To the Graduate Council:

I am submitting herewith a thesis written by Matthew A. Wilkerson entitled "Structural Analysis of Two Amyloidogenic Light Chain Antibodies, JtoD29A and JtoR68S." I have examined the final paper copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Biochemistry, Cellular, and Molecular Biology.

Chris Dealwis, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Vice Provost and Dean of Graduate Studies

Structural Analysis of Two Amyloidogenic Light Chain Antibodies, JtoD29A and JtoR68S

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Matthew A. Wilkerson

December 2003

Abstract

Resulting from a high degree of sequence variability, a few human light-chains have a propensity to form amyloid fibrils. The monoclonal $V_\lambda 6$ proteins are involved in fibril formation resulting in amyloidosis (AL). A previous study of a patient (Jto) with multiple myeloma in whom the $V_\lambda 6$ protein was deposited in the form of renal tubular casts was shown to form fibrils very slowly. The slow fibril formation resulted from the improved thermodynamic stability gained from an extra salt-bridge between Asp 29 and Arg 66B. To further investigate the effect of this interaction, two mutants Asp 29 to an Ala (JtoD29A) and Arg 66B to a Ser (JtoR68S) that disrupts the salt-bridge were made. Interestingly, the JtoD29A and JtoR68S have very different kinetics of fibril formation. The JtoD29A forms fibrils very slowly, while the JtoR68S forms fibrils at a rate similar to the pathogenic $V_\lambda 6$ Wil. In this study we have crystallized JtoD29A and JtoR68S in the same space group $P4_122$ with the same cell dimensions and solved their X-ray structures to 1.6 Å and 1.9 Å resolution, respectively. Structural comparisons reveal that although there are no significant main-chain conformational changes, there are several side-chain conformational changes due to the mutations. These differences contribute to JtoR68S having a larger exposed hydrophobic solvent surface as compared to JtoD29A. The Arg to Ser mutation in JtoR68S is also responsible for its increased negative electrostatic potential as compared to JtoD29A. We have also attempted to trap a pH induced structural intermediate of JtoR68S. While these findings are preliminary and need to be further investigated, these methods provide new means and insights for studying the amyloid phenomenon.

List of Figures Page

Chapter 1: Background and Introduction

The aggregation of normally soluble protein into fibrils is known as amyloidosis. The formation of these aggregates contributes to the disease state of several neurodegenerative diseases such as: Alzheimer's, Parkinson's, Hunington, Type II Diabetes, and L-chain associated amyloidosis (AL) (Delabar, Goldgaber et al. 1987) (Kruger, Eberhardt et al. 2002) (Jaikaran and Clark 2001) (Cardoso, Merlini et al. 2003) (Solomon, Frangione et al. 1982). Alzheimer's disease results from an accumulation of extracellular amyloid plaques accumulating in the central nervous system. The plaques result from a cleavage and secretion of Amyloid Precursor Protein (APP), which then propagates the formation of fibrils (Temussi, Masino et al. 2003). The fibrils form plaques around the nervous tissue and result in tissue death. Parkinson's disease is linked to intracellular amyloid fibrils resulting from intracellular aggregation of Transthyretin in the part of the brain responsible for control of motion, the substantia nigra. Accumulation of the fibrils destroys the neural tissue and leads to a loss in control of motion (Krishnan, Chi et al. 2003). AL is characterized by the deposition of aggregates composed of L-chain variable regions (V_L) in the kidneys, heart, and peripheral nervous system as amyloid fibrils. Wide-ranging effects occur from these depositions including loss of cellular functionality, tissue breakdown, and death. Patients with multiple myeloma, which is characterized by the overproduction of monoclonal L-chains, are linked to the development of AL. However, only as few as 10-15% multiple myeloma patients develop the AL disease. Each amyloid disease differs in the pathological effects and the protein pre-cursors that lead to the tissue specific formation of amyloid fibrils (Bellotti, Mangione et al. 2000). The wide-ranging deleterious effects from amyloid

disease have promoted a great deal of research to determine the mechanism in which a native protein aggregates into an amyloid fibril. It is apparent from recent work that the propensity to form amyloid fibrils results from both environmental conditions and variation of amino acid sequence leading to thermodynamic destablization.

Amyloid Hypothesis

The main step in the manifestation of amyloid disease is the conversion of normal, natively folded proteins into aggregates that are composed of mostly β-pleated sheets, which assemble into the amyloid fibril. Traditional hypothesis states that this unfolding event is driven by the relative stability of the native protein. Destabilization of the native fold will produce a partially unfolded intermediate, which becomes the nuclei responsible for the formation of the amyloid fibril. The process of destabilization of the native state is known as the lag phase (see Figure 1.1). The lag phase represents the time to form the nuclei. After the nucleus is formed, large polymers begin to grow at a rapid rate producing the fibril (Merlini, Bellotti et al. 2001) (Soto 2001) (Lansbury 1999) (Rochet and Lansbury 2000). As the fibrils grow they adopt a cross-β pleated structure, which was determined by X-ray fiber diffraction (Sunde and Blake 1997) (Sunde, Serpell et al. 1997).

Cohen and Calkin produced the first structural data pertaining to the amyloid fibril in 1959 (Cohen and Calken 1959). The structure of the fibril was ascertained from X-ray fiber diffraction studies on amyloid plaques isolated from amyloid diseased patients. This structural study only indicated that the amyloid plaque was of a fibrillar

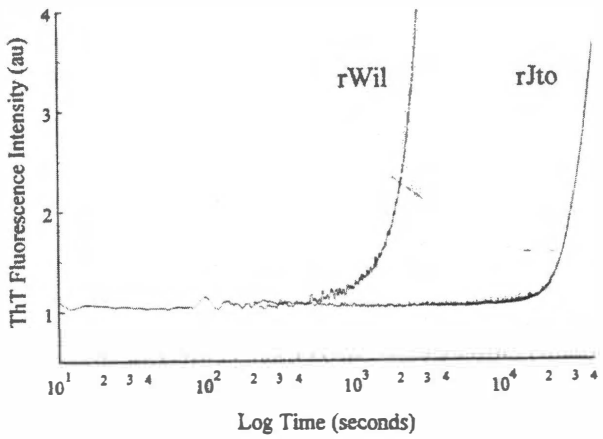


Figure 1.1. Fibrillogenesis assay of two $V_\lambda 6$ light chain antibodies, Jto and Wil. Notice that the non-amyloidogenic Jto has a much longer lag time than the amyloidogenic Wil.

nature (Cohen and Calken 1959). However, in 1967 the principle structure of the amyloid fibril was solved by electron microscopy (Shirahama and Cohen 1967). Utilizing fibrils isolated from light chain amyloidosis patients, Shirahama and Cohen found that a fibril ranged from 75-80Å in diameter. Each fibril was composed of 5-6 smaller repeating units known as protofilaments. The protofilaments are a smaller fibrillar unit arranged parallel to each other. Each protofilament appeared to be approximately 25-25Å in diameter. Each protofilament was found to be composed of three other subunit strands known as sub-protofilaments. Each sub-protofilament strand was between 10-15Å in diameter, and the group of strands was arranged helically and repeated every 35-50Å (see Figure 1.2). It was apparent from these studies that each amyloid fibril was composed of highly ordered structural repeats (Shirahama and Cohen 1967). It later became evident that each fibril adopted structural elements similar to the initial structures described by Shirahama and Cohen independent of the identity of the protein composing the fibrils (Cohen, Shirahama et al. 1982). As mentioned before, the traditional hypothesis states that the native state of the protein must be destabilized to initiate fibrillogenesis. Although the event that leads a protein to form a fibril has been rigorously studied over the past years, the mechanism of fibril formation is still unknown. Moreover other aggregates may be obtained instead of the amyloid fibril (Figure 1.3).

In Vitro Fibril Formation

It has been proposed that any protein will form fibrils if presented with the correct environmental stimuli. These stimuli would include lowering or increasing the pH, salts,

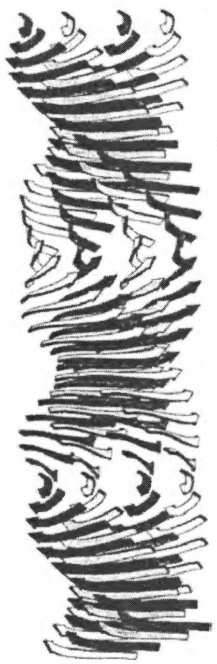


Figure 1.2. The model by Sunde and Blake of the generic amyloid fibril. A number of β-sheets (four shown here) comprise the protofilament. These sheets run parallel to the axis of the protofilament. Normal twisting of the sheet of the sheet results in result in a common helical axis giving a helical repeat of 115.5Å containing 24 β-sheets (Shirahama and Cohen 1967).

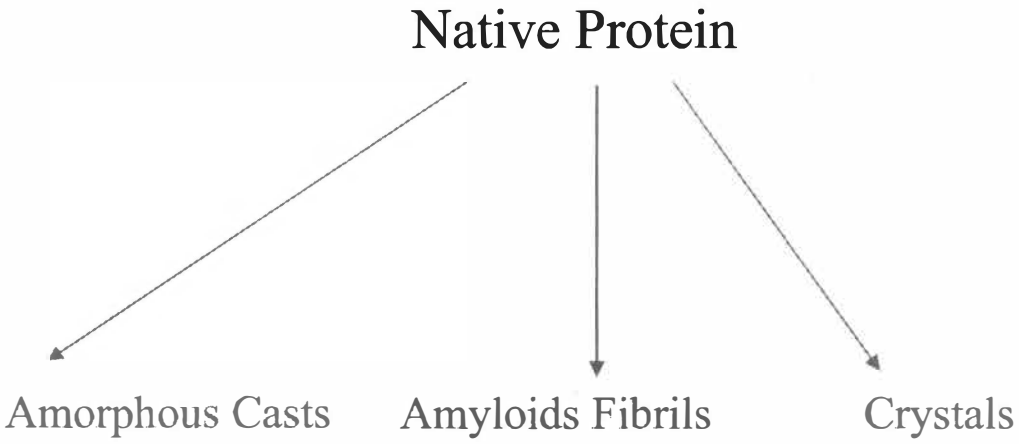


Figure 1.3. Three pathways are present that the native protein may take in forming an aggregate. The cast is characteristic of Light Chain Deposition disease and appears as amorphous clumps in the kidneys. Crystals may also form in the kidneys and result in Fanconi syndrome. The third aggregate is the amyloid fibril.

and other denaturant such as urea (Dobson 1999). The first case is the small α-helical protein acylphosphotase (Chiti, Webster et al. 1999). Chemical denaturing experiments with the denaturant trifluorethanol (TFE) indicate *in vitro* that a total unfolding of the protein perturbs the formation of fibrils. However, a slight denaturation of the structure over time will facilitate the formation of amyloid fibrils. This indicates that only a partial denaturation or rearrangement of structure is required to form fibrils (Chiti, Webster et al. 1999). A second study by Dobson's group used the SH3 domain of the P85α subunit of phoshpotidylinositol 3-kinase, which has not been associated with any amyloid disease, for study of fibril formation (Bucciantini, Giannoni et al. 2002). Even though the SH3 domain is not related to amyloid fibrils *in vivo*, *in vitro* the protein could be induced to form fibrils by incubations in low pH buffer. When proteins linked to amyloid disease were used for *in vitro* fibril assays, heterogenous mixtures of the proteins in different folded states were found (Khurana, Gillespie et al. 2001). Interestingly the SH3 domain, which is not linked to amyloid disease, had a uniform conformation at pH 5.0, which forms amyloid fibrils after several days of incubation. This shows that it is possible to study the unfolding pathway, which leads to the formation of fibrils, *in vitro* and lowering the pH of the proteins environment can induce formation of the nuclei (Bucciantini, Giannoni et al. 2002). A third study by Dobson, which used the amyloidogenic β2-Microglobulin protein, detected two partially folded intermediates that eventually formed amyloids (Chiti, Mangione et al. 2001). The study found that high concentrations of the proteins were not sufficient to form amyloid fibrils. However, incubation in chaotropic denaturants enabled the group to isolate the intermediates. Total unfolding was discovered at 3.0M Guanadine HCL. The partially unfolded intermediates were found

between 0.1M and 1.0M Guanadine HCL. These findings further show that fibril formation can be induced by perturbing the structure of the protein and that intermediates are formed along the particular unfolding pathway that leads to the formation of the amyloidogenic nuclei (Chiti, Mangione et al. 2001).

Anthony Fink's group applied Chris Dobson's idea of chemical perturbation of protein structure in forming amyloid fibrils to the study amyloidogenic light chains (Khurana, Gillespie et al. 2001). In this study Fink's group utilized the $V_{\kappa}4$ LC SMA to isolate partially unfolded intermediates by incubation in low pH buffers. Two intermediates were isolated. The first intermediate occurred at pH4, which resulted in forming amorphous casts. Amorphous casts are lesser aggregates that contribute to the Light Chain Deposition Disease (LCDD), which has a lower pathogenicity as compared to amyloidosis. A second intermediate was found at pH2. This intermediate led to the formation of amyloid fibrils (Khurana, Gillespie et al. 2001). This study shows that lowering the pH of the light chains environment can lead to the formation of intermediates along the amyloidogenic unfolding pathway (Khurana, Gillespie et al. 2001).

Probing The Primary Structure For Amyloidogenic "Hot Spots"

Several groups have studied the role of amino acid substitution in fibril formation. Through sight directed mutagenesis, Fred Stevens' work has shown several critical residues in $V_{\kappa}4$ light chains, which play a role in the formation of amyloids (Figure 1.4). (Stevens 2000). Stevens' group found that the mutation of a neutral glutamine residue at position 89 to an Asp increased the propensity for a $V_{\kappa}4$ LC protein LEN, to form fibrils

FW1	Red	Residues 4-13
FW2	Orange	Residues 19-25
FW3	Blue	Residues 33-40
FW3a	Blue	Residues 41-49
FW3b	Cyan	Residues 53-58
FW4	Yellow	Residues 61-67
FW5	Pink	Residues 72-81
FW6	Purple	Residues 84-92
FW7	Maroon	Residues 100-107
CDR1	Green	Residues 26-32
CDR2	Green	Residues 50-52
CDR3	Green	Residues 93-100

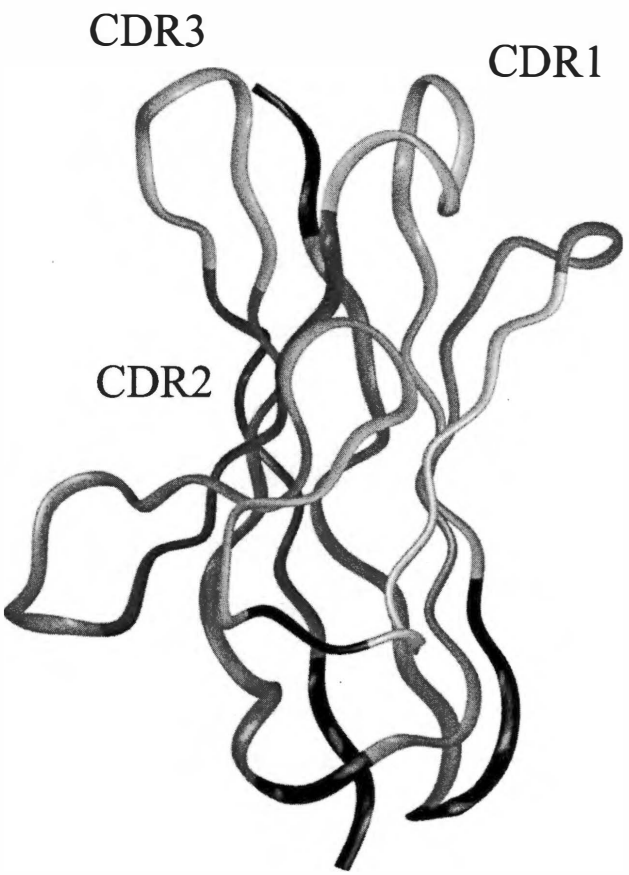


Figure 1.4. Diagram of a Vλ6 light chain antibody. Nine β-strands, FW 1,2,3a&b,4-7, compose the framework of the protein. The three complementary determining regions, which compose part of the antigen recognition motif, are shown in green.

(Pokkuluri, Gu et al. 2002). The increase in negative charge contributed by the Asp89 was proposed to destabilize the structure of LEN. Moreover, the study shows that a structurally related protein, Myelin Protein Zero (MPZ), contains Asp99 in an equivalent position that is involved in an ionic interaction with Arg38. The ionic interaction in MPZ contributes to stability and impairs fibril formation (Pokkuluri, Gu et al. 2002). Ron Wetzel's group has found that amino acid replacement in the β-framework (the scaffold of the light chain, see figure 1.4), results in the formation of fibrils (Hurle, Helms et al. 1994). Destabilization of the protein framework is achieved by amino acid substitutions, which disrupt the β-strand formation in the framework leading to a destabilized native fold and the propensity to form fibrils. However, Wetzel's group has also noted that amino acid substitutions in the antigen recognition loops of the light chains also increase amyloidogenesis (Helms and Wetzel 1996). Interestingly, the antigen recognition loops (CDRs) in the light chains are hyper variable and are normally not involved in structural stabilization. This indicates that amino acid substitutions at almost any region of the molecule can affect fibrillogenesis.

The nature of light chains promotes them to naturally undergo rapid substitutions in their amino acid sequence. Only a few light chains are able to form amyloid fibrils. Apparently, particular somatic mutations leads to fibrils (Raffen, Diekman et al. 1999). The comparison of non-amyloidogenic light chains to amyloidogenic light chains allows for comparison of structural properties leading to formation of amyloid fibrils. A great deal of work is now presented which attempts to explain what parts of a native light chain (Figure 1.5) promote the formation of the amyloid fibril.

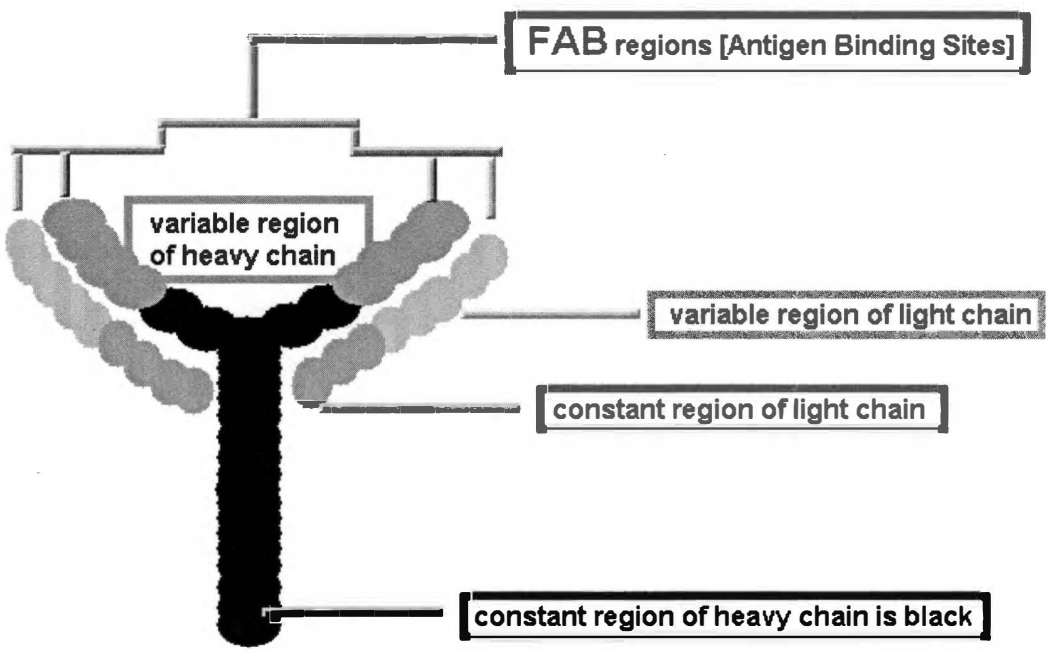


Figure 1.5. Cartoon diagram of the regions that compose an antibody. An antibody is composed of two types of chains: the heavy and the light chain. Both chains are divided into constant and variable regions. The variable region undergoes recombination at a rapid rate during B cell selection. The light chain variable region is the region that contributes to light chain amyloidosis.

Light Chain Amyloidosis

Light chain amyloidosis results from a deposition of amyloid fibrils composed of the variable region of monoclonal immunoglobulins light chains. This particular type of amyloid disease distinguishes itself from other amyloid diseases by several means. First, the immunoglobulin light chain is derived from an expanding clone of rapidly differentiating B cells where the variable region (V_L) of the light chain is highly mutated when compared to the germline gene. The consequence of the rapid differentiation results in a heterogenous pool of light chain variable region immunoglobulins. The wide variety of V_L have provided both a helpful system to study the phenomenon of amyloid fibril formation and a hindering wealth of proteins that may follow differing paths in forming the amyloid fibril. Second, the progression of the disease and the organs in which the disease manifest is reflective of the light chain involved. Some light chains do not form amyloid fibrils, but instead opt to form a lesser-ordered aggregate known as a cast. These casts result in LCDD, which has a decreased pathogenicty as compared to amyloidosis. Since it is apparent that the light chain has several paths to choose, of which different ends are achieved, the choice of path relies on the sequence of V_L light chain and ultimately the structure produced by that sequence (Bellotti, Mangione et al. 2000).

Organs Affected By Light Chain Amyloidosis

Light chain amyloidosis fits into both the primary and secondary category of amyloid disease. Primary amyloidosis results in the formation of amyloid fibrils without

any previous presence of disease, which may account for the presence of the plaque. Secondary amyloidosis results from a primary disease other than an amyloidosis, which produces amyloid fibrils as a secondary product. Development of amyloidosis in multiple myeloma patients is an example of secondary amyloidosis. With each form of amyloidosis, particular organs are targeted for the formation and accumulation of amyloid fibrils. In primary amyloidosis, the heart, lung, skin, tongue, thyroid gland, and intestinal tract may be involved. Localized amyloid plaques may also be found in the respiratory tract or other sites. Frequently the liver, spleen, kidney and the vascular system, especially the heart, are involved. Secondary amyloidosis shows a preference for the spleen, liver, kidney, adrenals, and lymph nodes. However, all organ systems are linked to the disease and vascular involvement may become widespread. The liver and spleen are often enlarged, firm, and rubbery. Fibril deposition in the kidneys can also result in an enlargement of the organs (Solomon, Weiss et al. 1992). Interestingly, amyloidosis is not the only light chain associated aggregate disease found in the kidneys. The kidneys also appear to accumulate other lesser-ordered aggregates composed of light chain V_L regions resulting in LCDD and Fanconi syndrome.

