" + \
                                        partners[i] + "\t / \t" + \
                                        datasets[i] + "\t\t" + \
                                        str('% 3.2f'  % float(ratio)
                                            ) + "\t\t" + \
                                        str('% 3.1f'  % float(
                                            final_p0[i])) + "\t\t" +
                                            \
                                        str('% 3.1f'  % float(
                                            final_p1[i])) + "\t\t" +
                                            \
                                        "!!! No peak could be found
                                            .\n"
                f.write(reportstring)
        f.close()

        print "\nScreening completed. Output written to: ",
```

```
output_filename
```

Amaterasu Processing Initialization: *ama_init.py*

```
import os
import ama_processing

def fox_go(data_location, user_defined_p0_p1, p0, p1):

        # __ Remember location of Amaterasu source files __
        ama_bin = os.getcwd() + '/bin'

        # __ Start processing __
        ama_processing.process_datapath(data_location, ama_bin,
           user_defined_p0_p1, p0, p1)
```

Amaterasu Input Data Reader Module: *ama_read_data.py*

```
import os

def read_input_data_structure(datafolder):

        # __ Read acqus (acquisition parameters) __
        myfile = datafolder + "/acqus"

        # __ Read TopSpin Version __
        f = open(myfile,'r')

        for line in f:
                if "##TITLE" in line:
                        columns = line.split()
                        ver = columns[5]
                        print "Data acquired by Bruker TopSpin
                           version: ", ver
        f.close()

        # __ Read BF3 __
        f = open(myfile,'r')

        for line in f:
                if "##$BF3" in line:
                        ac = line.split()
        BF3 = float(ac[1])

        f.close()

        # __ Read spin lock length (p25) __
        # __ Read 15N hard pulse length (p21) __
        # __ Read 15N hard pulse power (plw3) __
        with open(myfile) as fh:
                for line in fh:
                        if "##$P=" in line:
                                if ver == '3.2':
                                        for i in range(1):
                                                next_line = next(fh
                                                   )
                                                N_hard_pulse_length
                                                    = float(
                                                   next_line.split
                                                   ()[21])
                                                spin_lock_length =
                                                   float(next_line.
                                                   split()[25]) / 1
```

```
                                        E6
                        elif ver =='3.1':
                                for i in range(2):
                                        next_line = next(fh
                                            )
                                        N_hard_pulse_length
                                             = float(
                                            next_line.split
                                            ()[1])
                                        spin_lock_length =
                                            float(next_line.
                                            split()[5]) / 1
                                            E6
                if "##$PLW=" in line:
                        for i in range(1):
                                next_line = next(fh)
                                N_hard_pulse_power_W =
                                    float(next_line.split()
                                    [3])

# __ Read fq1list (resonance frequencies, ppm) __
myfile = datafolder + "/fq1list"
f = open(myfile,'r')
Skip_Header = f.readline() # 1 line to skip

Input_Peaks = []

for line in f:
        columns = line.split()
        Input_Peaks.append(columns[0])
f.close()

# __ Read valist (spinlock power values, dB) __
myfile = datafolder + "/valist"
f = open(myfile,'r')
Skip_Header = f.readline() # 1 line to skip

Input_Power = []

for line in f:
        columns = line.split()
        Input_Power.append(columns[0])
f.close()


# __ Number of peaks * number of spin lock powers should
   match the number of FID files __
```

```
Expected_FIDs = len(Input_Peaks) * len(Input_Power)

print "----------------------------"
print "Reading data ..."
print "----------------------------"
print "Data location:\t\t\t", datafolder
print "Number of resonances:\t\t", len(Input_Peaks)
print "Number of spin-lock durations:\t", len(Input_Power)
print "Number of FIDs expected:\t", Expected_FIDs


# __ Read vclist (distinguish reference data from data with
    actual spin-lock applied) __
# __ Flag is 0 (reference) or 1 (spin-lock applied) __
myfile = datafolder + "/vclist"
f = open(myfile,'r')

Reference_Flag = []

for line in f:
        columns = line.split()
        Reference_Flag.append(int(columns[0]))
f.close()

# __ Check FID datafiles in the /dat/ directory __
# Store their filenames in array datasets
datadir = datafolder + "/dat"
os.chdir(datadir)

datasets = []

for i in sorted(os.listdir(os.getcwd())):
        if i.endswith(".dat"):
                datasets.append(i)
                continue
        else:
                continue

#  __ Necessary debug for large datasets __
#  For larger datasets (> test1000.dat) the python array
   mixes up the
#  correct numbering of the data.
#  To correct the numbering:
filtered = []

for v in datasets:
        a = v.replace("test", "")
```

```
        b = a.replace(".dat", "")
        filtered.append(b)

z = sorted(filtered, key=int)    # sort filename numbers as
    ascending integers

datasets = []

for v in z:
        a = 'test' + v + '.dat'
        datasets.append(a)

os.chdir("../..")

print "Number of FIDs found:\t\t", len(datasets), "\n"
print "---------------------------"