Fibrils And Casts

Approximately one third of all primary amyloidosis patients possess fibrils in the kidneys. Localization of the fibril in different parts of the kidney is not seen. However, the glomerulus appears to be the primary site of fibril accumulation. Along with fibrils, the kidneys also accumulate light chains in the form of casts, which are disordered precipitates of the proteins. The casts represent another disease known as Light Chain

Deposition Disease (LCDD). Solomon and co-workers have reported the various aggregation phenomena associated with the kidneys(Ozaki, Abe et al. 1994). Chris Dobson and co-workers have reported that light chains can be coerced to form fibrils by environmental stimuli. Solomon's group observed that renal concentrations of urea varied depending on the area of the kidney and condition of the kidney's health. Moreover, they found with the $V_\kappa L$ SMA and LEN fibrillogenesis could be induced by incubation in concentrations (μM) of urea that were physiologically similar to that found in the kidney. He also found two kidney osmoloytes, betaine and sorbitol, which counteracted the effects of urea. While urea destabilizes the compact state of the native protein by binding to the surface of the protein and decreasing the solvent exposed surface area, these osmolytes are excluded from the protein surface and promote a tighter packing of the native state. Solomon and co-workers have suggested that relative concentrations of these solutes acted as a control mechanism, which would either promote or perturb the formation of amyloid fibrils (Ozaki, Abe et al. 1994).

Light Chain Variable Domains

Two classes of V_L antibodies exist: the V_κ and V_λ classes. Both classes are linked to amyloid disease. However, the most prevalent class associated with amyloid disease is the V_λ light chains. In a study of 35 light chain amyloidosis patients, 77% of the group contained fibrils composed of V_λ light chains (Bellotti, Merlini et al. 1990). Interestingly, another group performing a similar study with a large field of patients noted that approximately 50% of all LC amyloidosis patients contained fibrils composed of the

$V_\lambda 6$ subgroup of the V_λ family (Solomon, Frangione et al. 1982). Moreover, all but one isolated $V_\lambda 6$ light chain resulted in light chain amyloidosis. The V_κ and V_λ classes of light chains differ in structure. The noticeable difference between the V_λ and the V_κ classes occurs at framework 3. In the V_λ class, an insertion occurs at this region, which results in the addition of two new frameworks strands, 3a and 3b. However, even with this insertion the overall native structure of both classes is very similar, as can be seen in Figure 1.6.

Three-Dimensional Structures

The light chain variable region is composed of anti-parallel β-sheets. The β-strands composing the framework of the protein are connected by loops. Three of these loops compose the complementary determining region (CDR). The CDRs of a light chain variable region and a heavy chain variable region comprise the antigen recognition motif of the antibody. Sequences of the CDRs are highly variable between each light chain and these regions are the sites of rapid recombination during B cell maturation. A diagram of a $V_\lambda 6$ light chain can be seen in Figure 1.4. Shown here are all known structures of V_κ and V_λ light chains. It is easy to see from the diagram that the global structure of the proteins is relatively the same.

The Jto And Wil Structures

The Jto and Wil $V_\lambda 6$ light chains were isolated from two patients, Jto and Wil. Wil was a classical amyloidosis patient, who secreted a high amount of $V_\lambda 6$ light

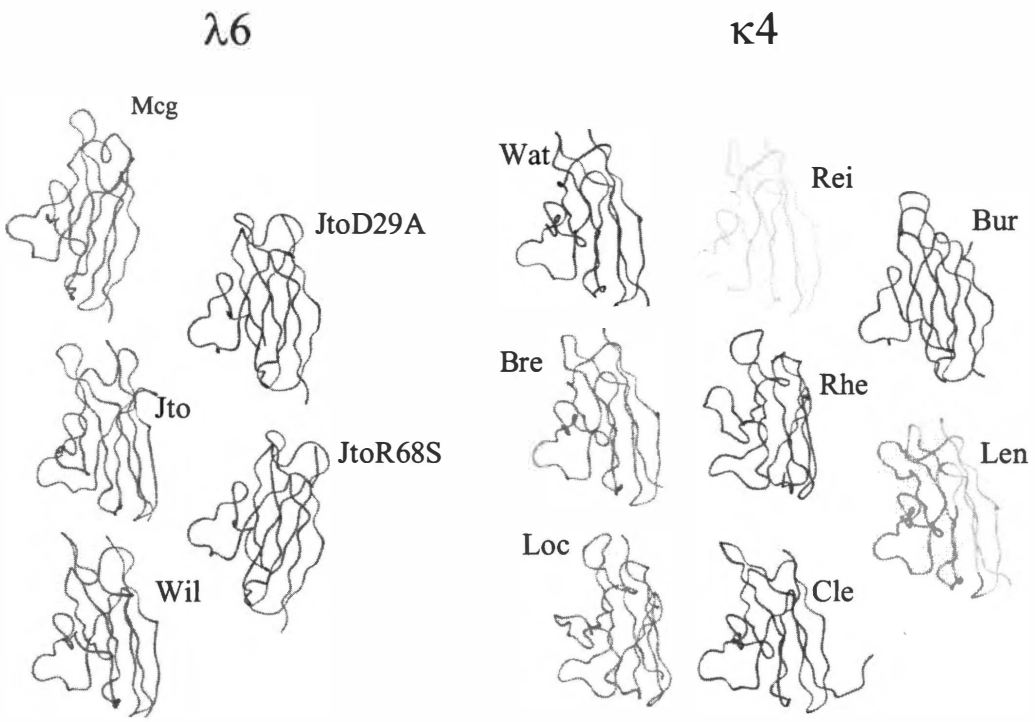


Figure 1.6. All known light chain variable region structures. On the left are the $V_{\lambda}6$ structures, while the $V_{\kappa}4$ structures are on the right. Notice that all adopt a similar global structure.

chains in the urine. The patient Jto also secreted a high amount of $V_\lambda 6$ light chains in the urine, but did not develop amyloidosis. Instead this patient developed renal casts characteristic of the lesser pathogenic light chain deposition disease. This presented an exceptional chance to study the $V_\lambda 6$ light chains, since the Jto light chain was the only isolated $V_\lambda 6$ light chain that did not form amyloids in the patient. The genes for Wil and Jto were both isolated, and the proteins were expressed recombinantly in *E. coli.* Fibrillogenesis assays performed on the two showed that Jto would form fibrils *in vitro*, but these fibrils occurred very slowly as compared to the pathogenic Wil (Figure 1.1). Even though the two proteins differed in their rates of fibril formation, the stability of the two proteins was negligible as the the Tm of Jto was 45.2°C and the Tm of Wil was 38.3°C (Wall, Schell et al. 1999). To determine the factors contributing to this stability, the X-ray structures of both Jto and Wil were solved (Pokkuluri, Solomon et al. 1999). The two structures were relatively superimposable with an *rms* deviation of the Cα atoms of approximately 0.8 Å. However, a difference was found in the region of CDR-2, which was an unusual arginine at position 68 in this loop. This Arg68 formed a new ionic interaction (Figure 1.7) with an aspartic acid at position 29 (see Figure 1.7). This new ionic interaction was hypothesized to be the stabilizing factor in Jto, which prevented the patient from developing amyloidosis.

To further study this new ionic interaction, two mutants of Jto were made: JtoD29A and JtoR68S. In both mutants, the acid (JtoD29A) and the base (JtoR68S) are

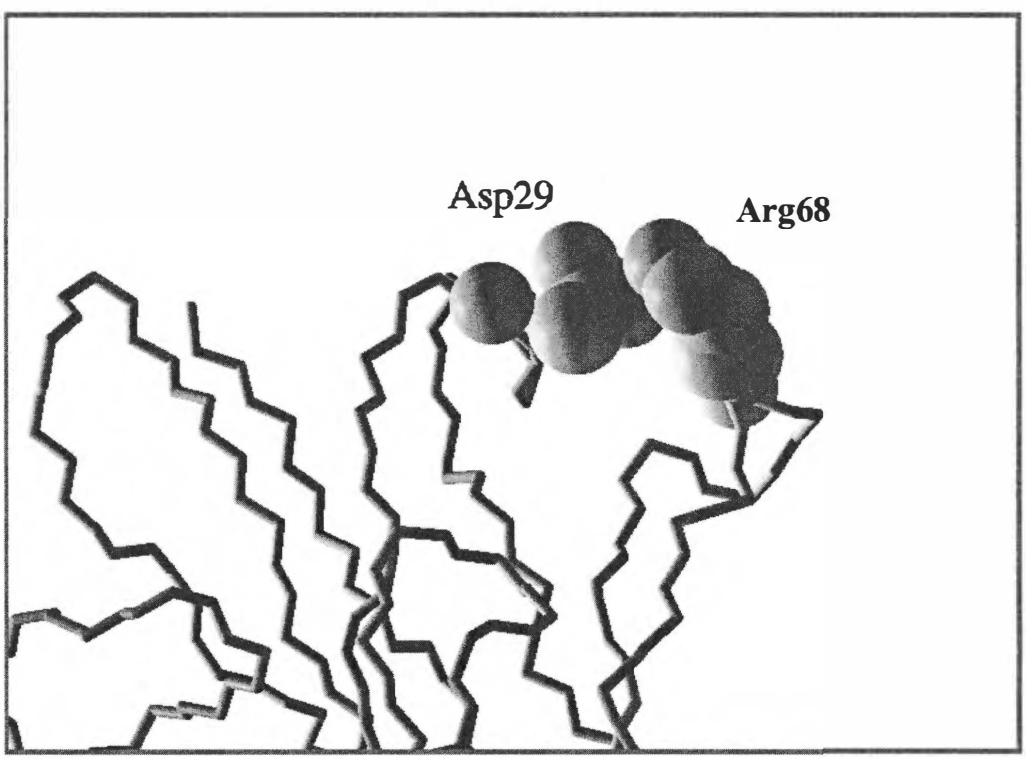


Figure 1.7. The new ionic interaction found in Jto between aspartic acid 29 and arginine 68.

mutated back to the germ line residue. Not surprisingly both the JtoD29A and JtoR68S exhibited superimposable unfolding curves with Tm around 42°C (see Figure 1.8a). However, they had different kinetic properties of fibril formation (see Figure 1.8b). JtoR68S was a fast fibril former occurring at a rate comparable to Wil. On the other hand, the JtoD29A formed fibrils at a rate slower than Jto (Gupta, Wilkerson et al. 2003). Since these two mutants differ by only one amino acid substitution and possess such differing rates of fibril formation while maintaining similar stabilities, the structure of the two proteins is sought.

Aim Of This Work

The work presented here involves the crystallization and structure determination of the JtoD29A and JtoR68S proteins. The properties of the native state dictate the speed of fibrilogenesis. Thus determination of the native structure of an amyloidogenic protein can allow for elucidation of structural characteristics promoting fibrilogenesis. For this reason we have solved the structures of the JtoD29A and JtoR68S proteins. Coupled to this we would also like to determine the structure of intermediates along the pathway to form amyloid fibrils. It has been shown that decreasing the pH of the buffer containing amyloidogenic proteins resulted in the capturing of intermediates (Chiti, Mangione et al. 2001). We also attempted to trap a structural intermediate of the JtoR68S mutant by incubation of crystals in low pH buffers. We hypothesize that the constraints placed on

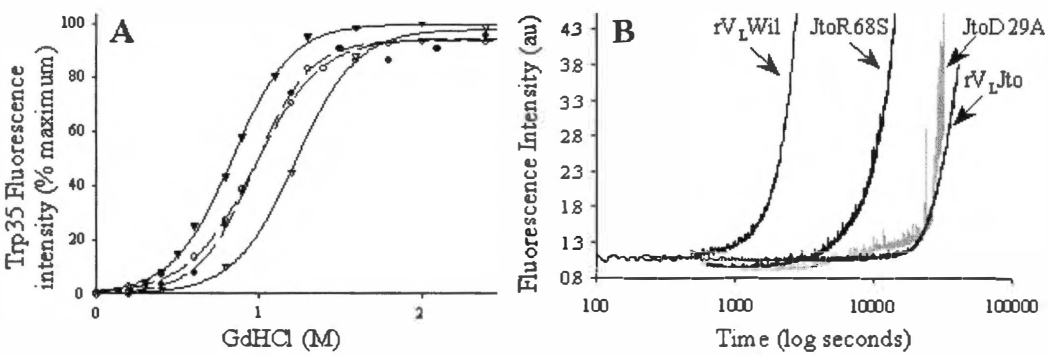


Figure 1.8. Fibrillogenesis of $V_\lambda 6$ proteins: Folding stability and kinetics (A) Protein unfolding curves (fluorescence intensity of Trp35 at 350 nm) (B) Fibril formation (PBS, pH 7.5, 37° C (increase in fluorescence emission of thioflavin T at 490 nm [excitation = 450 nm]) ▼, $V_\lambda 6$Wil; ○, JtoD29A; ●, JtoR68S; √, $V_\lambda 6$Jto.

the structure by the crystal packing forces may prevent a total unfolding of the protein. Instead the crystal may preserve an intermediate without breaking the crystal. These structures should be highly useful for two reasons. First, the structures of the JtoD29A and JtoR68S will allow us to better understand what drives a $V_{\lambda}6$ protein to form amyloid fibrils. Second, the structures of intermediates from pH denaturing will gives us insight into what occurs structurally on the path to forming the nuclei for amyloid fibrils.

Chapter 2: Methods

Expression And Purification Of JtoD29A And JtoR68S

The original cDNA encoding the $V_\lambda 6$ Jto protein was isolated from an enriched population of bone marrow cells from patient Jto with multiple myeloma and sub-cloned into a pET-29(+) vector for protein expression in *E. coli* cells. The procedure for cloning, protein expression, and purification of both $V_\lambda 6$ Jto and $V_\lambda 6$ Wil are described elsewhere. The JtoD29A and JtoR68S mutations were constructed by use of the Stratagene Quick-Change site-directed mutation kit. Oligonucleotides containing the appropriate bases changes, D29A and R68S, were used to PCR amplify the $V_\lambda 6$Jto/pET-29 plasmid. The template plasmid was digested with the Dpn1 nuclease. The daughter plasmids were transformed into DH5α *E. coli* for plasmid production. The mutant $V_\lambda 6$ proteins JtoD29A and JtoR68S proteins were re-transformed for protein production in *E. coli* as previously described. Recombinant proteins were subsequently purified by nickel affinity chromatography utilizing Nickel-NTA sepharose from Qiagen. All proteins were expressed and purified by our collaborator John Wahl at the University of Tennessee Medical Center.

Crystallization Of JtoD29A And JtoR68S

A protein crystal consists of a highly ordered lattice (Figure 2.1). The protein is arranged in a three-dimensional array known as the crystal lattice shown in Figure 2.1. The figure is a two-dimensional lattice representing the **a** and **b** axes. If one point in the lattice is moved one repeating unit along either axes it has moved to the adjoining unit

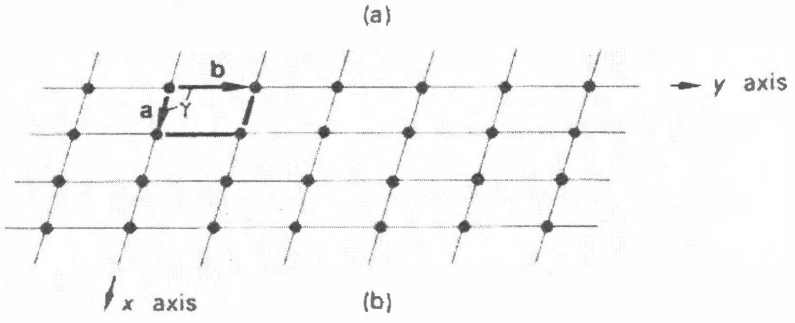


Figure 2.1. A two-dimensional lattice showing the a and b axes. A translation of one unit along either the **a** or **b** axis will result in a cell identical to the starting cell. Notice that the **γ** angle sits between the **a** and **b** axes. The combination of the **a** and **b** axis with the **c**, which is coming out of the plane of the page, defines the unit cell. Figure taken from Ladd and Palmer (Ladd and Palmer 1985)

cell. The combination of the **a** and **b** axes with the third axis, **c**, gives the unit cell. The unit cell is the smallest repeating unit, which is defined by its three cell edges (**a**, **b** and **c**) and the three angles formed from the cell edges (α, β, and γ). There are fourteen different classifications of unit cells known as Bravais Lattices (shown in Figure 2.2). The Bravais Lattice can be understood as the three-dimensional stacking of nets composed of lattice point. There are seven crystal classes unequally represented by the fourteen Bravais Lattices. These seven are: triclinic, monoclinic, orthorhombic, tetragonal, trigonal, hexagagonal, cubic. The crystal systems are defined by the lengths of a,b,c and angle of α, β, and γ. Each crystal system is then further classified by the placement of lattice points. These classifications are P, C, I, and F. P, or Primitive, is defined as a unit cell composed of one lattice point. The C stands for C-centering, F stands face-centering, and I for body-centering. (Ladd and Palmer 1985).

The smallest repeating unit in the unit cell is known as the asymmetric unit, which can be a motif of a protein or the entire molecule of the protein. The molecules within the asymmetric unit can arrange around symmetry elements resulting in further classification of the unit cell. Thirty two symmetry classes define the thirty two point groups. These elements are 2-, 3-, 4-, and 6- fold symmetry elements. Other symmetry elements that exist are mirror planes, glide planes, and centers of inversion. However, these do not occur in protein crystal as all naturally occurring amino acids are of the L conformation.

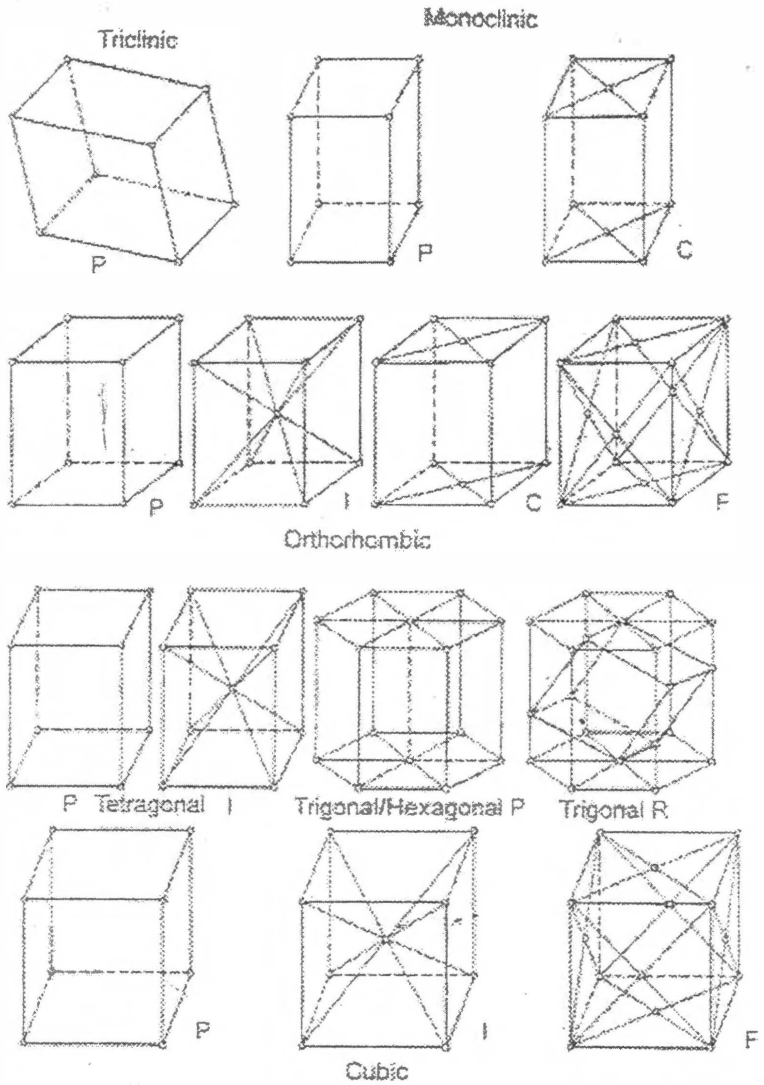


Figure 2.2. The fourteen Bravais lattices representing the seven crystal systems and the accounting for centering of lattice points. P is primitive, C is two-faced centered, F is all face centered, and I is body centered. Figure taken from Ladd and Palmer (Ladd and Palmer 1985).

The combination of the fourteen Bravais Lattices and thirty-two point groups result in 230 space groups. Protein that crystallizes in the space group $P4_122$, where P represents the primitive Bravais lattaice, 4_1 represents the tetragonal crystal system with a 4_1-screw axis of symmetry parallel to the **c** axis, and 2-fold symmetry axis parallel to **a** and **b**. The $P2_12_12_1$ space group has a 2-fold symmetry along all three axis which defines the orthorhombic crystal systems (Ladd and Palmer 1985). A diagram of both space groups is shown in Figure 2.3.

X-ray Diffraction

X-rays propogate at wavelengths on the order 1 – 10 Å, which are suitable for diffraction experiments of protein crystals. Normally X-rays at wavelengths between 0.5 and 1.6 Å are used. X-rays with wavelengths in this range will be scattered by electron clouds of atoms of similar size to the wavelength. Thus, X-rays on the range of 1.5 Å, the distance of a carbon-carbon bond will be useful for the study of protein structures. Unlike microscopy, where light is reflected off of a sample and then focused with a lens, X-rays cannot be focused by a lens to produce an image. However, the X-ray diffraction pattern of a crystal is its fourier transform, which can be deconvoluted to obtain the three-dimensional structure of the protein. X-ray diffraction is described Bragg's Law as follows:

$$n\lambda = 2d \sin\theta \quad \text{(eqn 1)}$$

where λ is the wavelength of the X-ray

n is the order of the Bragg reflection

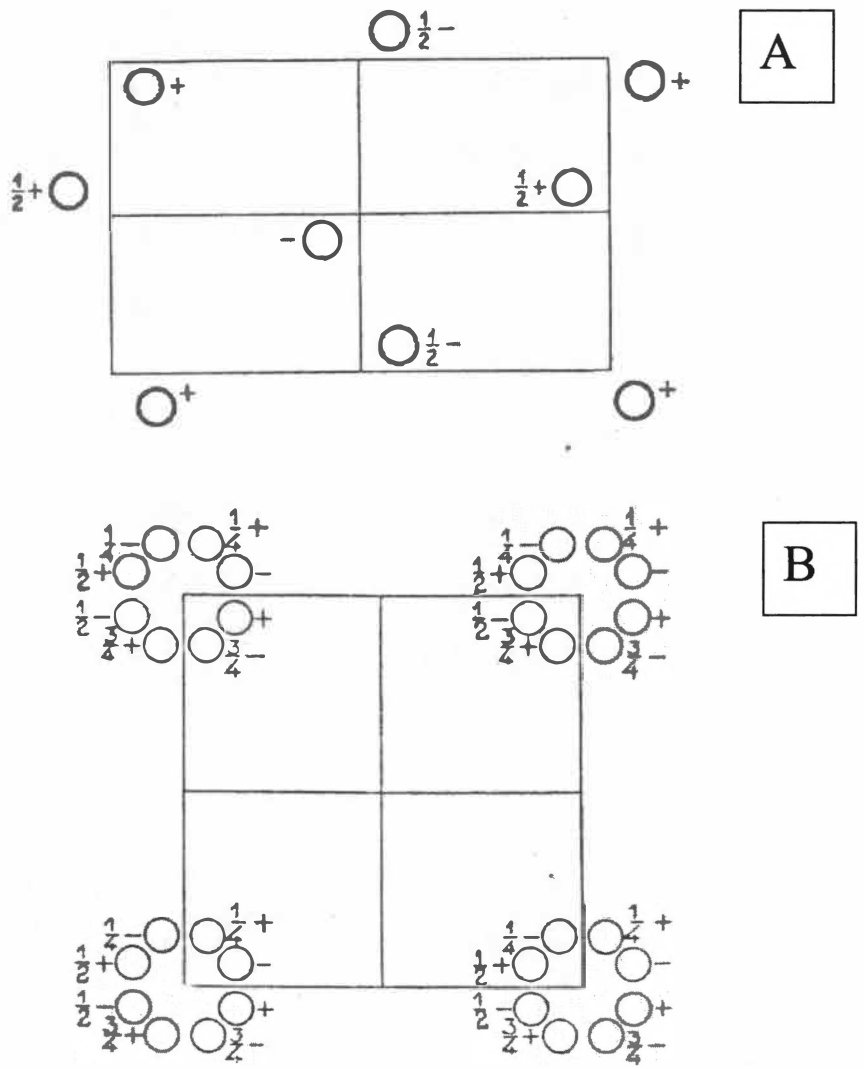


Figure 2.3. Stereographic projection of space groups. Shown is $P2_12_12_1$ (A) space group and $P4_122$ (B) space group. A + denotes that the point lies above the plane of the page, while a – denotes that a point lies below the plane of the page. Figure taken from the International Tables for Crystallography, Volume 2.

θ is the angle of diffraction from the incident X-ray beam

As shown in equation 1, as the angle of θ approaches 90° the resolution gets higher, characteristic of a small d value. The Bragg equation allows one to make the relationship between the reciprocal space (the diffraction pattern) and real space (see Figure 2.4). Though this method is a simple method for the evaluation of a protein crystal, it is not sufficient for determining the structure of the protein (Ladd and Palmer 1985).

X-rays are characterized as photons with wave-like properties. Therefore X-ray diffraction is caused by interference of waves. Now the wave equation resulting from X-ray diffraction can be written as:

$$F(x) = F\cos 2\pi (hx + \alpha) \quad \text{(eqn 2)}$$

where F(x) is the vertical height of the wave at any horizontal position x

F is the amplitude

h is the frequency of the wave

α is the phase

As shown from this equation a wave is characterized by the amplitude of the wave (F) and the phase. The diffracted X-ray is a complex wave resulting from the sum of all diffracted waves coming from the each atom in the crystal. This sum of waves is called a Fourier series. Further derivation will give us the structure factor equation:

$$F(h,k,l) = {}^{n}\Sigma_{j-1} f_j \, e^{2\pi I (hx+ky+lz)} \quad \text{(eqn 3)}$$

where F(h,k,l) is the three-dimensional structure factor that describes the reflection

${}^{n}\Sigma_j$ is summation of all atoms, j.

f_j is the scattering factor for the j^{th} atom.

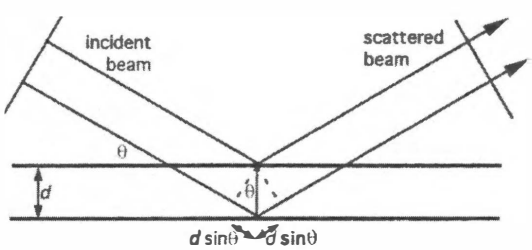


Figure 2.4. Scattering of X-rays at two planes. The extra path for X-rays scattered at the second plane is 2d sinθ, as per Bragg's Law.

x+y+z are the positions of atom j in the unit cell.

Performing a fourier transform on one will result in the other. Thus using the amplitude and the phase we can derive the electron density map of the unknown molecule.

Data Collection

The first step in structure determination is to collect a diffraction data set. Data is collected by oscillating the crystal to obtain maximum coverage of reciprocal space. For tetragonal crystals 90 degrees of data is required. The position of the reflections will be recorded as well as the intensities of each spot. The intensities are inversely proportional to amplitudes. In processing the data the intensities are scaled together. Usually to improve statistics a high redundancy of data is collected (Ladd and Palmer 1985).

Determination of structures

To solve the wave equation (equation 2) we need both the amplitude and the phase. The amplitude is obtained from the data, as mentioned above. However, the data gives no information about the phase. This is known as the "phase problem". To determine the three-dimensional structure the phase problem has to be solved. The Molecular Replacement method is one such method that can be employed to solve the phase problem.

Molecular Replacement

Molecular Replacement uses a known structure to provide phase information for the data of the unknown structure. If a known structure is to be used, then it must be

homologous to the experimental protein whose structure is being solved. The known structure is referred to as the search model. Examples of this are wild-type proteins being used as search models for mutants or a homologue of the target protein. The program AMoRe implements the Molecular Replacement method (Navaza 1994).

The first step of AmoRe (Navaza 1994) is to transform the search model from x,y,z data to Fourier coefficients representing the amplitude and phase. Next the amplitudes of the search model will be used to search against the known amplitudes found in the data. This is a two-step process, which involves a six-dimensional search. The first step is to locate the orientation of the target molecule by solving the rotation function.

As stated before to solve the structure from an X-ray diffraction, one must know the amplitude and the phase of the diffracted X-rays. The intensity of a diffraction spot is proportional to amplitude of the wave. However, the phase is not given by the experimental data. In 1934, Lyndo Patterson described a new method of processing a diffraction pattern while avoiding the phase problem. The function he described became known as the Patterson function. The Patterson function contains vectorial information between atoms in a molecule. The rotation function involves a Patterson search that searches for solutions that will give high overlap between the data and the model. The Patterson states that the vector describing the data (P(u)) is :

$$(P(u)) = \Sigma_h F_h^2 \exp^{[-2\pi ihu]} \quad (\text{eqn. 4})$$

where F_h^2 is the structure factor amplitudes squared

Σ_h is the summation of all reflections on the order of h.

h is order of diffraction.

u represents individual vectors.

The rotation function is calculated by the equation:

$$R(c) = \int_{u<r} Pm\,(Cu)\,Px\,(u)\,dV \quad \text{(eqn. 5)}$$

where (R(c)) is the rotation function

(c) is the rotation matrix.

(Pm) is the Patterson of the search model.

(Px) is the Patterson of the experimental data.

U is the search radius.

V is the volume of the unit cell.

The integration sums all the products of the Patterson of the model multiplied by the rotation Matrix (C) and the Patterson of the data over all values of the vector u that are less than the radius. The radius is usually defined as the radius of the search molecule. The self rotation function rotates the data on itself. The second rotation function is the cross rotation function. If the search model and the experimental are highly homologous, then these vectors should almost be identical. As shown in the equation for the rotation function, it is necessary to calculate the rotation matrix (C) (Rossman and Blow 1962). The set of angles, which traditionally have been used to define the rotation of one molecule to the next are the eulerian angles (shown in Figure 2.5). These angles are used to simplify the amount of rotation. The axis with highest symmetry is set to **z.** The first rotation, θ_1, rotates around the **z** axis. The second rotation, θ_2, rotates around the **y** axis. The last rotation, θ_3, is a second rotation around the **z** axis. This type of rotation is equivalent to rotating all axes. The rotation matrix is defined by the eulerian angles and

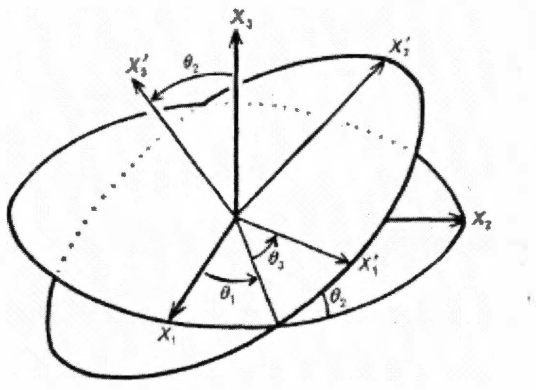


Figure 2.5. Eulerian angles α, β, γ used for the rotation search.

Table 2.1 Rotation matrix composed representing eulerian angles between the Patterson of the data and the Patterson of the search model (Rossman and Blow 1962).

A	B	C
$-\sin\theta_1 \cos\theta_2 \sin\theta_3 + \cos\theta_1 \cos\theta_3$	$\cos\theta_1 \cos\theta_2 \sin\theta_3 + \sin\theta_1 \cos\theta_3$	$\sin\theta_2 \sin\theta_3$
$-\sin\theta_1 \cos\theta_2 \cos\theta_3 - \cos\theta_1 \sin\theta_3$	$\cos\theta_1 \cos\theta_2 \cos\theta_3 - \sin\theta_1 \sin\theta_3$	$\sin\theta_2 \cos\theta_3$
$\sin\theta_1 \sin\theta_2$	$-\cos\theta_1 \sin\theta_2$	$\cos\theta_2$

is shown in Table 2.1. The plot of the rotation function in the eulerian angles, shown in Figure 2.6, show peaks that represents a solution. The highest signal to noise peaks from the rotation function are used for the translation function. These will also have the best correlation coefficient, which is defined by the equation:

$$\text{Correlation Coefficient} = \frac{\int v\,(Px - Pm)\,(Gx - Gm)\,dV}{\left(\int v\,(Px^2 - Pm^2)\,dV \int v\,(Gx^2 - Gm^2)\,dV\right)^{1/2}} \quad \text{(eqn 6)}$$

where Px is the Patterson of the data

Pm is the Patterson of the search model

Gx is the interference function associated with the data

Gm is the interference function associated with the search model

V is the volume of the unit cell

G represents the interfernce function, which is related to the volume of the sphere around the search molecule and the radius of the search molecule (Rossman and Blow 1962).

After the rotation function is calculated, the total correlation of the search model to experimental data is still fairly low as the translational position has yet to be determined. The translational coordinates are calculated by the translation function.

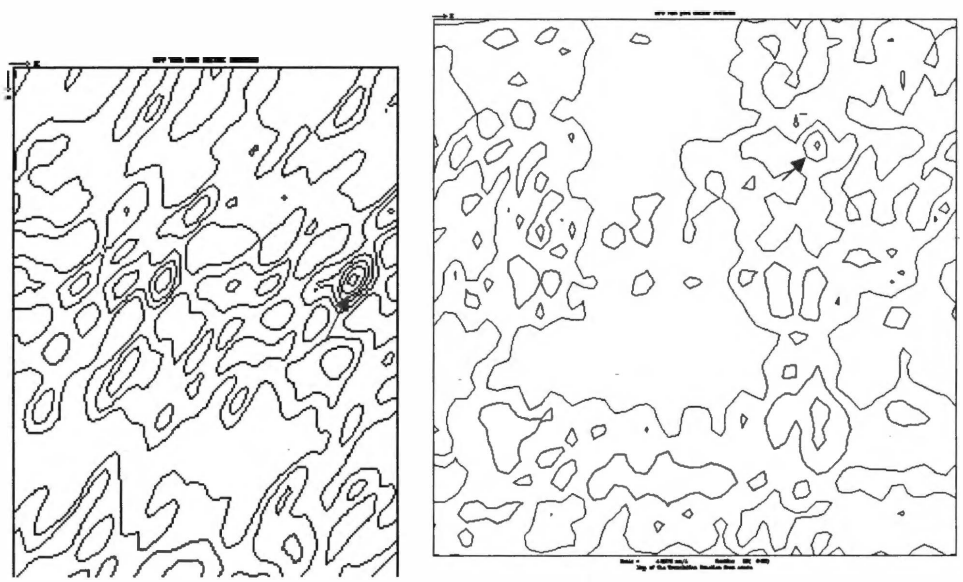


Figure 2.6. The rotation and translation functions for the JtoR68S structure. **(a)** A peak representing a solution from the rotation function and **(b)** a peak representing a solution from the translation function. Maps were contoured at 1σ.

However, before one attempts to perform the translation function, one must first ascertain the number of expected molecules in the unit cell.

Deriving Matthews Volume

The Matthews coefficient determines the solvent content of the asymmetric unit. The average crystal has a solvent content of approximately 30-70%. A typical crystal that diffracts to 2Å resolution is expected to contain about 40-50% solvent. By this point we know the space group of the crystal, which tells us how many asymmetric units we have in the unit cell. With this knowledge we then can ascertain how much of the unit cell is composed of protein molecules. To do this we use the relationship:

$$\text{Solvent content} = \frac{\text{volume of unit cell (Å)}}{(\text{no. of asymmetric units in the unit cell}) \times (\text{molecular weight})} \quad \text{(eqn 7)}$$

After calculation of the number of molecules in the asymmetric unit, we then perform the translation function for each molecule in the asymmetric unit using the solution from the rotation function as our search model (Blow 2002).

Translation Function

A translation function is derived for at least the top50 rotation function solutions. The translation function is defined by the equation:

$$T(x) = \Sigma_H \, m_H \, [Px]^2 \, [Pm]^2 \quad \text{(eqn 8)}$$

Where (Px) is the Patterson of the data

(Pm) is the Patterson of the model

$\Sigma_H \, m_H$, sets the limit of the translation to the unit cell defined by reflection H

Two sets of vectors are calculated in the translation function. The first is the crystallographic set of vectors, which is defined by crystal symmetry. An example would be the set of vectors between a molecule at the origin and its 4_1-related symmetry mates. The second set of vectors are the non-crystallographic vectors, which are calculated between sub-units in the asymmetric unit. Initially crystallogranophic vectors are used for ascertaining the translation function solutions. Then the top solution is fixed at the origin and the non-crystallographic vectors that relate this solution to the other molecules in the asymmetric unit are used to find their relative translational position. After each translation solution is found the correlation coefficient will increase if the solutions are correct. The correlation coefficients are calculated as follows:

$$corr.coef. = \frac{\sum(F_o - < F_c >)(F_c - < F_c >)}{\sqrt{\sum(F_o - < F_c >)^2 \sum(F_c - < F_c >)^2}} \qquad \text{(eqn 9)}$$

where Fo is the observed structure factor amplitudes from the data

Fc is the calculated structure factor amplitudes from the search model

After the translation function is performed the correlation between the solutions and the data should be increased. After the solution have been found, AmoRe will check the solution by performing a round of rigid body refinement with the program Fitting. Rigid-body refinement is considered to be another checking procedure, to prove the correctness of the solutions. In rigid-body refinement the molecule is treated as one rigid body where each atom does not move independent of the other atoms in the molecule. Performing rigid body refinement allows for optimizing the positions of each solution to match the

data. If the solutions are correct, the refinement should result in an increase of the correlation coefficient and a decrease of the R-factor, which is calculated by:

$$R - factor = \frac{\sum |F_o - F_c|}{\sum F_o} \qquad \text{(eqn 10)}$$

The R factor is a measure the agreement between the calculated model and the observed data. A lower R-factor implicates a model reflective of the observed data. If the solutions are determined to be good solutions, then the phases associated with search model can be applied to the data allowing us to solve the wave equation and the structure (Blow 2002).

Refinement Of The Structures

After the structure of the protein is determined, we must now analyze and refine the model to better fit the data. To do this we create an electron density map defined by the following equation:

$$\rho(xyz) = (1/V)\ \Sigma_h\Sigma_k\Sigma_l\ (2Fo\text{-}Fc)\ e^{(\phi c)}\ e^{-2\pi I(hx+ky+lz)} \quad \text{(eqn 11)}$$

where $\rho(xyz)$ represents the electron density

V is the volume of the unit cell

Fo is the structure factors related to the data

Fc is the structure factors related to the model.

ϕc represents the phase

$hx+ky+lz$ represents the position in reciprocal and real space

A 2Fo-Fc map is normally used in viewing electron density, because it gives weight to the structure factors of the data (2Fo), thus reducing biasing which may can be introduced

from the model. To properly address the bias implemented into the electron density map by the model, several simulated annealing (SA) omit maps are generated. The SA omit maps omit groups of five residues from the model before performing simulated annealing and calculating a 2Fo-Fc electron density map. In doing this, the bias from the model is eliminated. After the SA omit maps have been calculated, one must manipulate the model to fit the electron density. This is otherwise known as modeling. After the modeling is performed one must then subject the structure to a round of refinement, which restrains the structure to standardized bond angles and lengths as well as permitting better fitting of the model to the data. In refinement, the amplitudes of the model are calculated by inverse Fourier synthesis and compared to the amplitudes of the data. In calculating the structure factor amplitudes of the data, errors can be produced from weak reflections. These reflections, through counting errors, may sometimes have negative intensities. These negative intensities cannot be used or omitted for refinement calculations. To address this problem, a statistical method, developed by French and Wilson (French and Wilson 1978), based on Bayesian statistics has been implemented in the program TRUNCATE. Baye's theorem states that the confidence of a parameter, P, can be modified by an experimental measurement, X. Thus the desription of parameter P can be expected by comparison to the observation (French and Wilson 1978).

The effectiveness of the modeling is gauged by the R-factor (shown in equation 11). However, as refinement progresses, the data is biased towards the model. This can be seen in the equation from REFMAC, a program for minimization.

$$M = \Sigma_{hkl} W_f (F_o - GF_c)^2 + \Sigma_{hkl} W_p (\phi_o - \phi_c)^2 + \Sigma_{hkl} W_d (dt - dc)^2 + \Sigma_{hkl} W_b (b_0 - b_{min})^2 + \Sigma_{hkl} W_v V \quad \text{(eqn 12)}$$

Where W_f is the amplitude weighting coefficient

$(F_o\text{-}GF_c)$ is the Difference in the structure factor amplitudes of the data and the model with the model scaling factor G

W_p is the phase weighting coefficient

$(\phi_o\text{-}\phi_c)$ is the difference in the phase of the final solution from molecular replacement and the phase of the final model.

W_d is the restraints weighting coefficient

dt is the target interatomic distances

dc is the calculated interatomic distances

W_b is the non-bonded weighting coefficients

$(b_0\text{-}b_{min})$ describes the differences in non-bonding interactions

V is the determinant of the product-moment matrix of a planar group of atoms

W_v is the weighting coefficients for planarity restraints

Notice in this equation that $(\phi_o\text{-}\phi_c)^2$ is represent the phase of the final solution from molecular replacement minus the phase calculated from the model. Since both are phase from the models, this biases the electron density, which represents the data in real space, towards the model. To alleviate this bias, another value, R-free, is also calculated to better ascertain the agreement between the model and the data. To calculate an R-free value, data is withheld from the refinement calculation, thus is never biased by the refinement process. Usually five to ten percent of the data is used for calculating the R-free, and the R-free is calculated in a similar fashion as the R-value, which is shown in equation 7 (Murshudov, Vagin et al. 1996).

Conjugate Gradient Refinement

Refinement of a model structure manipulates the model to better fit the data. This process involves the movement of four parameters involved with the structure: **x, y, z,** and **B**. X, y, z, represent the coordinates required to describe a point in three dimensions, and the B-factor represents the amount of thermal motion associated with each atom. B factor are calculated by the equation:

$$B_j = 8\pi^2 \{u^2_j\} \text{ (eqn 13)}$$

where $\{u^2_j\}$ represents the displacement of an atom from its resting position

B_j represent the calculated B-factors in $Å^2$

This can now be accounted for in our structure factor calculation, where equation 4 becomes:

$$F(h,k,l) = {}^{n}\Sigma_{j-1} f_j \, e^{2\pi I (hx+ky+lz)} \, e^{(-Bj \sin 2\theta/\lambda 2)} \text{ (eqn 14)}$$

Maximum-Likelihood Method

To refine the JtoD29A structure and the JtoR68S structures we used two programs: CNS and REFMAC (Murshudov, Vagin et al. 1996). However, both programs implemented the same method of refinement: the Maximum Likelihood Method. The idea behind of maximum likelihood refinement is simple: the best model is most consistent with the observations. The consistency is measured statistically, measuring the probability that an observation should have been made. The probability is calculated by taking the joint probability of all reflection of the calculated and observed structure factor amplitudes. This is shown in the equation:

$$P(\text{max likelihood}) = \prod P[(Fobs);(Fcalc)] \text{ (eqn 15)}$$

where $\prod$ P is the joint probability that data is similar to the model

Fobs is the structure factor amplitudes of the data

Fcalc is the structure factor amplitudes of the model

When we perform modeling on the structure to better fit the electron density, which represent the data, the likelihood goes up, indicating that the model is better. The probabilities have to include the effects of all sources of error, including not just measurement errors implicit in the data but also errors in the model itself. But as modeling progresses and the structure improves, the model errors decrease, which means the probabilities start to increase. Increasing the probabilities also increases the likelihood (Murshudov, Vagin et al. 1996). A diagram outlining the cycle of refinement and model building can be found in Figure 2.7.