Number_Peaks            = len(Input_Peaks)
Number_Power_Values     = len(Input_Power)

return datasets, Reference_Flag, Number_Power_Values,
    Number_Peaks, Input_Peaks, Input_Power, BF3,
    spin_lock_length, N_hard_pulse_length,
    N_hard_pulse_power_W
```

Amaterasu Processing for Full Dataset: *ama_processing.py*

```
import numpy as np
import sys

import ama_read_data
import ama_findphase
import ama_output
import ama_intensity

import ama_ft

ver = '1.0'

def process_datapath(data_location, ama_bin, user_defined_p0_p1, p0
    , p1):

        print "----------------------------"
        print "Amaterasu version ", ver
        print "----------------------------"
        print "Location of Amaterasu files : ", ama_bin
        print "Location of data to process : ", data_location
        print "Phasing (0: auto / 1: user) : ", user_defined_p0_p1
        print "User-specified            p0: ", p0
        print "                          p1: ", p1

        # __ Read data __

        fid_filenames, Reference_Flag, Number_Power_Values,
            Number_Peaks, Expected_ppm, Power_Values_valist, BF3,
            spin_lock_length, N_hard_pulse_length,
            N_hard_pulse_power_W = ama_read_data.
            read_input_data_structure(data_location)

        N_hard_pulse_power_dB = 10 * np.log10(N_hard_pulse_power_W)
                # Watt to dB

        print "BF3 [MHz]                    :", BF3
        print "90-pulse duration (15N)[us] :", N_hard_pulse_length
        print "90-pulse power (15N)    [W] :", N_hard_pulse_power_W
        print "90-pulse power (15N)   [dB] :",
            N_hard_pulse_power_dB
        print "Spin-lock duration (15N)[s] :", spin_lock_length

        # __ Discriminate filenames corresponding to reference (no
            spin lock) and spin-locked data __
        reference_fid_filenames = []
```

```
data_fid_filenames      = []

i = 0   # count over entries in valist  [spin-lock powers]
j = 0   # count over datasets            [test%03d.dat]

for v in fid_filenames:
        if int(Reference_Flag[i]) == 0:
                reference_fid_filenames.append(
                   fid_filenames[j])
        elif int(Reference_Flag[i]) == 1:
                data_fid_filenames.append(fid_filenames[j])
        else:
                continue

        j = j + 1

        if i == (Number_Power_Values - 1):
                i = 0
        else:
                i = i + 1


# __ Determine p0 phase correction for the ==reference==
   dataset __
ref_p0 = []
reference_p1 = []

progress_counter= 0
progress_end    = float(len(reference_fid_filenames))

Reference_Intensities = np.empty((0,1),float)

print "\nPhasing reference spectra ..."

for i in reference_fid_filenames:
        # __ Find p0 for reference dataset __
        if user_defined_p0_p1 == 0:
                this_p0 = ama_findphase.find_phase(i,
                   data_location, ama_bin)
                this_p1 = 0
        # __ or: use user-supplied p0 p1 __
        elif user_defined_p0_p1 == 1:
                this_p0 = p0
                this_p1 = p1
        else:
                pass
        ref_p0.append(this_p0)
```

```
            reference_p1.append(this_p1)

            # __ Progress bar __
            progress_counter= progress_counter + 1
            percent         = float(progress_counter /
                progress_end)

            update_progress(percent)

            # __ Get reference intensity __
            this_reference_intensity, ref_ppm = ama_intensity.
                get_intensity_ppm(this_p0, this_p1, i,
                data_location, ama_bin)
            Reference_Intensities = np.vstack([
                Reference_Intensities, this_reference_intensity
                ])

    # __ Retain both reference data (A and B) for GLOVE __
    Reference_Intensities_A = np.empty((0,1),float)
    Reference_Intensities_B = np.empty((0,1),float)

    # __ Since 2 reference data were acquired, average their p0
        values __
    reference_p0_avg= []
    reference_p1    = reference_p1[0::2]    # p1 is not
        determined, thus not averaged. Array size is thus twice
        as large as p0. Resize.

    counter = 0

    for i in range(0, Number_Peaks):
            this_average_p0 = float(float(ref_p0[counter]) +
                float(ref_p0[counter + 1])) / 2
            this_average_int = float(float(
                Reference_Intensities[counter] +
                Reference_Intensities[counter + 1])) / 2
            reference_p0_avg.append(this_average_p0)

            # __ Retain both reference intensity data for
                analysis in glove __
            this_ref_A = float(Reference_Intensities[counter])
            this_ref_B = float(Reference_Intensities[counter +
                1])
            Reference_Intensities_A = np.vstack([
                Reference_Intensities_A, this_ref_A])
            Reference_Intensities_B = np.vstack([
                Reference_Intensities_B, this_ref_B])
```

```
        counter = counter + 2

# __ Get peak intensities of all spectra __
# 1. Loop over all (non-reference) datasets
# 2. Apply phase corrections
# 3. Get peak intensities

Final_Intensities       = np.empty((0,1),float)
Final_X_ppm             = np.empty((0,1),float)
Final_p0                = np.empty((0,1),float)