Simulated Annealing

Simulated annealing, which is a molecular dynamic method, is another method of refining a structure. A simulated annealing experiment computationally heats the protein to a high temperature (3500K) and then slowly cools the protein back down to ambient temperature. When a protein folds it will fall into a local energy minima, which is presumably the thermodynamically favorable state. However, a global minima may exist that is lower than local minima, but the protein is unable to proceed to it because of energy barriers surrounding the minima in which the protein sits. Increasing the temperature of the protein will provide the energy for the protein to rise above all energy barriers. However, restraints must be applied to the calculation, which will prohibit the breakdown of the protein. These restraints include bond lengths and bond angles. The

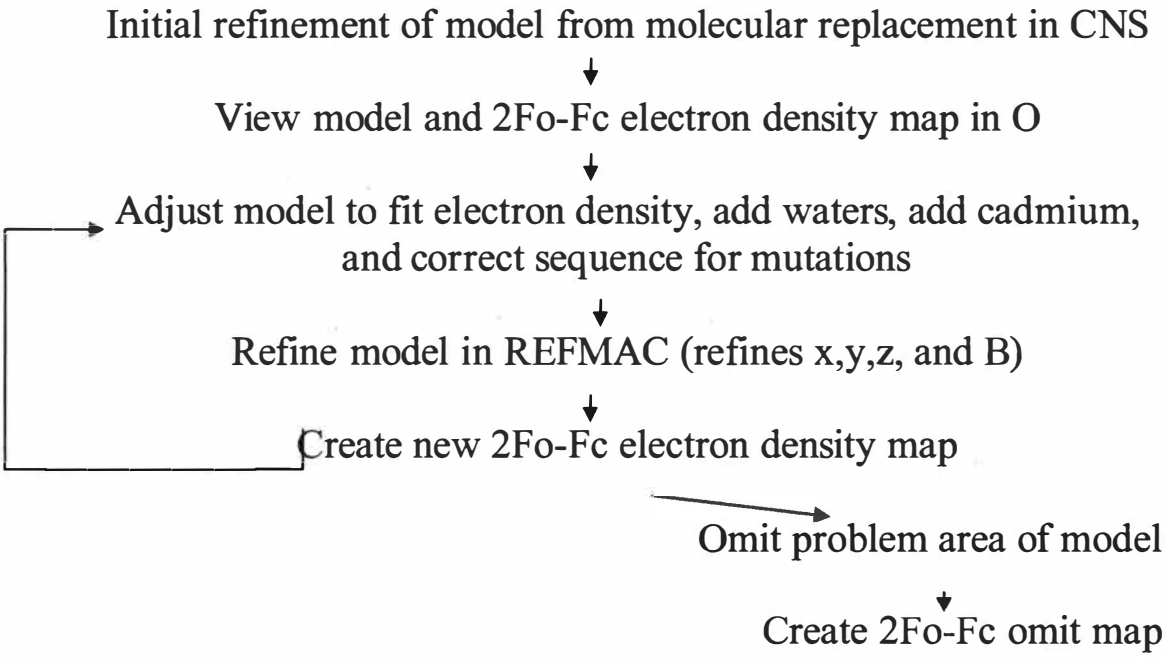


Figure 2.7. The refinement protocol used to refine both the JtoD29A and the JtoR68S structures.

restraints are applied in Hooke's Law, which states that the force exerted by a spring is equal to a constant (k, the Hooke's constant) multiplied by the change in position of the spring. This is shown in the equation:

$$F = k\,(x_1 - x_0) \quad \text{(eqn 16)}$$

Where F is the force

K is the Hooke's constant

$(x_1 - x_0)$ is the difference in initial position and the final position of the springThe Hooke's constant provides a limit to the amount of stretching of the spring. In the case of a protein, the spring is analogous to the bond lengths and bond angles. Thus, Hooke's law is applied to the simulated annealing calculation to prohibit the breakdown of the protein during the heating stage of the calculation.

As the protein is cooled it will fall into the lowest possible energy minima. However, this may produce a structure with distorted geometry. After the initial structure was solved we then performed positional refinement and modeling based on calculated electron density.

Hydrophobic Surface Area Calculation

To calculate the hydrophobic surface area of the JtoD29A and JtoR68S we first used the program Naccess to calculate the solvent accessible surface area (Hubbard and Thorton 1993). Naccess calculates the atomic accessible area when a probe is rolled over the surface of the protein. The program uses the Lee & Richards method (Lee and

Richards 1971), whereby a probe of given radius is rolled around the surface of the molecule. The path of the probe is followed and traced out by its center. This resulting path is the accessible surface. Typically, the probe has a 1.4Å radius, which is the same radius as a water molecule, and thus creates a solvent accessible surface. After NACCESS calculates the solvent accessible surface area, the results are shown for each atom in the structure. The solvent accessible surface area is divided into two sections, the polar and the apolar. Apolar residues include glycine, alanine, valine, leucine, isoleucine, proline, and methionine. However, polar residues may contribute apolar surface area to the structure. Examples may be exposure of C_β and C_γ by arginine residues. NACCESS reports the percent of each residue that contributes to the polar surface area and apolar surface area, as well as the total apolar and polar solvent accessible surface areas.

Difference Distance Matrix Plots

Difference distance matrix plots are useful tools in analyzing the difference in the Cα backbone of two similar structures. It is very useful when comparing two structures that are homologous or whose primary sequence differs by one residue, such as Jto, JtoD29A, and JtoR68S. The program works on a simple equation:

$$\text{Diff.} = (C_{\alpha n(x)} - C_{\alpha m(x)}) - (C_{\alpha n(y)} - C_{\alpha m(y)}) \quad \text{(eqn 17)}$$

where the distance in Å of a C_α between residue n and m of protein y is subtracted from the distance in Å of a C_α between residue n and m of protein x. The calculation is done for all possible distance, where n and m can be any possible residue in the protein. In doing so all possible distances between C_αs in a structure are calculated and compared to the second structure. The results are displayed in a matrix plot for ease of analysis. If

slight changes occur between two structures, then they can be observed in the matrix plot. The sensitivity range in difference density plot can set to observe differences as small as 1.5Å. All difference density matrix plots were calculated using DDMP from Yale University.

Electrostatic Surface Area Calculations

The protein surface is responsible for the interaction of the protein with the aqueous environment around it as well as other proteins and ligands. These interactions can be driven by the exposure of charged surfaces. To observe if any differences in charge occurred on the surface of the protein, we calculated electrostatic potential surfaces for Jto, JtoD29A, and JtoR68S using the program DelPhi (Rocchia, Alexov et al. 2001). DelPhi will calculate the electrostatic potential surface utilizing the Poissman-Boltzmann equation:

$$\nabla \cdot \epsilon(r) \nabla \phi(r) = \rho_{macro}(r) + \sum_i q_i n_i^0 exp^{-q_i \phi(r)/kT} \quad \text{(eqn 18)}$$

where ρmacro is the overall charge density plus the sum of the charged density due to dissolved charged ions

ni is the number of ions of type i in solution

q_i is the charge on the ion

ϕ is the electrostatic potential in that region of space

k is Boltzmann's constant

T is the temperature.

The equation results in the calculation of reverse finite difference in the electrostatic potential on the surface of the protein. The Green theroem is appled to the electrostaic potential, which results in the formation of grid of electrostatic potentials. This grid may be used to map the surface electrostatic potential across the surface of the protein (Watson and Warwicker 1982). A dielectric is a medium that is a sufficient supporter of an electrostatic field. The lower the dielectric, the greater the electrostic potential, while the higher the dielectric reults in a lower dielectric potential. This is shown in the case of water, which has a high dielectric, where ions have a greater chance of shielding because of the amount of water molecules present. However, in the interieor of a protein, ions have a lesser chance of shielding, which results from the low dielectric of the hydrophobic interior. Thus, they have a strong electrostic potential. DelPhi allows the user to set the dielectric of the solvent and the solute (macromolecule). The dielectric of the solvent is set to 80, which is the dielectric of water. The dielectric of the solute is set to 2, which is the approximate dielectric of the interior of a protein. After calculation of the electrostatic potential surface, the surface can be displayed in a program like Insight.

Chapter 3: The High Resolution Structures of JtoD29A and JtoR68S at pH 5.0

Crystallization Of JtoD29A And JtoR68S

The JtoD29A protein was crystallized in the space group $P4_122$, where P represents the primitive Bravais lattaice, 4_1 represents the tetragonal crystal system with a 4-screw axis of symmetry parallel to the c axis, and the 22 represents the 2-fold symmetry axes parallel to and b. JtoR68S crystallized in both the $P4_122$ and the $P2_12_12_1$ space groups. The $P2_12_12_1$ space group has a 2-fold symmetry along all three axis which defines the orthorhombic crystal systems (Ladd and Palmer 1985).

Crystals were grown by vapor diffusion at 25°C. JtoD29A was crystallized in a mother liquor containing 0.8 M ammonium sulfate, 0.1 M sodium acetate pH 5.0, and 0.1 M cadmium chloride. A 10 mg/ml solution of JtoD29A protein was equilibrated with the mother liquor in a 1:1 ratio. The crystals were improved by subjecting the above condition to detergent and additive screens purchased from Hampton Research, USA (Garvito 1986). The best crystals were obtained when the detergent n-Decyl-B-D-maltoside was added to the hanging drop at its CMC. JtoR68S crystals were similarly crystallized by using a mother liquor of 0.8 M ammonium sulfate, 0.1 M sodium acetate pH 5.0, and 40 mM cadmium chloride. A 14 mg/ml solution of JtoR68S protein was equilibrated with the mother liquor in a 1:1 ratio. Both JtoD29A and JtoR68S crystals grew in the space group $P4_122$ with the unit cell dimensions shown in Table 3.1. The physical dimensions of the JtoD29A and JtoR68S crystals were 0.5x 0.1x 0.1 mm^3.

Table 3.1. Statistics and unit cell dimensions of the JtoD29A and JtoR68S crystal structures

	JtoD29A	JtoR68S
Number of molecules per asymmetric unit	2	2
R_{sym}	6.5%	6.6%
Resolution	14.74 - 1.60	15.00 - 1.90
Completion	97.3%	95%
R_{free}	29.3%	23.3%
Non-hydrogen protein atoms/asymmetric unit	1766	1820
Metal ions/ asymmetric unit	4	4
Water molecules/ asymmetric unit	107	115
Rms Deviations from Ideality		
Bond lengths (Å)	0.017	0.009
Bond angles (deg)	1.219	1.190
Space group	P 4(1)22	P 4(1)22
Unit cell dimensions	a=73.57Å,b= 73.57Å, c=92.29Å	a=72.2Å, b=72.2Å, c=94.3Å
	α=β=γ=90°, 90°, 90°	α=β=γ=90°, 90°, 90°

Data Collection And Processing

Cryoprotection of both JtoD29A and JtoR68S crystals was achieved by soaking crystals in respective mother liquors containing 10-15% glycerol. Diffraction data for all of the JtoR68S were collected at a temperature of –160° C at a wavelength of 1.5418 Å using a Rigaku RU-H3R rotating anode X-ray generator with osmic focusing mirrors using a R-axis IV^{++} image plate detector. The diffraction data set for JtoD29A were collected at the synchrotron source in Hamburg on DESY 9 source at the BW7A beamline by Dr. Remy Loris at the University of Brussels. Diffraction data were collected at -160°C and a wavelength of 0.97622Å using a Mar image plate detector. All data were indexed, integrated, and scaled using the programs DENZO and SCALEPACK (Otwinowski 1992). The scaled data from SCALEPACK were converted to the mtz format using the CCP4 program SCALEPACK2MTZ and the structure factors were

calculated using the program TRUNCATE that implements French and Wilson statistics (French and Wilson 1978). In order to determine the space group, we created pseudo precession pictures using the CCP4 script HKLVIEW (Collaborative Computational Project 1994).

Structure Determination Of The JtoD29A And JtoR68S Mutants

The JtoD29A and JtoR68S structures were solved using the molecular replacement method using the program AmoRe implemented in the crystallographic suite CCP4 (Collaborative Computational Project 1994) by Dr. Vibha Gupta, a previous post-doctorial student in Dr. Dealwis' lab. In Amore ROTING calculates the rotation function, TRAING calculates the translation function, and FITTING performs a rigid-body refinement of the structure. The model cell used for JtoD29A and JtoR68S was a=68.018 Å, b=58.388 Å, c=59.037 Å, $\alpha = \beta = \gamma = 90°$. We used data in the resolution range of 50-3.0Å for conducting molecular replacement. The radius of integration used was 0.0 to 23.738 Å. The three-dimensional X-ray structure of $V_{\lambda}6$ Jto (PDB # 1CDO), which differs at only one residue with respect to both JtoD29A (PDB # 1PEW) and JtoR68S (PDB # 1PW3), was used as a search model. For JtoR68S, the correlation coefficients for the two molecules/asymmetric unit from the translation function were 25.0% and 24.7%, respectively. The largest noise peak had a correlation coefficient of 13%. After fixing molecule 1 to find the relative position of the second molecule using TRAING, the correlation coefficient increased to 40.4%. Final fitting using the program FITTING further increased the correlation coefficient to 67.2%. For JtoD29A, the correlation

coefficients for the two molecules/asymmetric unit from the translation function were 23.1% and 20.2%, respectively. After fixing molecule 1 to find the relative position of the second molecule using TRAING, the correlation coefficient increased to 39.4%. After final fitting the correlation coefficient increased to 65.6%.

In order to fine tune the three orientation and three translational parameters the original solutions from MR were subjected to rigid-body refinement using CNS (Bruenger, Adams et al. 1998). This was followed by a round of simulated annealing and positional refinement using CNS.

Refinement Of JtoD29A And JtoR68S

After several rounds of positional refinement using CNS (Bruenger, Adams et al. 1998) interspersed with model building using the program O (Jones, Zou et al. 1991), we started to use REFMAC (Murshudov, Vagin et al. 1996). Though REFMAC is another refinement method based on the maximum likelihood method of refinement, it differs from CNS in several ways. REFMAC permits manual manipulation of the weighting schemes more readily than CNS. When using REFMAC we applied a high weight to the data to ensure that the model refined towards the experimental data. We also found several side-chains that had multiple conformations, and REFMAC permitted the refinement of multiple conformations of side-chains with ease. Finally both structures were subjected to B-factor refinement using REFMAC. As REFMAC was unable to refine occupancy of the alternate conformations, we conducted B-factor refinement for comparing the relative occupancies by analogy to the B-factors.

The final model of JtoD29A refined to a crystallographic R-free value of 29.3% at 1.6 Å resolution. The *rms* deviations from ideality in bond angles and bond lengths were 1.219 ° and 0.017Å, respectively. The final model of JtoR68S refined to a crystallographic R-free value of 23.34% at 1.9 Å resolution. The *rms* deviations from ideality in bond angles and bond lengths were 1.19° and 0.009 Å, respectively. Both proteins crystallized in the space group $P4_122$ with two molecules, A and B, in the asymmetric unit. The two molecules of JtoR68S were structurally identical. However, in the JtoD29A crystal, the electron density of the B molecule contained regions of disorder and overall possessed poorer density than molecule A possibly contributing to the high R-free value. The JtoD29A structure was also refined in a low symmetry space groups, $P4_1$ and $P222_1$, in order to eliminate error in space group assignment. In neither case was the final R-free value significantly lower with respect to the model refined in the $P4_122$ space group. We ascribe the high R-free of the JtoD29A structure to the disorder observed in molecule B, which has a different crystal packing environment to that of molecule A.

Both JtoD29A and JtoR68S were crystallized in the presence of cadmium chloride and the resulting structures contained four bound Cd^{2+} ions per asymmetric unit. In addition, the refined structure of JtoD29A contained 107 water molecules while that of JtoR68S contained 115 water molecules. In JtoD29A, the loop containing residues 39-42 is disordered. At 1.6Å resolution we were able to observe 7 alternate side-chain conformations for residues Met3, Ser55, Arg68b, Leu78, Lys 79, Val96 and Arg103 in the JtoD29A structure.

The average main-chain B-factor for the JtoR68S molecule is 15.6 $Å^2$ and for the side-chains, 15.4 $Å^2$. As anticipated, due to the inherent disorder observed in the JtoD29A structure the corresponding B-factor values of 21.9 $Å^2$ and 21.3 $Å^2$ were higher than those of JtoR68S. This is also consistent with the fact that the JtoD29A structure has a high R_{free} value of 29.3% where as JtoR68S has an R_{free} value of. 23.3%. The *rms* difference between molecules A and B of both the JtoD29A and JtoR68S crystals was 0.438Å and 0.516 Å, respectively suggesting almost complete identity. Therefore from this point onwards we will only describe comparisons and structural analyses conducted using molecule A of both protein.

2Fo-Fc Omit Maps Of The JtoD29A And JtoR68S Mutants

The 2Fo-Fc difference omit maps derived after several rounds of rigid-body refinement followed by a few rounds of positional refinement unambiguously defined the electron density of the mutated residues in both JtoD29A and JtoR68S (Figures 3.1a and 1b). The omit map of JtoD29A was produced by computationally replacing Asp29 with Gly. Figure 3.1a shows clearly electron density for the Asp to Ala substitution at position 29 in JtoD29A, while figure 3.1b shows electron density for the Arg to Ser substitution in JtoR68S.

Comparison of the main-chain conformations of Jto, JtoD29A, JtoR68S, and Wil

To determine whether significant main-chain conformational changes had arisen due to substitution of residues 29 and 68, the structures of JtoD29A and JtoR68S were superimposed with the wild-type structure of Jto (Figure 3.2) and further compared to

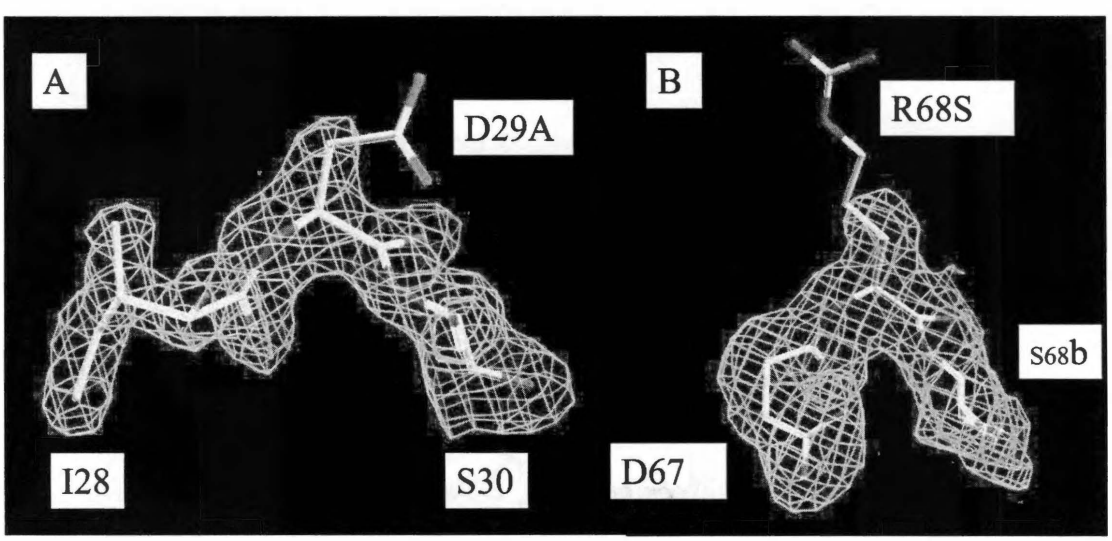
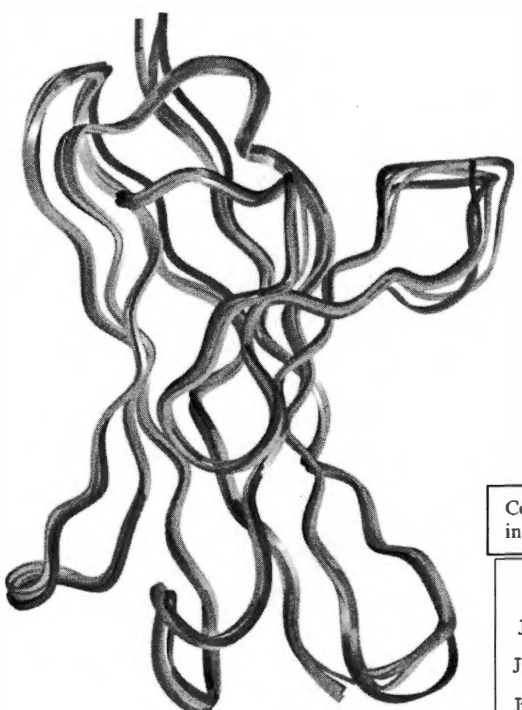


Figure 3.1. 2Fo-Fc omit maps generated at the site of the mutation for (A) JtoD29A model shown at the mutation site, D29A, and (B) the JtoR68S model shown at the mutation site, R68S. The contour level was set at 1 σ and the Jto sequence was modeled in order to show that the electron density for the mutation was unambiguous. The figure was prepared in O (Jones, Zou et al. 1991).



Comparison of *rms* deviations between the Jto structures and Wil in Å.

	Jto	Jto2	Jto5	Wil
Jto		1.368	0.669	0.865
Jto2			1.157	0.876
Jto5				0.807
Wil				

Figure 3.2. Main-chain Cα superposition of Jto (gray), JtoD29A (green), JtoR68S (red), and Wil (Blue) using the program LSQMAN implemented in O (Jones, Zou et al. 1991). Inset table shows the root mean squared *(rms)* differences between all four structures. Figure was prepared using the program Insight II (Biosym Technologies, San Diego).

another $V_\lambda 6$ LC structure Wil. All four molecules are virtually identical except within the immediate vicinity of the Asp29Ala and Arg68Ser mutations. Cα *rms* differences between the three models are shown in the inlay of Figure 3.2. The *rms* difference between Jto and JtoD29A is 1.368 Å, while it is 0.669 Å compared to JtoR68S. The rms difference between JtoD28A and JtoR68S is 1.157 Å. The main-chain *rms* differences between each of the Jto variants and the highly amyloidogenic $V_\lambda 6$ Wil protein, were also determined (Figure 3.2) (Pokkuluri, Solomon et al. 1999). Not surprisingly, there is a high structural homology between the four $V_\lambda 6$ LC structures as shown by low the Cα *rms* differences (Figure 3.2).

Side-Chain Conformational Analysis

Although the main-chain conformations of the four $V_\lambda 6$ LC proteins are almost identical, there are several side-chain conformational differences that result from the mutations. With reference to Jto, the side-chain residues most affected by the mutation in JtoR68S or JtoD29A are: His8, Ser36, Arg54, Arg61, Ser76, Leu78, Arg79, Glu81 and Gly108. Significant side-chain conformational differences are observed at the site of mutation (Figure 3.3). Both the D29A and the R68S mutations destroy the stabilizing ionic interaction between Asp29 and Arg68 seen in Jto. This in turn propagates reorganization of some of the neighboring side-chains. Examination of the structure surrounding the Arg68Ser substitution indicates that the side-chains of Asp29, Asn31 and Ser68b undergo conformational changes. For example, in JtoR68S the loss of an ionic interaction is being compensated by the formation of a new hydrogen bond between the

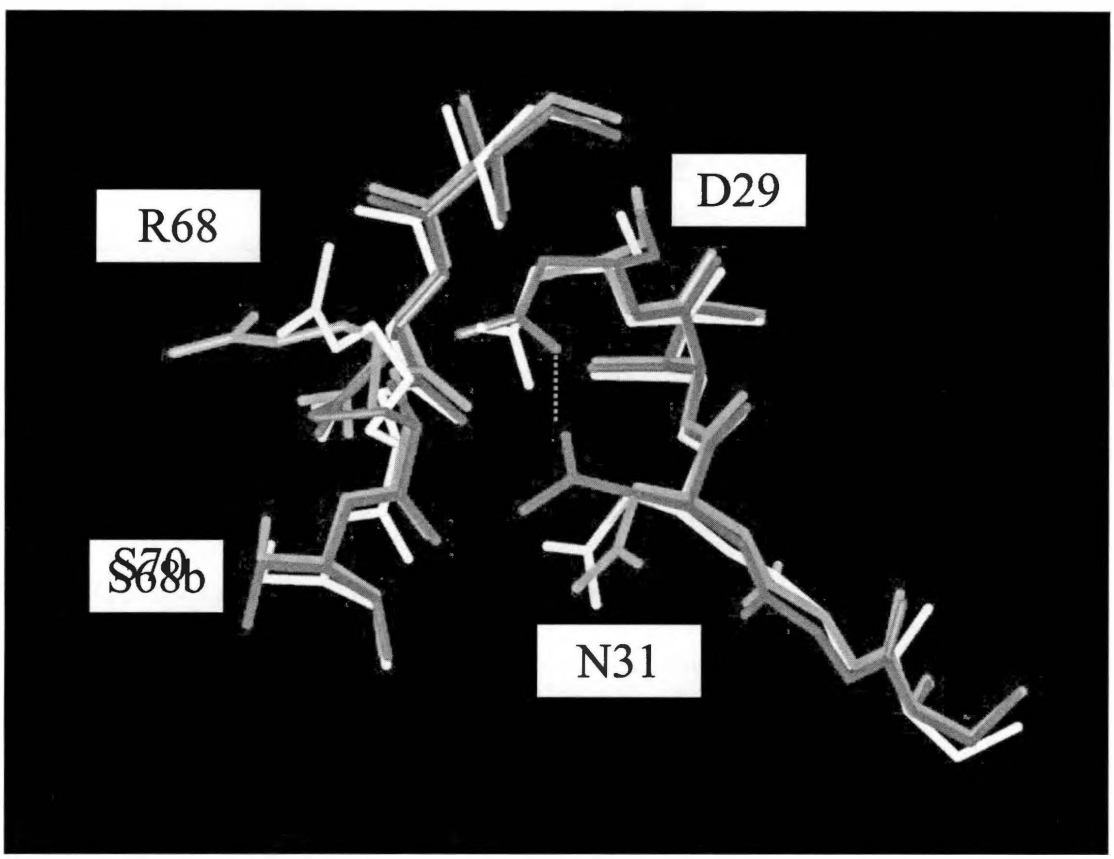


Figure 3.3. Superposition of Jto (white), JtoD29A (green), and JtoR68S (red) around the sites of mutations, residues 29 and 68. Side-chain conformational changes are observed at N29, D29, S68b and R68. Notice in JtoR68S (red) that N29 moves to form a new hydrogen bond with D29. The figure was prepared using the program Insight II (Biosym Technologies, San Diego).

carboxyl oxygen of Asp29 and the side-chain nitrogen atom of Asn31 (Figure 3.3). The hydroxyl group of the substituted Ser68 is solvent exposed and not involved in hydrogen bonding interactions with other parts of the protein. In the absence of the hydrogen bond donor Asp29, the guanadinium side-chain of Arg68 in JtoD29A, undergoes a conformational rearrangement and the guanadinium group points towards solvent as compared to Jto due to the disruption of the ionic interaction (Figure 3.3). In the vicinity of the mutation in JtoD29A, we do not observe the same side-chain conformational changes for Asn29 and Ser68b as observed for JtoR68S (Figure 3.3). Instead they adopt a similar conformation to the corresponding residues in the Jto structure. Slight differences in side-chain conformation were found in several regions of the JtoD29A and JtoR68S structures. Interestingly, we found that most of these differences were localized to the surface of the molecules. We expected that these conformational differences would affect the solvent exposed surface area of the protein. To test this we next examined both the hydrophobic surface area and the electrostatic surface potential.

Comparison Of Hydrophobic Surface Areas

As JtoD29A and JtoR68S crystallize in the same space group with identical unit cell dimensions, which implies similar crystal packing effects, we are able to do detailed comparisons such as hydrophobic surface accessible area and electrostatic potential surface areas with confidence. A comparison of the hydrophobic surface between the JtoR68S versus JtoD29A may offer some clues to explain the differing kinetics of fibril formation observed for these two mutants. We used the software package ACCESS (Hubbard and Thorton 1993) to calculate the solvent accessibilities for both the

JtoR68Sand JtoD29A structures. If Gly, Ala, Val, Ile, Leu, Met and Phe are used as the hydrophobic residues, there is a 42.5 $Å^2$ increase in solvent accessibility, indicating that a larger hydrophobic surface area is exposed in JtoR68S as compared to the JtoD29A structure.

A less conservative approach to looking at non-polar surfaces as compared to using strict hydrophobic residues alone will involve the inclusion of apolar atoms from all side-chains. Using this method we have generated an apolar surface area of JtoR68S, which includes the apolar atoms possessing greater accessible surface area as compared to JtoD29A (see Figure 3.4). Residues 5, 7, 14, 15, 18, 19, 20, 27A, 29, 32, 61, 63, 79 and 94 have higher apolar surface accessibilities in JtoR68S as compared to JtoD29A (Figure 3.4). It is important to note that 64% of these residues belong to the first N-terminal β-hairpin turn spanning from residues 1-30. Moreover, this finding is consistent with the results obtained from a difference distance matrix plot (DDMP) of JtoR68S versus JtoD29A shown in Figure 3.5. From Figure 3.5 it appears that the largest distance differences are found in two clusters. The first cluster is found between residues 4-28 and the second cluster is between residues 64-84. These same differences are also found in difference distance matrix plot between JtoD29A and Wil (data not shown). As mentioned above the first cluster lies within the first 30 residue N-terminal β-hairpin loop that has a majority of the exposed apolar atoms (Figure 3.4 a and b). Interestingly a study involving a peptide derived from homologous residues 74-80 of cluster 2 from a $V_κ4$ LC protein LEN was shown to inhibit fibril formation (Davis, Raffen et al. 2000). It is

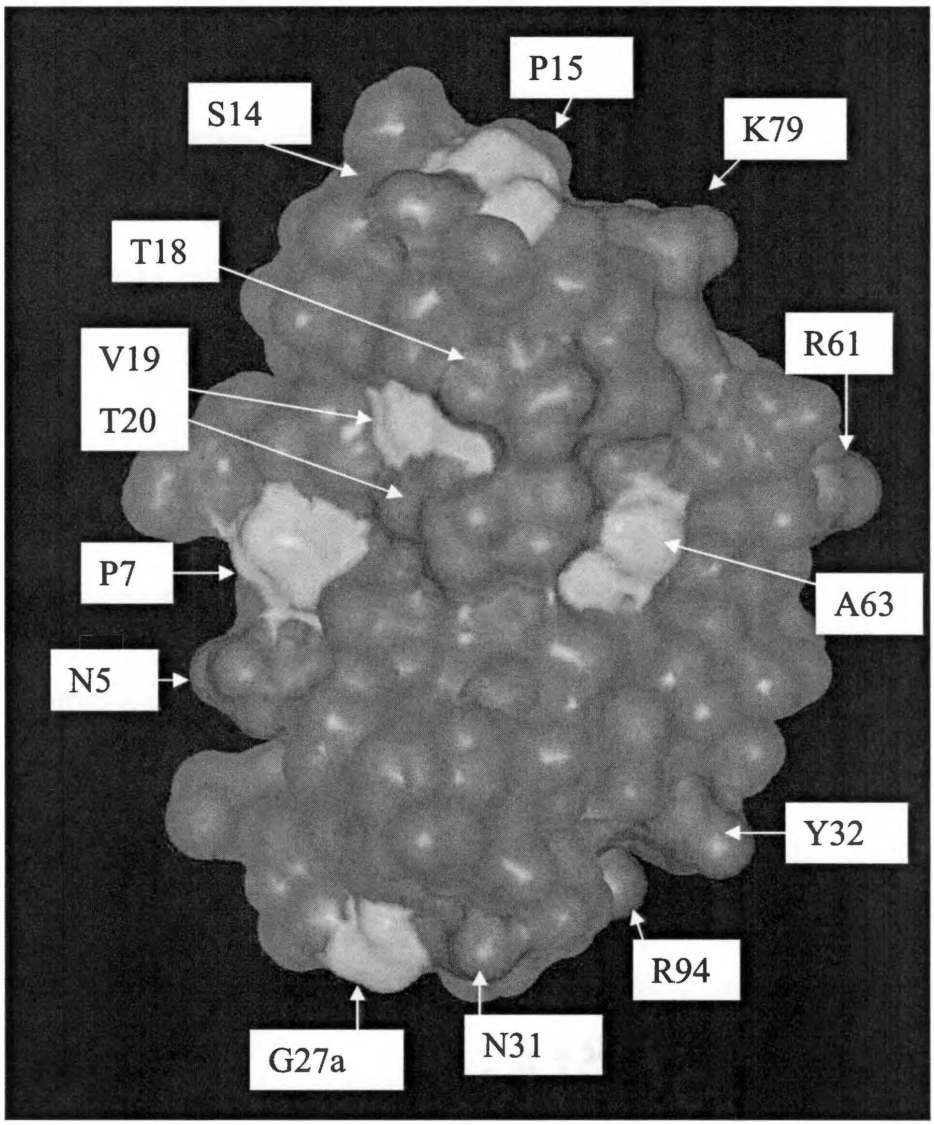


Figure 3.4. A comparison of accessible apolar surface area between JtoD29A and JtoR68S. **(a)** The surface area was calculated for JtoD29A, and JtoR68S using the program ACCESS (Hubbard and Thorton 1993). Shown in green are hydrophobic residues which are more exposed in the JtoR68S structure when compared to JtoD29A. Blue residues denote apolar regions of polar residues that contribute a greater amount of apolar surface in JtoR68S as compared to JtoD29A. The figure was prepared using the program Insight II (Biosym Technologies, San Diego).

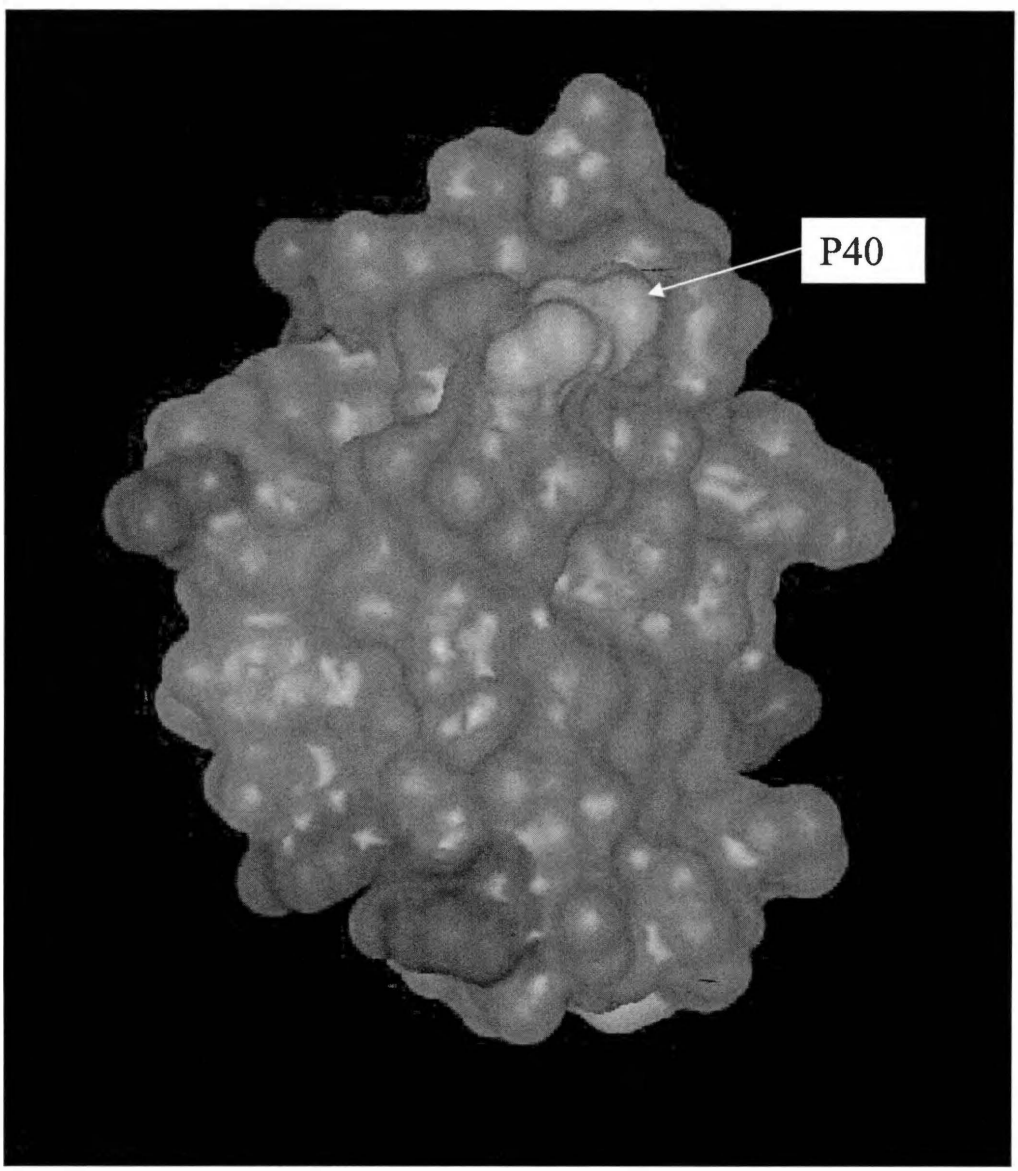


Figure 3.4 continuing from Figure 3.4 **(b)** 180° rotation of the structure shown in Figure 3.4a. Notice that most of the hydrophobic surface is localized to the opposite side of the protein. Shown here is one residue, Pro40, which has an increase in exposure in JtoR68S.

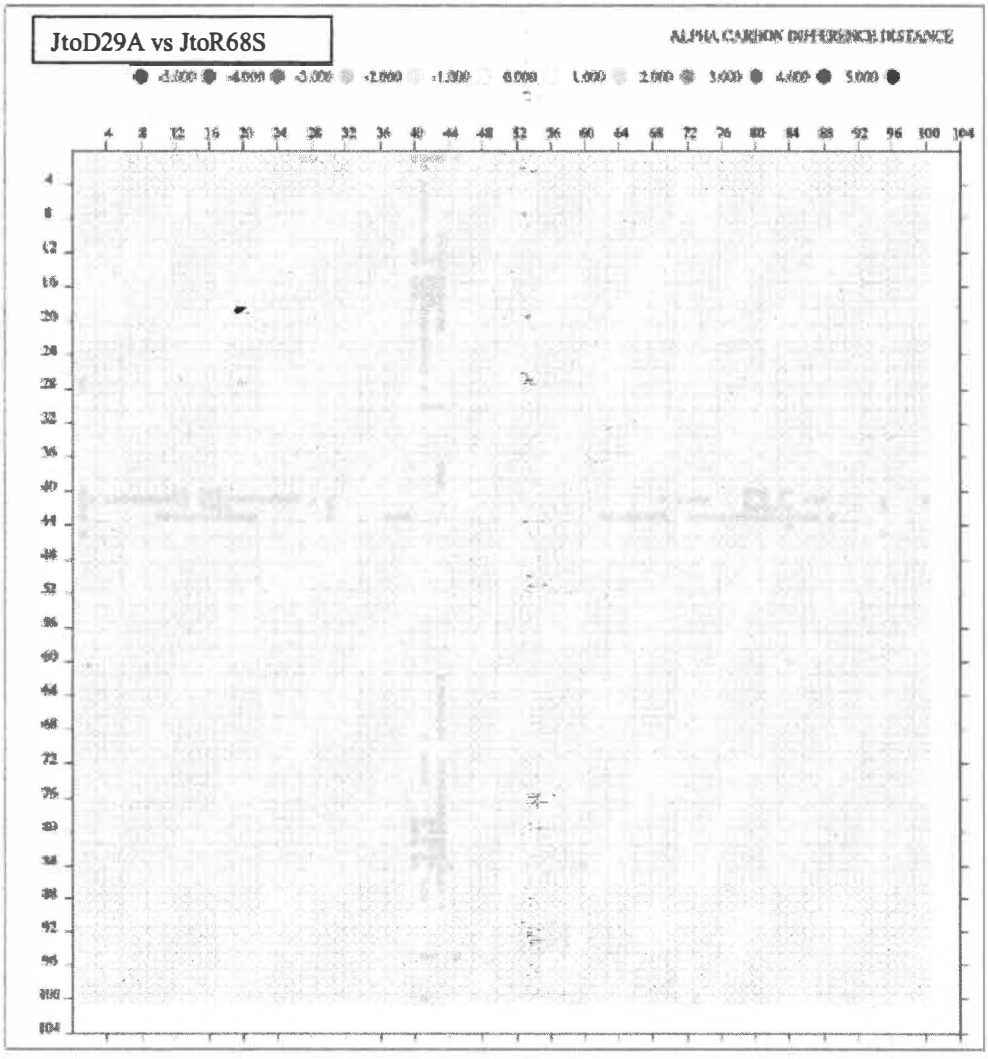


Figure 3.5. Difference Distance Matrix Plots were calculated using the program DDMP (DDMP). To calculate differences, the Cα distance of JtoR68S was subtracted from the Cα distance of JtoD29A. Plot of JtoD29A versus JtoR68S shows the greatest differences in Cα distances are found in two regions that include residues 4-28 and 64-84. As the region consisting of residues 39-42 in JtoD29A was disordered in the structure we had to model this region based on the partially observed electron density and by using the Jto5 and Jto structures as guides prior to calculating the difference distance plot.

worthwhile noting that cluster 2 consists of a strand that interacts with the above mentioned N-terminal β-hairpin loop containing the majority of the non-polar groups shown in Figure 3.4. Interestingly in another study, residues 29, 31, 61, 72 and 74 have been identified as risk factors (Stevens 2000). From these results we expect that both the first amino terminal β-hairpin and the β-strand comprising framework 4 may be susceptible spots for the formation of amyloid fibrils.