Number_Power_Values = Number_Power_Values - 2           #
   Adjust for reference power entries

q = 0   # counter for data files

# 1. (i) Loop over all resonances
# 2. (j) Loop over all spin-lock power values

print "\n\nProcessing datasets ..."

for i in range (0, Number_Peaks):
        for j in range (0, Number_Power_Values):
                update_progress(q / float(len(
                   data_fid_filenames)))

                # __ Use determined p0 to get peak
                   intensity __
                this_intensity, this_ppm = ama_intensity.
                   get_intensity_ppm(reference_p0_avg[i],
                   reference_p1[i], data_fid_filenames[q],
                   data_location, ama_bin)
                q = q + 1       # counter to get datafile
                   right

                # __ Append data to final arrays for output
                    __
                Final_p0                = np.vstack([
                   Final_p0, float(reference_p0_avg[i])])
                Final_Intensities       = np.vstack([
                   Final_Intensities, float(this_intensity)
                   ])
                Final_X_ppm             = np.vstack([
                   Final_X_ppm, float(this_ppm)])

# __ Write output file __
```

```
        ama_output.write_output(Number_Peaks, Number_Power_Values,
            Final_Intensities, Reference_Intensities_A,
            Reference_Intensities_B, data_location, Final_X_ppm,
            data_fid_filenames, Final_p0, Expected_ppm,
            Power_Values_valist, BF3, spin_lock_length,
            N_hard_pulse_length, N_hard_pulse_power_dB)



def update_progress(percent):
        bar_length = 20
        hashes = '#' * int(round(percent * bar_length))
        spaces = ' ' * (bar_length - len(hashes))
        sys.stdout.write("\rPercent: [{0}] {1}%".format(hashes +
            spaces, int(round(percent * 100))))
        sys.stdout.flush()
```

Amaterasu Automatic Phase Correction Module: *ama_findphase.py*

```
import numpy as np
import ama_ft
import ama_intensity

def find_phase(filename, data_location, ama_bin):

        # __ Preparation __
        intensities = np.empty((0,1),float)
        refined_intensities = np.empty((0,1),float)

        phases = []
        refined_phases = []

        # __ Find best value of p0 __
        guess = [0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300,
            330] # Initial guess

        for i in guess:
                # __ FT with new value of p0 __
                trialintensity, trial_ppm = ama_intensity.
                    get_intensity_ppm(i, 0, filename, data_location,
                     ama_bin)
                # __ Store phases and intensities __
                phases.append(i)
                intensities = np.vstack([intensities,
                    trialintensity])

        # __ Find/recall phase corresponding to maximal peak
            intensity __
        maximum_intensity = np.amax(intensities)

        for l in range(len(phases)):
                deviation = intensities[l] - maximum_intensity
                if int(deviation) == 0:
                        initial_guess = phases[l]

        # __ Refine initial guess: browse +/- 16 degrees from
            initial guess __
        for i in range((initial_guess - 16), (initial_guess + 16)):
                # __ FT with new value of p0 __
                trialintensity, trial_ppm = ama_intensity.
                    get_intensity_ppm(i, 0, filename, data_location,
                     ama_bin)
```

```
            # __ Store phases and intensities __
            refined_phases.append(i)
            refined_intensities = np.vstack([
               refined_intensities, trialintensity])

    # __ Find/recall phase corresponding to maximal peak
       intensity __
    maximum_intensity = np.amax(refined_intensities)

    for l in range(len(refined_phases)):
                    deviation = refined_intensities[l] -
                       maximum_intensity
                    if int(deviation) == 0:
                                    final_phase =
                                       refined_phases[l] # best
                                        p0
                                    ama_ft.apply_phasing(
                                       final_phase, 0, filename
                                       , data_location, ama_bin
                                       )  # FT with correct p0
    return final_phase
```

Amaterasu Peak Intensity Extraction: *ama_intensity.py*

```
import os
import numpy as np

def get_intensity_ppm(p, p1, filename, data_location, ama_bin):

        # __ Prepare FT script for nmrpipe __
        file_in  = ama_bin       + '/ft_1d_template.com'
        file_out = data_location + '/dat/find_p0_tmp.com'

        f = open(file_in,'r')
        ft_script_template = f.read()
        f.close()

        # __ Modify nmrpipe script by filling in applied phase /
           filename __
        b = ft_script_template.replace("p0 0.0",("p0 " + str(p)))
        a = b.replace("p1 0.0",("p1 " + str(p1)))
        ft_script_new = a.replace("-in test.dat",("-in " + str(
           filename)))

        f = open(file_out,'w')
        f.write(ft_script_new)
        f.close()