Electrostatic Potential Surface Analysis

There have been several studies that have shown correlation between surface charge of proteins and fibril formation (Gallo, Goni et al. 1996) (Tjernberg, Hosia et al. 2002) (Pokkuluri, Gu et al. 2002). Since this study involves the structural characterization of two mutations that disrupts an ionic interaction due to the replacement of an acidic and basic residue by two neutral residues, we compared their respective electrostatic potential surfaces generated by the program Delphi (Rocchia, Alexov et al. 2001). The Jto protein has an overall charge of –2, while the mutation (Asp29->Ala) in JtoD29A reduces the overall charge to -1, and the mutation (Arg68->Ser) in JtoR68S will increase the overall charge of the protein to -3. The respective electrostatic potential surfaces are shown in Figure 3.6. The effects of the mutations in JtoD29A and JtoR68S can be readily seen from the electrostatic potential surfaces. In Jto, the ionic interaction involving Arg68 represented by a blue positively charged area, and Asp29 represented by a red negatively charged area is readily observed in Figure 3.6a. Upon mutation of Asp29 to an Ala in JtoD29A, a decrease is observed in the negative electrostatic potential surface (see Figure 3.6b), while in JtoR68S, the Arg68 to a Ser mutation results in an

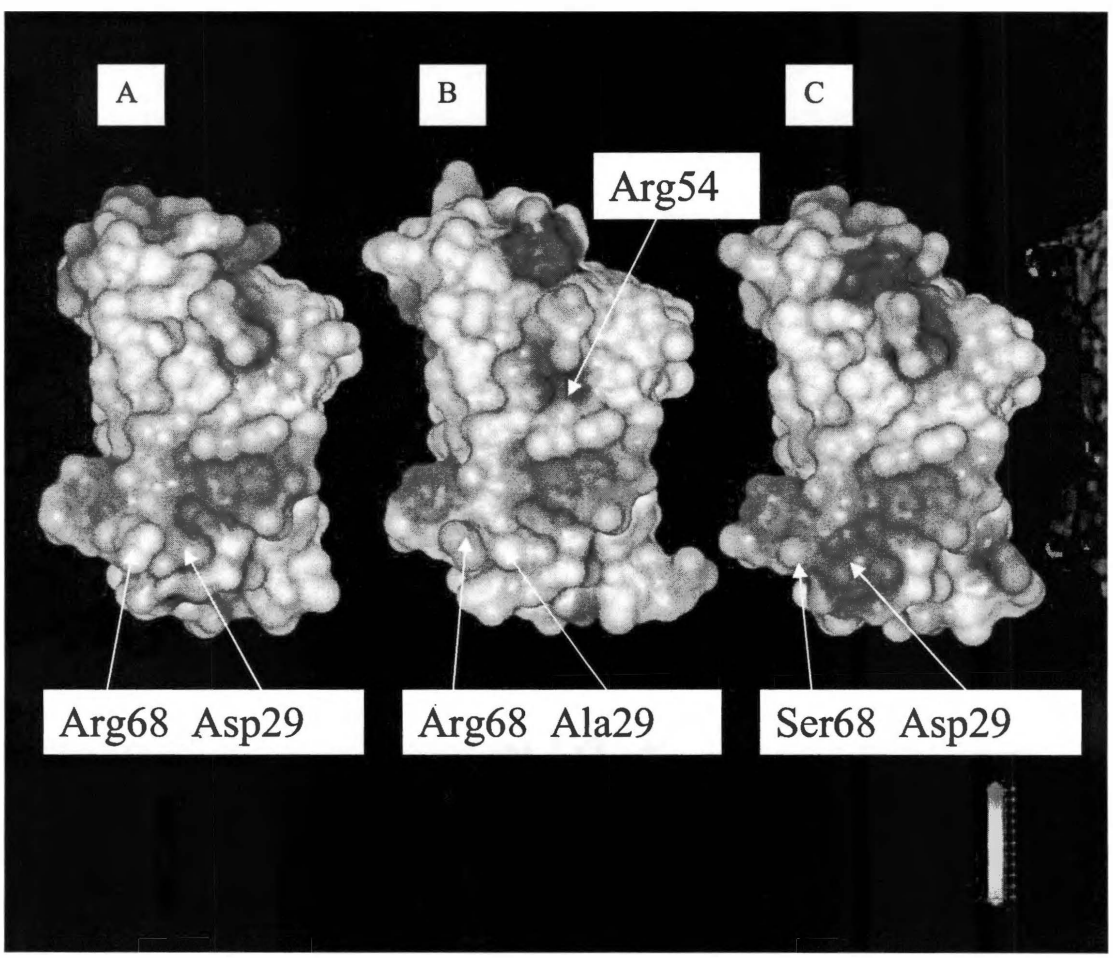


Figure 3.6. Electrostatic potential surface for (A) Jto, (B) JtoD29A, (C) JtoR68S were calculated using the program Delphi (Rocchia, Alexov et al. 2001). The positively charged residues are shown in blue while the negatively charged residues are shown in red, and neutral charged residues are shown in white. The figure was prepared using the program Insight II (Biosym Technologies, San Diego).

increase of the red negatively charged electrostatic potential surface (see Figure 3.6c). Interestingly, another group found that the mutation of a neutral glutamine residue at position 89 to an Asp increased the propensity for a κ4 LC protein LEN, to form fibrils (Pokkuluri, Gu et al. 2002). The increase in negative charge contributed by the Asp89 was proposed to destabilize the structure of the LEN. Moreover, the study shows that a structurally related protein, Myelin Protein Zero (MPZ), contains Asp99 in an equivalent position that is involved in an ionic interaction with Arg38. The ionic interaction in MPZ contributes to stability and impairs fibril formation. This is closely related to what is observed in this study. In Jto, Asp29 and Arg68 forms an ionic interaction, which is analogous to the stable MPZ ionic interaction. In addition, the increase in JtoR68S's negative electrostatic potential surface may increase the propensity to form fibrils reminiscent to a second study (Tjernberg, Hosia et al. 2002), in which tetra peptides derived from Aβ demonstrated increased fibrilogenesis as charged residues were substituted for apolar ones. Figure 3.6 also shows that JtoD29A has a slight increase in positive surface area around Arg54. This difference in charged surface area is attributed to a guanadinium side-chain conformational change of Arg54, which is only found in JtoD29A. This further demonstrates the wide-ranging effects on side-chain conformations resulting from a single amino acid mutation.

Side-Chain Differences Observed In Wil Compared To The Jto Structures

We have found that the unique ionic interaction in Jto, which is accredited with the increase in structural stability of the protein and prevention of amyloid fibrils, does not completely account for fibrillogenic properties. The main-chain of Wil superposes

well with the Jto structures where the *rms* differences is only 0.8 Å *rms* (Figure 3.2.). As shown in Figure 3.7, the side-chains of Tyr36, Tyr88, and Gln89 have the most dramatic conformational differences between the Jto structures and Wil. The origin of these conformational differences possibly results from the short hydrophobic Val96 in the Jto being substituted by a much longer polar Gln residue in Wil. This substitution causes the side-chain of Gln89 to undergo a conformational change in Wil (Figure 3.7). None of the Jto-based structures could accommodate the Gln89 side-chain conformation adopted by Wil, as this would lead to a steric clash with the tyrosyl side-chain of residue 36 (Figure 3.7). For instance, the ND1 side-chain atom of Gln89 of Wil almost occupies the exact position of the aromatic ring of Tyr36 in Jto, JtoD29A and JtoR68S (Figure 3.7). Such structural perturbations are responsible for Tyr36 in Wil to adopt a different rotamer as compared to its position in the Jto-based structures (Figure 3.7). The side-chain conformational difference of Gln89 between Wil and the Jto-based structures is also responsible for displacing the tyrosyl ring of residue 88. Close examination of the four structures show that Tyr88 in Wil will sterically interfere with the side-chain positions of Gln89 in the Jto-based structures. Therefore Tyr91 in the Jto-based structures have their tyrosyl side-chain displaced towards solvent alleviating steric clashes. Surface side-chain interactions have been shown to affect stability of protein (Davis, Raffen et al. 2000). The seventeen amino acid substitutions found in Wil as compared to the Jto-based proteins are responsible for the differing surface side-chain interactions, which may also contribute to Wil being the least thermodynamically stable $V_\lambda 6$ LC protein out of the four.

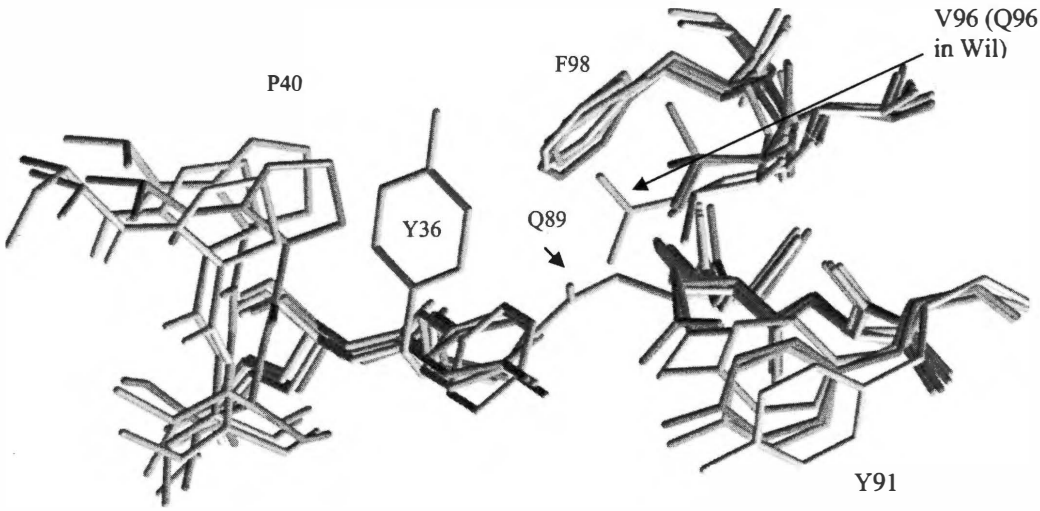


Figure 3.7. Vast side-chain conformational differences are observed in Wil when compared to the Jto, JtoD29A and JtoR68S structures resulting from seventeen amino acid variations in the Wil amino acid sequence. At residue 96, the substitution of a glutamine for a valine, which is found in the Jto structures, propagates sterically induced side-chain conformational changes in Q89, Y91, and Y36. The figure was prepared using the program Insight II (Biosym Technologies, San Diego).

Chapter 4: Determination of the structure of JtoR68S at pH 4.0 and pH 3.0

It has been shown previously that fibril formation can be induced by mild incubations in denaturing conditions (Chiti, Mangione et al. 2001) (Chiti, Webster et al. 1999). These denaturants can be chaotropic agents, temperature, and low pH. However, total denaturation of the protein inhibits the formation of fibrils (Chiti, Webster et al. 1999). It appears that the protein must be partially unfolded by these denaturing agents to promote the formation of the fibrils (Chiti, Mangione et al. 2001) (Chiti, Webster et al. 1999) (Khurana, Gillespie et al. 2001) (Dobson and Evans 1988). It was also shown that decreasing the pH by small steps permits the trapping of intermediates, which are on the path to forming fibrils (Chiti, Mangione et al. 2001). These findings indicate that a protein can be induced into forming an intermediate. Thus, decreasing the pH of the mother liquor may allow for conformational changes of the protein, while crystal contacts will prevent total unfoding. Crystals may be soaked in low pH buffers in a time dependent manner in order to observe structural intermediates.

JtoR68S Crystals Incubated At pH4.0

JtoR68S crystals were grown in 0.8 M ammonium sulfate, 0.1 M sodium acetate pH 5.0, which is the buffer of the initial JtoR68S crystals that are described in the preceding chapter. These crystals were subjected to X-ray diffraction experiments to ensure quality of crystals and isomorphism. The crystals belonged to an orthorhombic crystal system. Originally both tetragonal and orthorhombic crystals were obtained by Dr. Gupta. However, I was only able to crystallize the orthorhombic crystal forms. After the quality of the crystals were assessed to be good, an initial soaking experiment was

performed to test the longevity of the crystal at pH 4.0. A single crystal was incubated in 0.8 M ammonium sulfate, 0.1 M sodium acetate pH 4.0 until the crystal cracked and dissolved. This occurred after 45 minutes of soaking. After this, we performed a 5, 10, and 30 minute soak in the pH 4.0 buffer. After the soaks were completed crystals were incubated in a cryo-protectant containing 0.8 M ammonium sulfate, 0.1 M sodium acetate pH 4.0, 25% glycerol. X-ray diffraction data were collected on each crystal at −160°C.

The unit cell dimensions of the JtoR68S crystal at pH 4.0 were a=70.544Å, b=72.844Å, c=96.284Å, $\alpha=\beta=\gamma=90.0°$. These cell dimensions differ from the native crystal on two axes, the a and the c, which are 72.255Å and 94.255Å respectively (approximately 2% difference for each). The same unit cell dimensions were also found the ten minute soak. We attempted to process the data in P222 orthorhombic space group, and the space group was ultimately determined by utilizing pseudo precession pictures using the CCP4 script HKLVIEW. We found that the space group of both structures was $P2_12_12_1$. Both structures were solved by molecular replacement using the programs AMoRe and MOLREP (Vagin and Teplayakov 1997) using the CCP4 program suite. The R_{sym}s of the five and ten minute soaks were 13.9% and 9.9%, respectively. After one round of positional refinement in CNS the R_{free} for the five minute soak was 34.5%. Afterwards, a 2Fo-Fc map was calculated for analysis of the model. We examined the 2Fo-Fc map to locate regions that had undergone conformational changes. However, the electron density revealed no such changes. The model from molecular replacement for the ten minute soak was used for one round of positional refinement in CNS where the R_{free} was 36.6%. Data collection and refinement statistics are shown

Table 4.1. Statistics and unit cell dimensions of the JtoR68S pH 4.0 and 3.0 crystal structures.

	pH 4.0	pH 3.0
Number of molecules per asymmetric unit	2	2
R_{sym}	9.9%	21.4%
Resolution	35 – 2.5Å	35 – 2.5Å
Completion	97.6%	98.9%
R_{free}	36.6%	37.7%
Non-hydrogen protein atoms/asymmetric unit	1820	1820
Space group	$P2_12_12_1$	$P2_12_12_1$
Unit cell dimensions	a=70.54Å,b= 72.84Å, c=96.28Å	a=70.54Å,b=72.84Å, c=96.28Å
	α=β=γ=90°, 90°, 90°	α=β=γ=90°, 90°, 90°

table 4.1. A 2Fo-Fc map was generated to search for conformational changes, which were not found.

A thirty minute soak, at pH 4.0, was performed next. However, the processing of the data set from this crystal was not possible. The crystal possessed a high mosaicity, which is a measure of disorder in the crystal, and the data could not be properly scaled or processed. This indicates that the crystal had become greatly disordered due to the length of the soak at pH 4.0. It was apparent that pH 4.0 would induce a change in the crystal, but we were not able to see any changes at short soak, while a longer soak resulted in an unscalable data set. At this point we decided to lower the pH even further and perform a short soak to try to promote conformational changes in the protein while maintaining the crystalline state.

JtoR68S Crystals Incubated At pH 3.0

JtoR68S crystals were subjected to an initial soak at pH 3.0 in 0.8 M ammonium sulfate, 0.1 M glycine pH 3.0 to determine longevity of the crystal at this pH. We found that the crystals would start to crack after fifteen minutes in the pH 3.0 buffer. We next subjected a single crystal to a five minute soak at pH 3.0. Afterwards, cryoprotection of the crystal was achieved by a short incubation in 0.8 M ammonium sulfate, 0.1 M glycine pH 3.0, 25% glycerol. An X-ray diffraction data set was collected on this crystal at -160°C.

After the five minute soak the crystal was still isomorphous with cell dimensions of: a=71.399Å, b=72.140Å, c=96.053Å, $\alpha=\beta=\gamma=90°$. The structure was solved by molecular replacement using the CCP4 program MOLREP (Vagin and Teplayakov 1997). The resulting R_{sym} for the data was 21.4%. After a single round of positional refinement in the CNS the R_{free} value was 41.1% (Table 4.1). A 2Fo-Fc map was generated to search for conformational differences. Several side-chains of residues were observed to have undergone conformational change. These residues were at positions His8, Ser9, Lys16, Thr46, Ser72, Asp82, Arg94, Asn95, Val96, Val97, and Phe98. However, main-chain residues 94 to residue 98 were disordered (shown in Figure 4.1). Interestingly these residues comprise the bulk of CDR-3. However, the CDRs display a high degree of sequence variability (Chothia and Lesk 1987), and in the case of Jto, the stabilization of CDR1 by an ionic interaction between Asp29 and Arg68 prevents the formation of amyloid fibrils *in vivo* (Pokkuluri, Solomon et al. 1999). Since the CDRs are sites of high variability and in Jto the ionic interaction to CDR1 appears to play a role in preventing fibril formation, the conformational change of a CDR may be an early

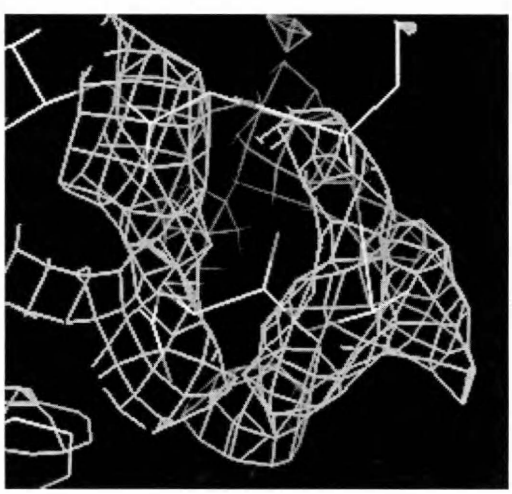


Figure 4.1. 2Fo-Fc electron density map shown with the model of the JtoR68S structure at CDR3 after one round of positional refinement in CNS. Notice that main-chain Cα density is missing.

event in partial unfolding of a light chain. A simulated annealing omit map, omitting residues 90 to 95 was calculated for the molecule and used for extensive modeling of this region. A second round of positional refinement in CNS was performed resulting in a R_{free} value of 37.7%. The decrease in the R_{free}-value indicates that the modeling of this region was correct and that this region has a different main-chain conformation from the native model. The new conformation of the CDR is shown in Figure 4.2a in yellow compared to the native in red. The resulting model and 2Fo-Fc map calculated for the modeled region is shown in Figure 4.2b.

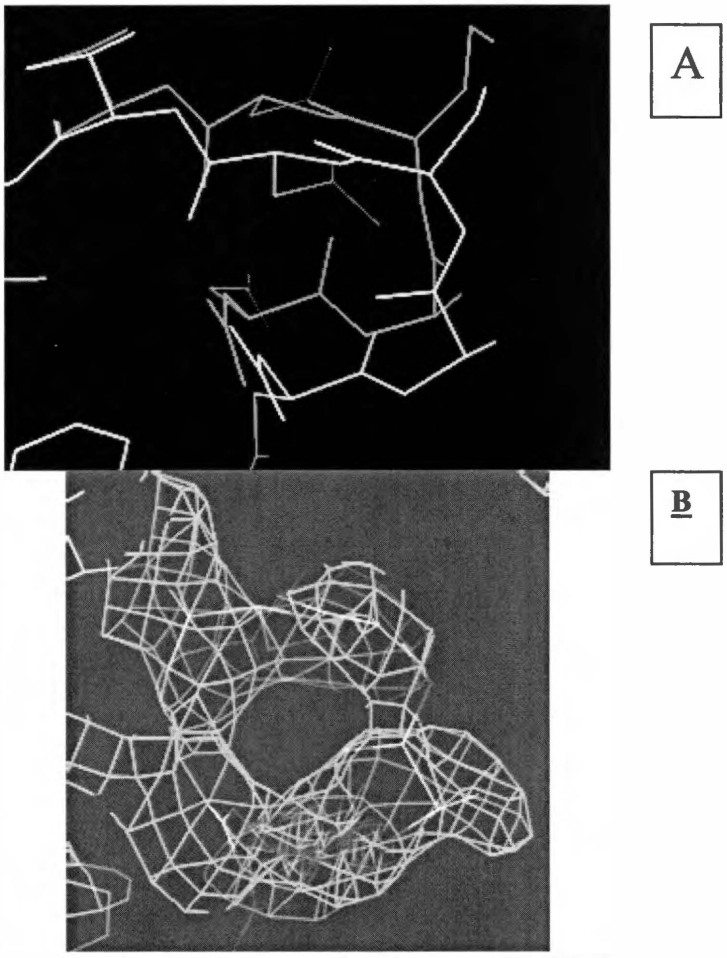


Figure 4.2. Modeling of the CDR3 region. **A.** Modeling of CDR3 shown in red compared to initial starting structure shown in yellow. **B.** A 2Fo-Fc simulated annealing omit map calculated for CDR3 shown with the model, which was built using the omit map.

Chapter 5: Conclusions and Future Directions

The JtoD29A And JtoR68S Structures

We have solved the structures of both the JtoD29A and JoR68S $V_\lambda 6$ light chain antibodies. Since both structures crystallized in the $P4_122$ space group we were able to make detailed analysis of the two structures. No significant main-chain conformational differences were observed. However, some side-chain conformational differences were observed resulting in a change in the solvent exposed surface areas. We found that the amyloidogenic JtoR68S has a greater exposure of hydrophobic surface area. Interestingly 67% of the residues contributing to this increase are localized to the first 30 residues of the protein, which comprises the first β-hairpin. Another cluster of residues is found in the FW5, which has been previously suggested as a potential "hot spot" for amyloid fibril formation. Interestingly FW5 packs against the first β-hairpin. We suggest that the first β-hairpin is in fact a site of early unfolding, which results in the formation of the amyloid nuclei. This has been independently confirmed by inhibition of fibril formation, when using the 30 residue N-terminal peptide of LEN (Wall, personal communication). This type of assay was also performed for the $V_\kappa 4$ LC LEN where a peptide composing residues from FW5, which was found to inhibit the formation of fibrils by LEN. Future work should concentrate on crystallization of the N-terminal peptide with the Light Chain molecule.

We also found that the JtoR68S structure exposed more charged surface than the Jto and the JtoR68S. Exposure of charged surface has already been shown in $V_\kappa 4$ LCs to promote the formation of fibrils. The increase in negative charge around the site of

mutation in JtoR68S may play a role in this. To access the role of charge in fibril formation, we suggest performing fibril formation assays in higher concentrations of salt to counteract the effects of protein surface charge on the formation of amyloid fibrils.

The analysis of both the JtoD29A and the JtoR68S has indicated that global structural shifts do not occur between the amyloidogenic and the non-amyloidogenic protein. However, the difference in side-chain conformations results in the exposure of a different solvent accessible surface. In this study we have demonstrated how a native structure can provide clues about the amyloid phenomenon. Though the methods we have used are not new, they have not been used before for analyzing amyloidogenic proteins.

Low pH soaks Of JtoR68S Crystals

We were unable to find any conformational differences from the five and ten minute pH4 soaks. A thirty minute soak resulted in data that was unscalable. A short five minute soak at pH3 revealed conformational differences around CDR3. Further analysis indicates that the CDR3 is not involved in crystal packing. However, the stabilizing effects of the crystal lattice may prevent other parts of the protein from undergoing conformational changes. This presents a problem in the interpretation of the unfolding event, which leads to the nuclei. To alleviate this problem, we propose to do NMR experiments under similar buffering conditions. A second problem occurs at ascertaining the exact species that could be deemed the nuclei. Since, the structure of the nuclei is not known, we do not know if the structure(s) we see are relevant. To address

this problem we plan on using the native structure and the structures we determine at different pHs to perform molecular dynamic simulations.

This work shows how small pH changes can perturb the native structure. Longer soaks and soaks in both lower and higher pH buffers will be attempted in the future. These structures will shed further light on the type of structural intermediates that might be present in the unfolding and/or fibrillogenic pathway.

References

Bellotti, V., P. Mangione, et al. (2000). "Review: Immunoglobulin Light Chain Amyloidosis-The Archetype of Structural and Pathogenic Variability." J Struct Biol **130**: 280-289.

Bellotti, V., G. Merlini, et al. (1990). "Relavance of class, molecular weight, and isolecetric point in predicting human light chain amyloidogenicity." British Journal of Haematology **74**: 65-69.

Blow, D. (2002). Outline of Crystallography for Biologists. Oxford, Oxford University Press.

Bruenger, A. T., P. Adams, et al. (1998). "Crystallography and NMR System: A new software suite for macromolecular structure determination." Acta Crystallogr D **54**: 905-921.

Bucciantini, M., E. Giannoni, et al. (2002). "Inherent toxicity of aggregates implies a common mechanism for protein misfolding diseases." Nature **416**(6880): 507-11.

Cardoso, I., G. Merlini, et al. (2003). "4'-iodo-4'-deoxydoxorubicin and tetracyclines disrupt transthyretin amyloid fibrils in vitro producing noncytotoxic species: screening for TTR fibril disrupters." Faseb J **17**(8): 803-9.

Chiti, F., P. Mangione, et al. (2001). "Detection of two partially structured species in the folding process of the amyloidogenic protein beta 2-microglobulin." J Mol Biol **307**(1): 379-91.

Chiti, F., P. Webster, et al. (1999). "Designing conditions for in vitro formation of amyloid protofilaments and fibrils." Proc Natl Acad Sci U S A **96**(7): 3590-4.

Chothia, C. and A. M. Lesk (1987). "Canonical structures for the hypervariable regions of immunoglobulins." J Mol Biol **196**(4): 901-17.

Cohen, A. S. and E. Calken (1959). Nature **183**: 1202-1203.

Cohen, A. S., T. Shirahama, et al. (1982). Electron Microscopy of Proteins. I. Harris. London, Academic Press. **3:** 165-205.

Collaborative Computational Project, N. (1994). "The CCP4 Suite: Programs for Protein Crystallography." Acta Crystallogr D **50**: 760-763.

Davis, D., R. Raffen, et al. (2000). "Inhibition of Amyloid Fiber Assembly by both BiP and Its Target Peptide." Immunity **13**: 433-442.

DDMP "Difference distance matrix plots were produce using the DDMP program from the

Center for Structural Biology at Yale University, New Haven, CT.".

Delabar, J. M., D. Goldgaber, et al. (1987). "Beta amyloid gene duplication in Alzheimer's disease and karyotypically normal Down syndrome." Science **235**(4794): 1390-2.

Dobson, C. M. (1999). "Protein misfolding, evolution and disease." Trends Biochem Sci **24**(9): 329-32.

Dobson, C. M. and P. A. Evans (1988). "Protein structure .Trapping folding intermediates." Nature **335**(6192): 666-7.

French, S. and K. Wilson (1978). "On the Treatment of Negative Intensities." Acta Crystallogr A **34**: 517-525.

Gallo, G., F. Goni, et al. (1996). "Light chain cardiomyopathy. Structural analysis of the light chain tissue deposits." Am J Pathol **148**(5): 1397-406.

Garvito, R. M. (1986). "Detergent for screening crystal conditions." J Crystal Growth **76**: 701-709.

Gupta, V., M. Wilkerson, et al. (2003). "The three-dimensional X-ray structures of two variants Jto5 and Jto2 of the lamda 6 LC protein Jto: A structural study of how the disruption of a salt-bridge by replacement of acidic and basic groups results in opposite amyloidogenic properties." JMB **to be submitted**.

Helms, L. R. and R. Wetzel (1996). "Specificity of abnormal assembly in immunoglobulin light chain deposition disease and amyloidosis." J Mol Biol **257**(1): 77-86.

Hubbard, S. J. and J. M. Thorton (1993). NACCESS. Department of Biochemistry and Molecular Biology, University College London.

Hurle, M., L. Helms, et al. (1994). "A role for destabilizing amino acid replacements in light-chain amyloidysis." Proc Natl Acad Sci U S A **91**: 5446-5450.

Jaikaran, E. T. and A. Clark (2001). "Islet amyloid and type 2 diabetes: from molecular misfolding to islet pathophysiology." Biochim Biophys Acta **1537**(3): 179-203.

Jones, T. A., J. Y. Zou, et al. (1991). "Improved Methods for building protein models in electron density maps and the location of errors in these models." Acta Crystallogr A **47**: 110-119.

Khurana, R., J. R. Gillespie, et al. (2001). "Partially folded intermediates as critical precursors of light chain amyloid fibrils and amorphous aggregates." Biochemistry **40**(12): 3525-35.

Krishnan, S., E. Chi, et al. (2003). "Oxidative Dimer Formation is the Critical Rate-Limiting Step for Parkinson's Disease alpha-synuclein Fibillogenesis." Biochemistry **42**(829-837).

Kruger, R., O. Eberhardt, et al. (2002). "Parkinson's disease: one biochemical pathway to fit all genes?" Trends Mol Med **8**(5): 236-40.

Ladd, M. F. and R. A. Palmer (1985). Structure Determination by X-ray Crystallography. New York, Plenum Press.

Lansbury, P. (1999). "Evolution of amyloid: What normal protein folding may tell us about fibrillogenesis and disease." Proc Natl Acad Sci U S A **96**: 3342-3344.

Lee, B. and F. M. Richards (1971). "The interpretation of protein structures: the estimation of static accessibility." J Mol Biol **55**: 379-400.

Merlini, G., V. Bellotti, et al. (2001). "Protein Aggregation." Clin Chem Lab Med **39**: 1065-1075.

Murshudov, G., E. Vagin, et al. (1996). Application of Maximum Likelihood Refinement. the Refinement of Protein structures, Proceedings of Daresbury Study Weekend.

Navaza, J. (1994). "AMoRe: an automated packagefor molecular replacement." Acta Crystallogr A **50**: 157-163.

Otwinowski, Z. (1992). An Oscillation Data Processing Suite for Macromolecular Crystallography. New Haven, CT.

Ozaki, S., M. Abe, et al. (1994). "Preferential expression of human lambda-light-chain variable-region subgroups in multiple myeloma, AL amyloidosis, and Waldenstrom's macroglobulinemia." Clin Immunol Immunopathol **71**(2): 183-9.

Pokkuluri, P. R., M. Gu, et al. (2002). "Factors contributing to decreased protein stability when aspartic acid residues are in Beta sheet regions." Prot Sci **11**: 1687-1694.

Pokkuluri, P. R., A. Solomon, et al. (1999). "Tertiary structure of human lambda 6 light chains." Amyloid **6**(3): 165-71.

Raffen, R., L. Diekman, et al. (1999). "Physiochemical consequences of amino acid variations that contribute to fibril formation by immunoglobulin light chains." Prot Sci **8**: 509-517.

Rocchia, W., E. Alexov, et al. (2001). "Extending the applicability of the nonlinear Poisson-Boltzmann equation: Multiple dielectric constants and multivalent ions." Journal of phys chem B **105**(28): 6507-6514.

Rochet, J. and P. Lansbury (2000). "Amyloid fibrillogenesis: themes and variations." Curr Opin Struct Biol.

Rossman, M. and D. Blow (1962). "The detection of sub-units within the crystallographic asymmetric unit." Acta Crystallogr A **15**: 24-31.

Shirahama, T. and A. S. Cohen (1967). J. Cell Biol. **33**: 679-706.

Solomon, A., B. Frangione, et al. (1982). "Bence Jones proteins and light chains of immunoglobulins. Preferential association of the V lambda VI subgroup of human light chains with amyloidosis AL (lambda)." J Clin Invest **70**(2): 453-60.

Solomon, A., D. Weiss, et al. (1992). "Induction in of mice of human light-chain associated amyloidosis." Am J Pathol **140**: 629-637.

Soto, C. (2001). "Protein misfolding and disease; protein refolding and therapy." FEBS Lett **498**: 204-207.

Stevens, F. J. (2000). "Four structural risk factors identify most fibril-forming kappa light chains." Amyloid **7**(3): 200-11.

Sunde, M. and C. C. Blake (1997). "The structure of Amyloid Fibrils by electron microscopy andx-ray diffraction." Adv. Protein Chem **50**: 123-159.

Sunde, M., L. C. Serpell, et al. (1997). "Common core structure of amyloid fibrils by synchrotron x-ray diffraction." J. Mol. Biol. **273**: 729-739.

Temussi, P. A., L. Masino, et al. (2003). "From Alzheimer to Huntington: why is a structural understanding so difficult?" Embo J **22**(3): 355-61.

Tjernberg, l., W. Hosia, et al. (2002). "Charge attraction and Beta propensity are necessary for amyloid fibril formation from tetrapeptides." J Biol Chem **277**: 43243-43246.

Vagin, A. A. and A. Teplayakov (1997). "MOLREP: an automated program for molecular repelcement." J Applied Cryst **30**: 1022-1025.

Wall, J., M. Schell, et al. (1999). "Thermodynamic instability of human lambda 6 light chains: correlation with fibrillogenicity." Biochemistry **38**(42): 14101-8.

Watson, H. C. and J. Warwicker (1982). "Calculation of electric-potential in the active site cleft due to alpha-helix dipoles." J mol Biol **157**: 671-679.