        # __ Use nmrPipe to FT, fit the peaks and get the intensity
            report __
        spectra_directory = data_location + '/dat'              #
           Location of the FT'ed data
        os.chdir(spectra_directory)

        copy_tcl = 'cp ' + ama_bin + '/pk.tcl ./ >& ama.tmp'    #
           pk.tcl
        os.system(copy_tcl)
        os.system("chmod +x find_p0_tmp.com")
        os.system("chmod +x pk.tcl")
        os.system("./find_p0_tmp.com >& ama.tmp")               #
           FT
        os.system("./pk.tcl >& ama.tmp")                        #
           Peak-picking
        retain_ft = 'cp test.ft ' + filename + '_phased.ft'
        os.system(retain_ft)                                    #
           Retain the phased spectra for analysis

        os.chdir(ama_bin)                                       #
           go back
```

```
# __ Extract the peak intensity from the peak picking
   report __
peak_table = spectra_directory + '/test.tab'
r = open(peak_table, 'r')

for j in range (0, 17):
        SkipHeader = r.readline()               # 1D report
           : 17 lines to skip


# __ Store peak intensities __
intensities = np.empty((0,1),float)
H_ppm = np.empty((0,1),float)

for line in r:
        columns = line.split()
        intensities = np.vstack([intensities, float(columns
           [9])])
        H_ppm = np.vstack([H_ppm, float(columns[3])])

# __ In case that nmrPipe does not find any peaks, set the
   intensity for this peak to 0 (necessary debug) __
if len(intensities) == 0:
        intensities = np.vstack([intensities, 0])

# __ Select the main peak (to suppress noise peaks and
   peaks from incomplete zeta-filtering) __
"""
__ Comment :    The peak of maximal intensity is chosen.
                This may be bad in the case, where a
                   residual water signal
                is present. In this case, a pk.tcl script
                should be used that ignores any peaks of
                a proton frequency of less than about 5.8
                   ppm
"""

maximum_intensity = np.amax(intensities)        # get only
   the peak of max. intensity

for l in range(len(intensities)):
        deviation = intensities[l] - maximum_intensity
        if int(deviation) == 0:
                if len(H_ppm) == 0:
                        H_ppm_of_max = 0 # debug for the
```

```
                                case that no peak is found
                else:
                                H_ppm_of_max = H_ppm[l]
r.close()

return maximum_intensity, H_ppm_of_max
```

Amaterasu Fourier Transform Interface: *ama_ft.py*

```
import os

def apply_phasing(p0, p1, filename, data_location, ama_bin):
        filein = ama_bin + "/ft_1d_template.com"
        fileout = data_location + '/dat/find_p0_tmp.com'

        f = open(filein,'r')
        filedata = f.read()
        f.close()

        # __ Fill in p0 / p1 into FT script __
        a = filedata.replace("p0 0.0",("p0 " + str(p0)))
        b = a.replace("p1 0.0",("p1 " + str(p1)))
        c = b.replace("-in test.dat",("-in " + str(filename)))
        newdata = c.replace     ("-out test.ft",("-out " + str(
           filename) + "_phased.ft"))

        f = open(fileout,'w')
        f.write(newdata)
        f.close()

        # __ FT and peak picking by nmrPipe __
        os.chdir(data_location + '/dat/')
        copy_tcl = 'cp ' + ama_bin + '/pk.tcl ./ >& ama.tmp'
        os.system(copy_tcl)
        os.system("chmod +x find_p0_tmp.com")
        os.system("chmod +x pk.tcl")
        os.system("./find_p0_tmp.com >& ama.tmp")
        os.system("./pk.tcl >& ama.tmp")
        os.chdir(ama_bin)
```

Amaterasu Output Function: *ama_output.py*

```
import numpy as np

def write_output(number_of_resonances,
    number_of_spinlock_power_values, Final_Intensities,
    Reference_Intensities_A, Reference_Intensities_B, data_location,
     H_ppm_peak_picked, data_fid_filenames, Final_p0, H_ppm_expected
    , Power_Values_valist, BF3, spin_lock_length,
    N_hard_pulse_length, N_hard_pulse_power_dB):

        # __ Output file for glove __
        output_filename = '../output/Int_r1rho_' + (data_location.
            rsplit('/', 1)[1]) + '.txt'
        f = open(output_filename,'w')

        # __ Line 1 __
        header = str(BF3)

        # __ For line 2: back-calculate spin-lock power values [Hz]
            from dB values (valist) __
        header_power = []
        spin_lock_line = []

        for i in Power_Values_valist:
                this_Hz = round(1 / ((np.power(10, ((float(i) +
                    float(N_hard_pulse_power_dB)) / 20)) / 250000) *
                     float(N_hard_pulse_length)))

                if this_Hz > 0:
                        header_power.append(int(this_Hz))
                        spin_lock_line.append(spin_lock_length)
                else:
                        continue

        # __ Line 2 (spin-lock duration [s]) __
        header = header + '\n0 0'

        for i in spin_lock_line:
                header = header + ' ' + str(i)

        # __ Line 3 (spin-lock power values [Hz]) __
        header = header + '\n0 0'

        for i in header_power:
                header = header + ' ' + str(i)
```

```
header = header + '\n'

f.write(header)


# __ Lines 4+ : Write peak intensity list (used to make
    input file for GLOVE later) __
q = 0

for i in range (0, number_of_resonances):
        peakstring = ''
        for j in range (0, number_of_spinlock_power_values)
            :
                peakstring = peakstring + ' ' + str(
                    Final_Intensities.item(q))
                q = q + 1       # counter to loop over both
                     peaks and intensites

        linestring = str(i+1) + " " + str(i+1) + " HN " +
            str(Reference_Intensities_A.item(i)) + " " + str
            (Reference_Intensities_B.item(i)) + "" +
            peakstring + "\n"
        f.write(linestring)
f.close()

print "Processing finished. \nOutput written to: ",
    output_filename


# __ Write detailed report __
report_filename = '../output/amaterasu_' + (data_location.
    rsplit('/', 1)[1]) + '_report.txt'
g = open(report_filename,'w')
report_header = 'Peak \t Power \t Filename \t\t Intensity \
    t  Peak ppm \t ppm expected \tp0 \t
    deviation_from_expected_ppmiation \t comment \n'
g.write(report_header)

warning_peaks = []
warning = 0             # flag, if warning was encountered
    for this resonance

q = 0

for i in range (0, number_of_resonances):
        warning = 0 # Going to next resonance. Thus, reset
            warning flag.
```

```
            for j in range (0, number_of_spinlock_power_values)
                :
                    deviation_from_expected_ppm = abs(float(
                        H_ppm_peak_picked.item(q)) - float(
                        H_ppm_expected[i]))
                    linestring = str(i + 1) + ' \t ' + \
                                 str(j + 1) + ' \t ' + \
                                 str(data_fid_filenames[q]) + '
                                     \t ' + \
                                 str('% 15.1f' %
                                     Final_Intensities.item(q))
                                     + ' \t ' + \
                                 str('% 3.3f'  %
                                     H_ppm_peak_picked.item(q))
                                     + ' \t ' + \
                                 str('% 3.3f'  % float(
                                     H_ppm_expected[i])) + ' \t'
                                      + \
                                 str(Final_p0.item(q)) + '\t' +
                                      \
                                 str('% 3.3f'  %
                                     deviation_from_expected_ppm
                                     )
                    if deviation_from_expected_ppm > 0.2:
                            linestring = linestring + '\t\t
                                *****\n'
                            warning = 1
                    elif deviation_from_expected_ppm > 0.1:
                            linestring = linestring + '\t\t***\
                                n'
                            warning = 1
                    else:
                            linestring = linestring + '\t\t\n'
                    q = q + 1       # counter to loop over both
                         peaks and intensites
                    g.write(linestring)
            if warning == 1:
                    warning_peaks.append(i+1)
    g.close()

    print "----------------------------"
    print "Detailed report written to: ", report_filename
    make_glove_in = 'Int_r1rho_' + (data_location.rsplit('/',
        1)[1]) + '.txt'
    print "Run:\n r1rho2glove -i ", make_glove_in, " [
        temperature] > glove.in\n to generate the input file for
         glove.\n"
```

```
if len(warning_peaks) > 0:
        print " WARNING: Peaks with problematic peak
            picking:\n\n", warning_peaks
```

Amaterasu F-test Module: *ama_ftest.py*

```
"""
This script will take 2 input files from glove which
are fitted to a R1rho dispersion model (glove_r1rho.out,
alternative hypothesis) and a constant function
(glove_const.out, null hypothesis), respectively
and perform a F-test on the data.

"""

alpha = 0.99                    # significance cutoff

import numpy as np
from scipy import stats

filename_richards = 'glove_r1rho.out'
filename_const = 'glove_const.out'

c = open(filename_const,'r')
r = open(filename_richards,'r')

const = []
richards = []

for line in c:
        columns = line.split()
        const.append(columns)
c.close()

for line in r:
        columns = line.split()
        richards.append(columns)
r.close()


string_a = 'X2/DoF'
string_b = '(DoF'

Chi_2_const = np.empty((0,1),float)
Chi_2_richards = np.empty((0,1),float)

DF_richards = np.empty((0,1),float)
DF_const = np.empty((0,1),float)

for i in range(len(const)):
        if string_a in const[i]:
```

```
                if string_b in const[i]:
                        if const[i][0] == string_a:
                                Chi_2_const = np.vstack([
                                    Chi_2_const, float(const[i][1])
                                    ])
                                a = const[i][4]
                                b = a[:-1]
                                c = int(b)
                                DF_const = np.vstack([DF_const, c])



for i in range(len(richards)):
        if string_a in richards[i]:
                if string_b in richards[i]:
                        if richards[i][0] == string_a:
                                Chi_2_richards = np.vstack([
                                    Chi_2_richards, float(richards[i
                                    ][1])])
                                a = richards[i][4]
                                b = a[:-1]
                                c = int(b)
                                DF_richards = np.vstack([
                                    DF_richards, c])


F_ratio = np.empty((0,1),float)

for i in range(len(Chi_2_const)):
        Chi_2_const[i] = Chi_2_const[i] * DF_const[i]
        Chi_2_richards[i] = Chi_2_richards[i] * DF_richards[i]
        F = ((Chi_2_const[i] - Chi_2_richards[i]) / Chi_2_richards[
            i])
        G = ((DF_const[i] - DF_richards[i]) / DF_richards[i])
        F_ratio = np.vstack([F_ratio, F/G])

for i in range(len(F_ratio)):
        p_value = scipy.stats.f.cdf(float(F_ratio[i]), int(DF_const
            [i]), int(DF_richards[i]))
        if p_value > alpha:
                print "Residue: ", i+1, " __ F-value: ", float(
                    F_ratio[i]), "\t __ DF1-DF2: ", int(DF_const[i]
                    - DF_richards[i]), "\t __ DF2: ", int(
                    DF_richards[i]), "\t p-value: ", p_value
```

Amaterasu Peak Plotting Module: *ama_visualize.py*

```
import os
import sys
import numpy as np
import nmrglue as ng
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick

import ama_read_data

from matplotlib import rcParams
from pylab import *
from pyPdf import PdfFileWriter, PdfFileReader


rcParams['axes.labelsize'] = 7
rcParams['xtick.labelsize'] = 7
rcParams['ytick.labelsize'] = 7
rcParams['legend.fontsize'] = 5
rcParams['font.family'] = 'serif'
rcParams['font.serif'] = ['Arial']

def append_pdf(input,output):
    [output.addPage(input.getPage(page_num)) for page_num in range(
        input.numPages)]

def visualize(data_location):

        # __ Remember own path __
        ama_bin = os.getcwd() + '/bin'

        # __ Read data __
        fid_filenames, Reference_Flag, Number_Power_Values,
            Number_Peaks, Expected_ppm, Power_Values_valist, BF3,
            spin_lock_length, N_hard_pulse_length,
            N_hard_pulse_power_W = ama_read_data.
            read_input_data_structure(data_location)

        spectra_directory = data_location + "/dat/"
        os.chdir(spectra_directory)

        # __ Store filenames in array. This also contains reference
            data * 2 __
        datasets = []

        for i in sorted(os.listdir(os.getcwd())):
```

```
        if i.endswith(".dat_phased.ft"):
                datasets.append(i)
                continue
        else:
                continue

#  __ Necessary debug for large datasets __
#  For larger datasets (> test1000.dat) the python array
   mixes up the
#  correct numbering of the data.
#  To correct the numbering:

filtered = []

for v in datasets:
        a = v.replace("test", "")
        b = a.replace(".dat_phased.ft", "")
        filtered.append(b)
        z = sorted(filtered, key=int)   # sort filename
           numbers as ascending integers

datasets = []

for v in z:
        a = 'test' + v + '.dat_phased.ft'
        datasets.append(a)

os.chdir("../..")

# __ dB to Hz __

N_hard_pulse_power_dB = 10 * np.log10(N_hard_pulse_power_W)

hz_power = []

for i in Power_Values_valist:
        this_Hz = round(1 / ((np.power(10, ((float(i) +
           float(N_hard_pulse_power_dB)) / 20)) / 250000) *
            float(N_hard_pulse_length)))
        hz_power.append(this_Hz)

# __ Group peaks by peak number __

i = 0   # to count over peaks
j = 0   # to count over spin lock power values
r = 0   # to count over filename array
```

```
output = PdfFileWriter()

while (i < Number_Peaks):
        j = 0
        i = i + 1

        # __ Prepare figure containing subplots __
        f, axarr = plt.subplots(5, 5, sharey=True, sharex=
            True)

        title = 'Resonance ' + str(i) + '-HN'
        f.suptitle(title, fontsize=14, fontweight='bold')

        # __ Zoom on expected ppm range __
        x_l = float(Expected_ppm[i-1]) - 0.5
        x_u = float(Expected_ppm[i-1]) + 0.5
        row_counter    = 0
        column_counter = 0


        while (j < Number_Power_Values):
                j = j + 1
                a = datasets[r]
                r = r + 1

                spectra_directory = data_location + '/dat'
                os.chdir(spectra_directory)

                # __ Read spectrum __
                dic,data = ng.pipe.read(a)

                # __ Guess ideal axis limits __
                y_u = data.max() * 1.3
                y_l = data.min() * 1.05

                # __ Create a unit conversion object for
                    the axis __
                uc = ng.pipe.make_uc(dic, data)

                legend_Hz = str(int(hz_power[j-1])) + ' Hz'

                # __ Data for subplot for a single (current
                    ) peak __
                axarr[row_counter, column_counter].plot(uc.
                    ppm_scale(), data, label=legend_Hz,
                    color='k')
                axarr[row_counter, column_counter].set_xlim
```

```
                    (x_l, x_u)
                axarr[row_counter, column_counter].set_ylim
                    (y_l, y_u)
                axarr[row_counter, column_counter].yaxis.
                    set_major_formatter(mtick.
                    FormatStrFormatter('%.1e'))
                axarr[row_counter, column_counter].yaxis.
                    set_major_locator(MaxNLocator(4))
                axarr[row_counter, column_counter].legend(
                    shadow=False, fancybox=True)
                column_counter = column_counter + 1


                if column_counter == 5:
                        row_counter = row_counter + 1    #
                            advance 1 row
                        column_counter = 0               #
                            reset columns
                else:
                        pass

        # Fine-tune figure; hide x ticks for top plots and
            y ticks for right plots
        plt.setp([a.get_xticklabels() for a in axarr[0,
            :]], visible=False)
        plt.setp([a.get_yticklabels() for a in axarr[:,
            1]], visible=False)


        outputfile = '../../../plot/resonance_' + str(i) +
            '_plot.pdf'
        print "Plotting spectra of resonance ", str(i), " [
            expected ppm: ", Expected_ppm[i-1], "]"
        f.savefig(outputfile)

        plt.close()
        append_pdf(PdfFileReader(file(outputfile,"rb")),
            output)

output.write(file("../../../output/amaterasu_spectra.pdf","
    wb"))
```

Calculation of ζ-delays before acquisition : *ama_zeta.py*

```
"""

Script to calculate zeta-delays before acquisition from a peak list

Input file: Bruker TopSpin peak-picked list (exported as Microsoft
   Excel xls file) as text file

Example of input file format:

Peak    Region  Type    Index (F2)      Index (F1)      v(F2) [ppm]
        v(F1) [ppm]     v(F2) [Hz]      v(F1) [Hz]      Intensity [
   abs]
17      1       Automatic       202.9948044     188.0004639
   9.5223  128.09  6667.32 9088.107104     295109.05
11      1       Automatic       238.0028544     191.0014574
   9.3168  127.99  6523.44 9080.827776     277439.55
10      1       Automatic       249.9958798     578.000925
   9.2464  114.76  6474.14 8142.106545     272775.5


"""


# __ Specify parameters below __

bf = 70.948623                  # [15]N
N_distance_ppm = 1.9            # Chemical shift distance under
   which 2 peaks are considered "near"
H_distance_ppm = 0.19           # Chemical shift distance under
   which 2 peaks are considered "near"







import sys
import os
import numpy as np

nearest_proton = 0              # debug


class Compare:
```

```
        def __init__(self, current, to_compare):
                self.dataset = self
                self.current = current
                self.to_compare = to_compare

        def check_hydrogen(self):
                difference = abs(self.current - self.to_compare)

                if difference > H_distance_ppm:
                        zeta = 0
                        return zeta
                elif difference < H_distance_ppm:
                        zeta = 10
                        return zeta
                elif difference == H_distance_ppm:
                        zeta = 10
                        return zeta

        def check_nitrogen(self):
                difference = abs(self.current - self.to_compare)

                if difference > N_distance_ppm:
                        zeta = 0
                        return zeta
                elif difference < N_distance_ppm:
                        zeta = 10
                        return zeta
                elif difference == N_distance_ppm:
                        zeta = 10
                        return zeta



# __ Get command line input __


if len(sys.argv) < 2:
        print "\n Usage: ", sys.argv[0], " [input file]."
        sys.exit(0)


# __ Read input table __


r = open(sys.argv[1], 'r')
SkipHeader = r.readline()       # Skip over the 1-line header of
```

```
    the excel file

peaks = []
nitrogen_ppm = np.empty((0,1),float)
hydrogen_ppm = np.empty((0,1),float)


for line in r:
        if len(line) > 4:               # ignore incomplete lines
            of the Excel table
                columns = line.split()
                peaks.append(columns[0])
                hydrogen_ppm = np.vstack([hydrogen_ppm, float(
                    columns[5])])
                nitrogen_ppm = np.vstack([nitrogen_ppm, float(
                    columns[6])])
r.close()


# __ Prepare output __


f = open('zeta_report.txt', 'w')
g = open('zeta_values.txt', 'w')




"""

__ Check residues to see if they have neighbour peaks __

1. Loop over all residues               peaks[i]
2. Loop over all other residues         peaks[j]

Meanwhile, do 2 checks:

1. Check if residue pair is near in proton dimension
2. Then, if near in [1H], check if it is also near in [15N]

"""



for i in range(len(peaks)):

        # Array to store candidate zeta values for this residue (
```

```
    select the ideal zeta value later)
this_residue_zeta_candidates = np.empty((0,1),float)

# Store its neighbour peaks for the report
neigbours_for_this_residue = np.empty((0,1),float)

# Store the proton delta-omega values for each zeta
    candidate
zeta_candidate_dwH = np.empty((0,1),float)

# current residue is (i)
# loop over all other residues (j) with i not equal to j

for j in range(len(peaks)):
        a = Compare(hydrogen_ppm[i], hydrogen_ppm[j])
        is_it_near_in_hydrogen = a.check_hydrogen()

        # If residue is near in [1H], also check if it near
            in [15N]

        if is_it_near_in_hydrogen > 0:
                b = Compare(nitrogen_ppm[i], nitrogen_ppm[j
                    ])
                zeta_calculation_necessary = b.
                    check_nitrogen()

                if zeta_calculation_necessary > 0:
                        # Chemical shift of the residue
                            pair i,j is near in both [1H]
                            and [15N] dimensions

                        # Ignore the case that residue i is
                             near to itself
                        if i !=j:
                                # Calculation of zeta: 1 /
                                    (delta_n [ppm] * bf [MHz
                                    ] * 4)
                                delta_n = nitrogen_ppm[i] -
                                     nitrogen_ppm[j]

                                if delta_n !=0:
                                        zeta = 1 / (delta_n
                                            * bf * 4)
                                elif delta_n == 0:
                                        zeta = 999

                                    # [infinity]
```

```
                                    # Store the [1H] CS
                                        difference delta-omega
                                        to filter zeta list for
                                        multiple residues later
                                    zeta_candidate_dwH = np.
                                        vstack([
                                        zeta_candidate_dwH, (abs
                                        (hydrogen_ppm[i]-
                                        hydrogen_ppm[j]))])

                                    # Append zeta to candidate
                                        list (there may be
                                        multiple zeta candidates
                                        )

                                    if zeta > 0:
                                            this_residue_zeta_candidates
                                                = np.vstack([
                                               this_residue_zeta_candidates
                                               , zeta])
                                            neigbours_for_this_residue
                                                = np.vstack([
                                               neigbours_for_this_residue
                                               , peaks[j]])

                                    elif zeta < 0:
                                            this_residue_zeta_candidates
                                                = np.vstack([
                                               this_residue_zeta_candidates
                                               , -zeta])
                                            neigbours_for_this_residue
                                                = np.vstack([
                                               neigbours_for_this_residue
                                               , peaks[j]])

                                    elif zeta == 0:
                                            this_residue_zeta_candidates
                                                = np.vstack([
                                               this_residue_zeta_candidates
                                               , 0])


# __ Zeta report __

f.write("\n_____________\n\n")
```

```
        no_neighbours = str(len(neigbours_for_this_residue))    #
            Number of neighbor peaks of this residue

        linestring = "Residue " + str(peaks[i]) + " has " +
            no_neighbours + " neighbours : \n\n" + str(
            neigbours_for_this_residue) + "\n\n with the possible
            zeta values of: \n\n" + str(this_residue_zeta_candidates
            )

        f.write(linestring)
        f.write("\n\n")


        # __ Ideal zeta __

        # Figure out the "ideal value" of zeta for the measurement
        # Definition: ideal zeta is the one that suppresses the
            most nearby proton resonance

        counter = 0

        for i in this_residue_zeta_candidates:
                this_proton = zeta_candidate_dwH[counter]
                print "Zeta candidate # ", counter, " --- ", i, "
                    with dw[H] --- ", this_proton
                counter = counter + 1
                nearest_proton = np.amin(zeta_candidate_dwH)
                if this_proton == nearest_proton:
                        print "Chemical shift distance to most
                            nearby proton signal: ", nearest_proton
                        best_zeta = float(i)
                        print "Best zeta: ", str(best_zeta)

        if len(this_residue_zeta_candidates) > 0:
                linestring = "\n Selecting ideal value of zeta for
                    measurement: " + str(best_zeta)
                f.write(linestring)
                valuestring = str(best_zeta) + "\n"
                g.write(valuestring)
        elif len(this_residue_zeta_candidates) == 0:
                valuestring = str(0) + "\n"
                g.write(valuestring)

f.close()
g.close()
```

Peak picking script for nmrPipe : *pk.tcl*

```
#!/bin/sh
# The next line restarts using nmrWish \
exec nmrWish "$0" -- "$@" -notk

set tabName  test.tab
set specName test.ft
set tabCount 1

set tabDir [file dirname $tabName]

if {![file exists $tabDir]} {file mkdir $tabDir}


set thisSpecName $specName
set thisTabName  $tabName

set x1      1
set xN      420
set xInc    1024
set xExtra  1
set xLast   [expr $xN + $xExtra + 1]

    set xFirst  $x1

while {$xFirst <= 1 + $xN - $xExtra} \
   {
    set xNext [expr $xFirst+$xInc+2*$xExtra-1]
    if {$xNext > $xLast} {set xNext $xLast}

    readROI -roi 1 -ndim 1 -in $thisSpecName \
       -x X_AXIS $xFirst $xNext           \
       -noverb -silent

    pkFindROI -roi 1 \
      -sigma 38090.6 -pChi 0.0001 -plus 28666 -minus -28666 \
      -dx         1 \
      -idx        1 \
      -tol     2.00 \
      -hiAdj   1.20 \
      -lw     15.00 \
       -sinc -mask -out $thisTabName -noverb -silent

    set xFirst [expr 1 + $xNext - 2*$xExtra]
   }
```

```
exit
```

FT script for nmrPipe : *ft_1d_template.com*

```
#!/bin/sh

nmrPipe -in test.dat                                    \
| nmrPipe -fn SOL                                        \
| nmrPipe -fn SP -c 0.5 -off 0.4 -end 0.95 -pow 2        \
| nmrPipe -fn ZF -auto                                   \
| nmrPipe -fn FT -verb                                   \
| nmrPipe -fn PS -p0 0.0 -p1 0.0 -di                     \
| nmrPipe -fn POLY -auto -ord 0                          \
| nmrPipe -out test.ft -ov
```