Appendix

Programs Used

```
Auto.dat for Denzo
format raxis4 100

wavelength 1.5418
[monochromator filter]

cassette rotx 0. roty 0.

spot elliptical 0.8 0.8 0
box 2.4 2.4
background elliptical 0.9 0.9 0
overlap spot

[crossfire y  0.259 x  0.272 xy  0.057]

x beam   149.938  y beam  149.929
[skew  0.00035]
[Y scale 1.0]
film rotation  180.000
oscillation range 1.0
oscillation start 2

distance 100


mosaicity 0.9900

title  'jto5-tetragonal 10min pH4 soak'

sector 1 to 81

raw data file 'a2_40###.osc'

film   output file 'a2_40###.x'


box 2.4 2.4 0
overlap spot
[error systematic 5.0 partiality 0.1]
[error positional 0.050]
profile fitting radius 30.0

peak search file 'peaks.file'
FIT x beam y beam cell crystal rotx roty rotz

WRITE PREDICTIONS

GO

fix all

START REFINEMENT

[print no shifts no profiles]
```

```
resolution limits 100 4.0
fit  crystal rotx roty rotz
go go go
resolution limits 100 3.0
fit cell x beam y beam crossfire x y xy
fit cassette rotx roty
resolution limits 100 2.5
fit cell x beam y beam crossfire x y xy
fit cassette rotx roty

fit distance yscale skew

go go go go go

print profiles 1 1
calculate go


Scalepack

/usr/local/scalepack<<EOF>scalepack_pH4_p4122_y.log
RESOLUTION 100 2.5
NUMBER OF ZONES 10
NUMBER OF ITERATIONS 10
ESTIMATED ERROR 0.080 0.100 0.100 0.100 0.10 0.010 0.08 0.04 0.04 0.04
ERROR SCALE FACTOR 2.5
SPACE GROUP P4122
OUTPUT FILE jto5_pH4_p4122_y.sca
reference film 1
add partials  1 to 81
format DENZO_york1

[scale restrain 0.1]


POSTREFINE 10
fit crystal a* 1 to 81
fit crystal b* 1 to 81
fit crystal c* 1 to 81
fit crystal beta* 1 to 81
fit film rotx 1 to 81
fit film roty 1 to 81
fit crystal mosxx 1 to 81

rejection probability 1.e-2
write rejection file
@reject
[vertical axis 0 0 1]
[spindle axis 0 1 0]
Sector 1 to 81
FILE  'a2_40###_y.x'
EOF

Scalepack2mtz
```

```
scalepack2mtz hklin jto5_ph4_p4122.sca hklout jto5_ph4_p4122.mtz <<
eof>scalepack2mtz.log
SYMM 91
END
eof
#n example of extending data to p4122
cad     hklin1 jto5_ph4_p4122.mtz   hklout jto5_p4122_ph4_cad.mtz
<<eof-cad > cad.log
SYMMETRY 91
LABIN FILE 1  E1=IMEAN E2=SIGIMEAN
END
eof-cad
sortmtz HKLIN jto5_p4122_ph4_cad.mtz  HKLOUT
jto5_p4122_ph4_cad_sort.mtz  << EOF-sortmtz > sort.log
#
# Sort keys since default keys are H K L
#
H K L
EOF-sortmtz
truncate hklin jto5_p4122_ph4_cad_sort.mtz hklout
jto5_p4122_ph4_trunc.mtz <<EOF-trunc > truncate_p4122.log
title truncate after scalepack o/p through cad
wilson
resolution  58 2.5
#rscale      4.0 2.0
nresidues   110
labout  F=FP SIGF=SIGFP
EOF-trunc
~

AmoRe Roting


#    sorting run:
#   ##############
#
#  mtz file contains cell and symmetry.
#
amore hklin jto5_p4122_trunc.mtz  hklpck0 spmipch.hkl <<eof>roting.log
TITLE   ** spmi  packing h k l F for crystal**
SORTFUN RESOL 100.  3.
LABI FP=FP  SIGFP=SIGFP
eof
#
#    tabling run:
#   ###############
#
#  rotate and shift coordinates and produce table:
#    xyzout is the rotated and shifted coordinates.
#
amore  xyzin1  ../auto/mono_renum.pdb xyzout1 searchrot.pdb TABLE1
search.tab << eof
TITLE :  Produce table for MODEL FRAGMENT
VERBOSE
TABFUN
```

```
CRYSTAL ORTH 1 CELL_CRYS  72.25    72.25   94.25
MODEL 1 BREPLACE 0 BADD 0
SAMPLE 1 RESO 3 SHANN 3.0 SCALE       4.0
eof
#
#
#
#    roting run:
#   #############
#
amore  TABLE1 search.tab  HKLPCK1 search.hkl hklpck0 spmipch.hkl clmn1
search.clmn clmn0 spmipch.clmn ROTING_MI 400000  ROTING_MC 800000
MAPOUT amore_cross.map <<  eof
ROTFUN
VERB
TITLE : Generate HKLPCK1 from MODEL FRAGMENT   1
GENE 1   RESO 20.0 3.0  CELL_MODEL  68.018  58.388  59.037
CLMN CRYSTAL ORTH  1 FLIM 0.E0 1.E8  SHARP 0.0 RESO  100.0  3.0 -
SPHERE  0.0 23.738
CLMN MODEL 1  FLIM 0.E0 1.E8    RESO  100.0  3.0 SPHERE  0.0 23.738
ROTA  CROSS  MODEL 1  BESLIMI 6 126 STEP 2.5 PKLIM 0.5  NPIC 100
eof

AmoRe Traing

#
amore  TABLE1 search.tab HKLPCK0 spmipch.hkl MAPOUT
amore_transjunk5.map <<  eof > traing_p4122.log
TRAFUN CB   NMOL 1 RESO 20 3. PKLIM 0.5  NPIC 10
SYMM P4122
HKLM 219 219 219  25
VERB
TITLE :jto5
CRYSTAL ORTH  1 FLIM 0.E0 1.E8  SHARP 0.0
SOLUTIONRC    1   18.59   47.43  291.35  0.0000  0.0000  0.0000 10.9
0.0   1
 SOLUTIONRC    1   71.41  132.57  111.35  0.0000  0.0000  0.0000 10.9
0.0   2
 SOLUTIONRC    1   85.67   78.00  183.88  0.0000  0.0000  0.0000  7.6
0.0   3
 SOLUTIONRC    1    4.33  102.00    3.88  0.0000  0.0000  0.0000  7.6
0.0   4
 SOLUTIONRC    1   28.57   47.95  299.00  0.0000  0.0000  0.0000  7.2
0.0   5
 SOLUTIONRC    1   61.43  132.05  119.00  0.0000  0.0000  0.0000  7.2
0.0   6
 SOLUTIONRC    1   69.63   73.89  249.36  0.0000  0.0000  0.0000  6.5
0.0   7
 SOLUTIONRC    1   20.37  106.11   69.36  0.0000  0.0000  0.0000  6.5
0.0   8
 SOLUTIONRC    1   45.44   31.86  200.38  0.0000  0.0000  0.0000  6.1
0.0   9
 SOLUTIONRC    1   44.56  148.14   20.38  0.0000  0.0000  0.0000  6.1
0.0  10
```

```
 SOLUTIONRC    1   61.32   58.76   74.15  0.0000  0.0000  0.0000  6.0
0.0  11
 SOLUTIONRC    1   28.68  121.24  254.15  0.0000  0.0000  0.0000  6.0
0.0  12
 SOLUTIONRC    1   44.09   45.74  303.08  0.0000  0.0000  0.0000  6.0
0.0  13
 SOLUTIONRC    1   45.91  134.26  123.08  0.0000  0.0000  0.0000  6.0
0.0  14
 SOLUTIONRC    1    8.37  104.77  348.30  0.0000  0.0000  0.0000  5.7
0.0  15
 SOLUTIONRC    1   81.63   75.23  168.30  0.0000  0.0000  0.0000  5.7
0.0  16
 SOLUTIONRC    1   86.00   28.50   49.00  0.0000  0.0000  0.0000  5.7
0.0  17
 SOLUTIONRC    1    4.00  151.50  229.00  0.0000  0.0000  0.0000  5.7
0.0  18
 SOLUTIONRC    1   19.19   17.16  200.69  0.0000  0.0000  0.0000  5.6
0.0  19
 SOLUTIONRC    1   70.81  162.84   20.69  0.0000  0.0000  0.0000  5.6
0.0  20
 SOLUTIONRC    1    6.00  151.55  233.50  0.0000  0.0000  0.0000  5.6
0.0  21
 SOLUTIONRC    1   84.00   28.45   53.50  0.0000  0.0000  0.0000  5.6
0.0  22
 SOLUTIONRC    1   14.32   97.83  195.72  0.0000  0.0000  0.0000  5.5
0.0  23
 SOLUTIONRC    1   75.68   82.17   15.72  0.0000  0.0000  0.0000  5.5
0.0  24
eof

AmoRe Fiting

#
#    fiting run:
#   ##############
#
amore  TABLE1 search.tab HKLPCK0 spmipch.hkl <<eof > fiting_2mol.log
FITFUN  NMOL 2  RESO 20 3.0  ITER 10 CONV   1.E-3
TITLE *** jto5   structure ***
VERBOSE
CRYSTAL ORTH  1 FLIMI 0.E0 1.E8   SHARP 0.0
HKLM 219 219 219  25
REFSOL   AL     BE   GA     X   Y    Z   BF
SOLUTIONTF1  1   18.59   47.43  291.35  0.8728  0.3536  0.4240  0.0
0.0   2
SOLUTIONTF2  1   71.41  132.57  111.35  0.9038  0.3781  0.8315 40.4
51.2   2
eof

REFMAC

#!/bin/csh -f
#
/software/ccp4-4.2.1/bin/refmac5 HKLIN jto5_p4122.mtz \
HKLOUT refmac_cycle1.mtz \
```

```
XYZIN jto5_refmac.pdb \
XYZOUT jto5_refmac_out.pdb \
TLSIN    refmac_tls.in \
TLSOUT refmac_tls_cycle2a.in \
<< eor>t.log
#
#####   Makecif parameters
#       Do not define links automatically
#       Do not add hydrogens
MAKE LINK N
MAKE HYDR N
#
#
#   Input mtz labels
#
LABIN FP=F SIGFP=SIGF FREE=FreeR_flag
#
#   Output mtz labels
#
LABOUT FC=FC PHIC=PHIC FWT=2FOFCWT PHWT=PH2FOFCWT DELFWT=FOFCWT
PHDELWT=PHFOFCWT
#
#   Parameters of bulk solvent based on constant value. They are
default
#
SOLVENT VDWP 1.4 IONP 0.80 RSHR 0.8
#
#  Weighting X-ray and geometry. Fairly tight restrants
#
WEIG MATR 0.1
#
#   This keyword says program that use 20 cycles of tls before
#   going to individual atomic reifnement. Parameters of rigid
#   body groups are defined in TLSIN tls.in
REFI TLSC 15
#
#  It is good idea to set all B values to some value prior to
#  TLS refinement
#  - do it externally so additional cycles use the refined B-values
BFACtor SET_to 30
#
#   Use restrained refinement after TLS. All reflections
#   will be used
REFInement TYPE RESTrained RESOlution 8 2
#
#   Maximum likelihood refinement
#
REFInement RESIdual MLKF
#
#   After TLS use istotropic B values refinement. In very low
resolution

#   it could be set to Overall B value refinement

#
```

```
REFInement BREFinement ISOTropic
#
#   Damp down shifts
#
DAMPing_factor 0.5 0.5
#
#   First and last cycles will print information about outliers
#   In all other cycles only overall statistics will be monitorred
#
MONI MEDI
#
# Bulk solvent plus anisotropic scaling
#
SCAL TYPE BULK
SCAL LSSC ANISOT
#
#   Number of individual atomic refinement 8

#
NCYC 10
#
END
#
eor

CNS Minimize

{+ file: minimize.inp +}
{+ directory: xtal_refine +}
{+ description: Crystallographic conjugate gradient minimization
refinement +}
{+ authors: Axel T. Brunger, and Paul D. Adams +}
{+ copyright: Yale University +}

{+ reference: A.T. Brunger, The Free R Value: a Novel Statistical
              Quantity for Assessing the Accuracy of Crystal
Structures,
              Nature 355, 472-474 (1992) +}
{+ reference: N.S. Pannu and R.J. Read, Improved structure refinement
              through maximum likelihood, Acta Cryst. A52, 659-668
(1996) +}
{+ reference: P.D. Adams, N.S. Pannu, R.J. Read and A.T. Brunger,
              Cross-validated Maximum Likelihood Enhances
Crystallographic
              Simulated Annealing Refinement, Proc. Natl. Acad. Sci.
USA
              94, 5018-5023 (1997) +}

{- Guidelines for using this file:
   - all strings must be quoted by double-quotes
   - logical variables (true/false) are not quoted
   - do not remove any evaluate statements from the file
   - the selections store1 through store8 are available for general use
-}
```

```
{- begin block parameter definition -} define(

{====================== molecular structure ======================-}

{* molecular topology file *}
{===>} structure_infile="alternate.mtf";

{* parameter files *}
{===>} parameter_infile_1="CNS_TOPPAR:protein_rep.param";
{===>} parameter_infile_2="CNS_TOPPAR:water_rep.param";
{===>} parameter_infile_3="cd_xplor_par.txt";
{===>} parameter_infile_4="cis_peptide3.param";
{===>} parameter_infile_5="";

{* coordinate file *}
{===>} coordinate_infile="alternate.pdb";

{=================== crystallographic data ======================}

{* space group *}
{* use International Table conventions with subscripts substituted
   by parenthesis *}
{===>} sg="P4(1)22";

{* unit cell parameters in Angstroms and degrees *}
{+ table: rows=1 "cell" cols=6 "a" "b" "c" "alpha" "beta" "gamma" +}
{===>} a=73.577;
{===>} b=73.577;
{===>} c=92.298;
{===>} alpha=90;
{===>} beta=90;
{===>} gamma=90;

{* anomalous f' f'' library file *}
{* If a file is not specified, no anomalous contribution will be
included *}
{+ choice: "CNS_XRAYLIB:anom_cu.lib" "CNS_XRAYLIB:anom_mo.lib" ""
user_file +}
{===>} anom_library="";

{* reflection files *}
{* specify non-anomalous reflection files before anomalous reflection
files. *}
{* files must contain unique array names otherwise errors will occur *}
{===>} reflection_infile_1="jto2.hkl";
{===>} reflection_infile_2="";
{===>} reflection_infile_3="";

{* reciprocal space array containing observed amplitudes: required *}
{===>} obs_f="fobs";

{* reciprocal space array containing sigma values for amplitudes:
required *}
{===>} obs_sigf="sigma";
```

```
{* reciprocal space array containing test set for cross-validation:
required *}
{* cross-validation should always be used, with the possible exception
   of a final round of refinement including all data *}
{* cross-validation is always required for the maximum likelihood
targets *}
{===>} test_set="test";

{* number for selection of test reflections: required for cross-
validation *}
{* ie. reflections with the test set array equal to this number will be
       used for cross-validation, all other reflections form the
working set *}
{===>} test_flag=1;

{* reciprocal space array containing weighting scheme for observed
   amplitudes: optional *}
{* only used for the "residual" and "vector" targets - this will
   default to a constant value of 1 if array is not present *}
{===>} obs_w="";

{* reciprocal space array containing observed intensities: optional *}
{* required for the "mli" target *}
{===>} obs_i="";

{* reciprocal space array containing sigma values for intensities:
optional *}
{* required for the "mli" target *}
{===>} obs_sigi="";

{* reciprocal space arrays with experimental phase probability
   distribution: optional *}
{* Hendrickson-Lattman coefficients A,B,C,D *}
{* required for the "mlhl" target *}
{+ table: rows=1 "HL coefficients" cols=4 "A" "B" "C" "D" +}
{===>} obs_pa="";
{===>} obs_pb="";
{===>} obs_pc="";
{===>} obs_pd="";

{* complex reciprocal space array containing experimental phases:
optional *}
{* required for the "mixed" and "vector" targets *}
{===>} obs_phase="";

{* reciprocal space array containing experimental figures of merit:
optional *}
{* required for the "mixed" target *}
{===>} obs_fom="";

{* resolution limits to be used in refinement *}
{* the full resolution range of observed data should be used in
refinement.
   A bulk solvent correction should be applied to allow the use of low
```

```
   resolution terms. If no bulk solvent correction is applied, data
must
   be truncated at a lower resolution limit of between 8 and 6
Angstrom. *}
{+ table: rows=1 "resolution" cols=2 "lowest" "highest" +}
{===>} low_res=20;
{===>} high_res=1.8;

{* apply rejection criteria to amplitudes or intensities *}
{+ choice: "amplitude" "intensity" +}
{===>} obs_type="amplitude";

{* Observed data cutoff criteria: applied to amplitudes or intensities
*}
{* reflections with magnitude(Obs)/sigma < cutoff are rejected. *}
{===>} sigma_cut=0.0;

{* rms outlier cutoff: applied to amplitudes or intensities *}
{* reflections with magnitude(Obs) > cutoff*rms(Obs) will be rejected
*}
{===>} obs_rms=10000;

{------------------ non-crystallographic symmetry ================}

{* NCS-restraints/constraints file *}
{* see auxiliary/ncs.def *}
{===>} ncs_infile="";

{=========== initial B-factor and bulk solvent corrections =========}

{* initial B-factor correction *}
{+ choice: "no" "isotropic" "anisotropic" "anisotropic_fixed_isotropic"
+}
{===>} bscale="anisotropic";

{* lower resolution limit for B-factor correction *}
{* the correction can only be calculated using data truncated at a
   lower resolution limit of between 8 and 6 A. The correction will be
   applied to all reflections *}
{===>} low_res_bscale=6.0;

{* bulk solvent correction *}
{* a mask is required around the molecule(s). The region
   outside this mask is the solvent region *}
{+ choice: true false +}
{===>} bulk_sol=true;
bulk_sol=true;
bulk_mask="";
sol_k=-9999.;
sol_b=-9999.;

{* bulk solvent mask file *}
{* mask will be read from O type mask file if a name is given
   otherwise calculated from coordinates of selected atoms *}
{===>} bulk_mask_infile="";
```

```
{* solvent density level *}
{* if negative, determined automatically
   if positive, fixed at value given *}
{===>} sol_k=-1;

{* solvent B-factor *}
{* if negative, determined automatically
   if positive, fixed at value given *}
{===>} sol_b=-1;

{========================== atom selection ============================}

{* select atoms to be included in refinement *}
{* this should include all conformations if multiple conformations are
used *}
{===>} atom_select=(known and not hydrogen);

{* select fixed atoms *}
{* note: atoms at special positions are automatically fixed. So,
   you don't have to explicitly fix them here. *}
{===>} atom_fixed=(none);

{* select atoms to be harmonically restrained during refinement *}
{===>} atom_harm=(none);

{* harmonic restraint constant - for harmonically restrained atoms *}
{===>} k_harmonic=10;

{* select atoms in alternate conformation 1 *}
{===>} conf_1=(none);

{* select atoms in alternate conformation 2 *}
{===>} conf_2=(none);

{* select atoms in alternate conformation 3 *}
{===>} conf_3=(none);

{* select atoms in alternate conformation 4 *}
{===>} conf_4=(none);

{* additional restraints file *}
{* eg. auxiliary/dna-rna_restraints.def *}
{===>} restraints_infile="";

{==================== minimization parameters ======================}

{* number of minimization steps *}
{===>} minimize_nstep=200;

{* number of cycles *}
{===>} num_cycles=1;

{* refinement target *}
{+ list: mlf: maximum likelihood target using amplitudes
```

```
         mli: maximum likelihood target using intensities
        mlhl: maximum likelihood target using amplitudes
              and phase probability distribution
    residual: standard crystallographic residual
      vector: vector residual
       mixed: (1-fom)*residual + fom*vector
        e2e2: correlation coefficient using normalized E^2
        e1e1: correlation coefficient using normalized E
        f2f2: correlation coefficient using F^2
        f1f1: correlation coefficient using F +}
{+ choice: "mlf" "mli" "mlhl" "residual" "vector" "mixed"
           "e2e2" "e1e1" "f2f2" "f1f1" +}
{===>} reftarget="mlf";

{* Wa weight for X-ray term *}
{* this will be determined automatically if a negative value is given.
   Note: wa can be very different depending on the target - if it is
not
         determined automatically make sure an appropriate value is
used *}
{===>} wa=-1;

{* number of bins for refinement target *}
{* this will be determined automatically if a negative value is given
   otherwise the specified number of bins will be used *}
{===>} target_bins=-1;

{* memory allocation for FFT calculation *}
{* this will be determined automatically if a negative value is given
   otherwise the specified number of words will be allocated *}
{===>} fft_memory=-1;

{========================== output files ==========================}

{* output coordinate file *}
{===>} coordinate_outfile="minimize5.pdb";

{* format output coordinates for use in o *}
{* if false then the default CNS output coordinate format will be used
*}
{+ choice: true false +}
{===>} pdb_o_format=true;

{=========================================================================
=====}
{        things below this line do not normally need to be changed
}
{=========================================================================
=====}

 ) {- end block parameter definition -}

 checkversion 1.1

 evaluate ($log_level=quiet)
```

```
structure @&structure_infile end

coordinates @&coordinate_infile

parameter
  if ( &BLANK%parameter_infile_1 = false ) then
     @@&parameter_infile_1
  end if
  if ( &BLANK%parameter_infile_2 = false ) then
     @@&parameter_infile_2
  end if
  if ( &BLANK%parameter_infile_3 = false ) then
     @@&parameter_infile_3
  end if
  if ( &BLANK%parameter_infile_4 = false ) then
     @@&parameter_infile_4
  end if
  if ( &BLANK%parameter_infile_5 = false ) then
     @@&parameter_infile_5
  end if
end

xray

  @CNS_XTALLIB:spacegroup.lib (sg=&sg;
                               sgparam=$sgparam;)

  a=&a b=&b c=&c  alpha=&alpha beta=&beta gamma=&gamma

  @CNS_XRAYLIB:scatter.lib

  if ( &BLANK%reflection_infile_1 = false ) then
    reflection @@&reflection_infile_1 end
  end if
  if ( &BLANK%reflection_infile_2 = false ) then
    reflection @@&reflection_infile_2 end
  end if
  if ( &BLANK%reflection_infile_3 = false ) then
    reflection @@&reflection_infile_3 end
  end if

end

if ( &BLANK%anom_library = false ) then
  @@&anom_library
else
  set echo=off end
  xray anomalous=? end
  if ( $result = true ) then
    display Warning: no anomalous library has been specified
    display          no anomalous contribution will used in refinement
  end if
  set echo=on end
end if
```

```
{- copy define parameters of optional arrays into symbols so
   we can redefine them -}

evaluate ($obs_i=&obs_i)
evaluate ($obs_sigi=&obs_sigi)
evaluate ($obs_w=&obs_w)
xray
  @@CNS_XTALMODULE:checkrefinput (
                                  reftarget=&reftarget;
                                  obs_f=&obs_f;
                                  obs_sigf=&obs_sigf;
                                  test_set=&test_set;
                                  obs_pa=&obs_pa;
                                  obs_pb=&obs_pb;
                                  obs_pc=&obs_pc;
                                  obs_pd=&obs_pd;
                                  obs_phase=&obs_phase;
                                  obs_fom=&obs_fom;
                                  obs_w=$obs_w;
                                  obs_i=$obs_i;
                                  obs_sigi=$obs_sigi;
                                  )

  query name=fcalc domain=reciprocal end
  if ( $object_exist = false ) then
     declare name=fcalc domain=reciprocal type=complex end
  end if
  declare name=fbulk domain=reciprocal type=complex end
  do (fbulk=0) ( all )

  binresolution &low_res &high_res
  mapresolution &high_res

  if ( &obs_type = "intensity" ) then
    if ( &BLANK%obs_i = true ) then
      display  Error: observed intensity array is undefined
      display         aborting script
      abort
    end if
    evaluate ($reject_obs=&obs_i)
    evaluate ($reject_sig=&obs_sigi)
    show min (amplitude(&STRIP%obs_i)) (all)
    evaluate ($obs_lower_limit=$result-0.1)
  else
    evaluate ($reject_obs=&obs_f)
    evaluate ($reject_sig=&obs_sigf)
    evaluate ($obs_lower_limit=0)
  end if

  declare name=ref_active domain=reciprocal type=integer end
  declare name=tst_active domain=reciprocal type=integer end

  do (ref_active=0) ( all )
```

```
   do (ref_active=1) ( ( amplitude($STRIP%reject_obs) >
$obs_lower_limit ) and
                       ( &low_res >= d >= &high_res ) )

   statistics overall
     completeness
     selection=( ref_active=1 )
   end
   evaluate ($total_compl=$expression1)

   show sum(1) ( ref_active=1 )
   evaluate ($total_read=$select)
   evaluate ($total_theor=int(1./$total_compl * $total_read))

   show rms (amplitude($STRIP%reject_obs)) ( ref_active=1 )
   evaluate ($obs_high=$result*&obs_rms)
   show min (amplitude($STRIP%reject_obs)) ( ref_active=1 )
   evaluate ($obs_low=$result)

   do (ref_active=0) ( all )
   do (ref_active=1)
                  ( ( amplitude($STRIP%reject_obs) >=
&sigma_cut*$STRIP%reject_sig ) and
                    ( $STRIP%reject_sig # 0 ) and
                    ( $obs_low <= amplitude($STRIP%reject_obs) <=
$obs_high ) and
                    ( &low_res >= d >= &high_res ) )

   do (tst_active=0) (all)
   if ( &BLANK%test_set = false ) then
     do (tst_active=1) (ref_active=1 and &STRIP%test_set=&test_flag)
   end if

   show sum(1) ( ref_active=1 and tst_active=0 )
   evaluate ($total_work=$select)
   show sum(1) ( ref_active=1 and tst_active=1 )
   evaluate ($total_test=$select)
   evaluate ($total_used=$total_work+$total_test)

   evaluate ($unobserved=$total_theor-$total_read)
   evaluate ($rejected=$total_read-$total_used)
   evaluate ($per_unobs=100*($unobserved/$total_theor))
   evaluate ($per_reject=100*($rejected/$total_theor))
   evaluate ($per_used=100*($total_used/$total_theor))
   evaluate ($per_work=100*($total_work/$total_theor))
   evaluate ($per_test=100*($total_test/$total_theor))

   associate fcalc ( &atom_select )

   tselection=( ref_active=1 )

   cvselection=( tst_active=1 )

   method=FFT
```

```
  fft
    if ( &fft_memory < 0 ) then
      automemory=true
    else
      memory=&fft_memory
    end if
  end

  tolerance=0.0 lookup=false

  if ( &wa >= 0 ) then
     wa=&wa
  end if

end

if ( &BLANK%ncs_infile = false ) then
   inline @&ncs_infile
end if

if ( &BLANK%restraints_infile = false ) then
    @&restraints_infile
end if

do (store9=0) (all)

evaluate ($nalt=1)
evaluate ($alt=1)
evaluate ($done=false)
while ( $done = false ) loop nalt
  if ( &exist_conf_$alt = true ) then
    show sum(1) ( &conf_$alt )
    if ( $result > 0 ) then
      evaluate ($nalt=$nalt+1)
    end if
  else
    evaluate ($done=true)
    evaluate ($nalt=$nalt-1)
  end if
  evaluate ($alt=$alt+1)
end loop nalt

evaluate ($alt=1)
while ( $alt <= $nalt ) loop alt
  do (store9=$alt) ( &conf_$alt )
  evaluate ($alt=$alt+1)
end loop alt

igroup
  interaction ( &atom_select and not(attr store9 > 0))
              ( &atom_select and not(attr store9 > 0))
  evaluate ($alt=1)
  while ( $alt <= $nalt ) loop alcs
    interaction ( &atom_select and ( attr store9 = $alt or attr store9
= 0 ))
```

```
                ( &atom_select and ( attr store9 = $alt ))
    evaluate ($alt=$alt+1)
  end loop alcs
end

{- check isolated atoms and atoms at special positions and add to
   list of fixed atoms if needed - store9 will be used -}

@CNS_XTALMODULE:setupfixed (
                          mode="minimization";
                          atom_select=&atom_select;
                          atom_fixed=&atom_fixed;
                          atom_total_fixed=store9;
                          atom_multiplicity=rmsd;
                          )

fix selection=( store9 ) end

fastnb grid end

flags
   exclude elec include pvdw xref
   ?
end

show sum(1) (&atom_harm)
if ( $result > 0 ) then
  evaluate ($harmonic=true)
else
  evaluate ($harmonic=false)
end if

xray
  predict
    mode=reciprocal
    to=fcalc
    selection=(ref_active=1)
    atomselection=( &atom_select )
  end
end

@@CNS_XTALMODULE:scalenbulk (bscale=&bscale;
                           sel=( ref_active=1 );
                           sel_test=( tst_active=1 );
                           res_bscale=&low_res_bscale;
                           atom_select=( &atom_select );
                           bulk_sol=&bulk_sol;
                           bulk_mask=&bulk_mask_infile;
                           bulk_atoms=( &atom_select );
                           sol_k=&sol_k;
                           sol_b=&sol_b;
                           fcalc=fcalc;
                           fobs=&STRIP%obs_f;
                           sigma=&STRIP%obs_sigf;
                           iobs=$STRIP%obs_i;
```

```
                                sigi=$STRIP%obs_sigi;
                                fpart=fbulk;
                                B2_11=$aniso_11;
                                B2_22=$aniso_22;
                                B2_33=$aniso_33;
                                B2_12=$aniso_12;
                                B2_13=$aniso_13;
                                B2_23=$aniso_23;
                                trace=$trace;
                                sol_k_ref=$sol_k_ref;
                                sol_b_ref=$sol_b_ref;
                                b_too_low=$low_b_flag;)

if ( $harmonic = true ) then
  do (refx=x) (all)
  do (refy=y) (all)
  do (refz=z) (all)
  do (harm=0) (all)
  do (harm=&k_harmonic) (&atom_harm)
  flags include harm end
end if

xray
   @@CNS_XTALMODULE:calculate_r (fobs=&STRIP%obs_f;
                                 fcalc=fcalc;
                                 fpart=fbulk;
                                 sel=(ref_active=1);
                                 sel_test=(tst_active=1);
                                 print=true;
                                 output=OUTPUT;
                                 r=$start_r;
                                 test_r=$start_test_r;)
end

evaluate ($cycle=1)

while ($cycle <= &num_cycles) loop main

  xray
    @@CNS_XTALMODULE:refinementtarget (target=&reftarget;
                                       sig_sigacv=0.07;
                                       mbins=&target_bins;
                                       fobs=&STRIP%obs_f;
                                       sigma=&STRIP%obs_sigf;
                                       weight=$STRIP%obs_w;
                                       iobs=$STRIP%obs_i;
                                       sigi=$STRIP%obs_sigi;
                                       test=tst_active;
                                       fcalc=fcalc;
                                       fpart=fbulk;
                                       pa=&STRIP%obs_pa;
                                       pb=&STRIP%obs_pb;
                                       pc=&STRIP%obs_pc;
                                       pd=&STRIP%obs_pd;
                                       phase=&STRIP%obs_phase;
```

```
                                        fom=&STRIP%obs_fom;
                                        sel=(ref_active=1);
                                        sel_test=(tst_active=1);
                                        statistics=true;)
  end

  if ( &wa < 0 ) then
    @@CNS_XTALMODULE:getweight (
                                selected=&atom_select;
                                fixed=(store9);
                               )
  end if

  if ( &minimize_nstep > 0 ) then
    minimize lbfgs
      nstep=&minimize_nstep
      nprint=5
      drop=10.0
    end
  end if

  evaluate ($cycle=$cycle+1)

end loop main

xray
   predict
     mode=reciprocal
     to=fcalc
     selection=(ref_active=1)
     atomselection=( &atom_select )
   end
   @@CNS_XTALMODULE:calculate_r (fobs=&STRIP%obs_f;
                                 fcalc=fcalc;
                                 fpart=fbulk;
                                 sel=(ref_active=1);
                                 sel_test=(tst_active=1);
                                 print=true;
                                 output=OUTPUT;
                                 r=$full_r;
                                 test_r=$full_test_r;)
end

print threshold=20.0 bond
evaluate ($rmsd_bond=$result)

print threshold=50.0 angle
evaluate ($rmsd_angle=$result)

set display=&coordinate_outfile end

display REMARK coordinates from minimization refinement
display REMARK refinement resolution: &low_res - &high_res A
if ( $total_test > 0 ) then
```

```
   display REMARK starting r= $start_r[f6.4] free_r=
$start_test_r[f6.4]
   display REMARK final    r= $full_r[f6.4] free_r= $full_test_r[f6.4]
 else
   display REMARK starting r= $start_r[f6.4]
   display REMARK final    r= $full_r[f6.4]
 end if
 display REMARK rmsd bonds= $rmsd_bond[f8.6]  rmsd angles=
$rmsd_angle[f8.5]
 xray wa=? end
 evaluate ($wa_print=$result)
 display REMARK wa= $wa_print
 display REMARK target= &STRIP%reftarget cycles= &num_cycles steps=
&minimize_nstep
 display REMARK sg= &STRIP%sg a= &a b= &b c= &c alpha= &alpha beta=
&beta gamma= &gamma
 if ( &BLANK%parameter_infile_1 = false ) then
    display REMARK parameter file 1  : &STRIP%parameter_infile_1
 end if
 if ( &BLANK%parameter_infile_2 = false ) then
    display REMARK parameter file 2  : &STRIP%parameter_infile_2
 end if
 if ( &BLANK%parameter_infile_3 = false ) then
    display REMARK parameter file 3  : &STRIP%parameter_infile_3
 end if
 if ( &BLANK%parameter_infile_4 = false ) then
    display REMARK parameter file 4  : &STRIP%parameter_infile_4
 end if
 if ( &BLANK%parameter_infile_5 = false ) then
    display REMARK parameter file 5  : &STRIP%parameter_infile_5
 end if
 display REMARK molecular structure file: &STRIP%structure_infile
 display REMARK input coordinates: &STRIP%coordinate_infile
 if ( &BLANK%anom_library = false ) then
   display REMARK anomalous f' f'' library: &STRIP%anom_library
 end if
 if ( &BLANK%reflection_infile_1 = false ) then
   display REMARK reflection file= &STRIP%reflection_infile_1
 end if
 if ( &BLANK%reflection_infile_2 = false ) then
   display REMARK reflection file= &STRIP%reflection_infile_2
 end if
 if ( &BLANK%reflection_infile_3 = false ) then
   display REMARK reflection file= &STRIP%reflection_infile_3
 end if
 if ( &BLANK%restraints_infile = false ) then
    display REMARK additional restraints file: &STRIP%restraints_infile
 end if
 if ( &BLANK%ncs_infile = false ) then
   display REMARK ncs= &STRIP%ncs_type  ncs file= &STRIP%ncs_infile
 else
   display REMARK ncs= none
 end if
 if ( &bscale # "no" ) then
   display REMARK B-correction resolution: &low_res_bscale - &high_res
```

```
   if ( $low_b_flag = true ) then
     display REMARK warning: B-correction gave atomic B-values less
than zero
     display REMARK          they have been reset to zero
   end if
 end if
 if ( &bscale = "anisotropic" ) then
   display REMARK initial B-factor correction applied to &STRIP%obs_f :
   display REMARK   B11=$aniso_11[f8.3] B22=$aniso_22[f8.3]
B33=$aniso_33[f8.3]
   display REMARK   B12=$aniso_12[f8.3] B13=$aniso_13[f8.3]
B23=$aniso_23[f8.3]
   display REMARK B-factor correction applied to coordinate array B:
$trace[f8.3]
 elseif ( &bscale = "anisotropic_fixed_isotropic" ) then
   display REMARK initial B-factor correction applied to &STRIP%obs_f :
   display REMARK   B11=$aniso_11[f8.3] B22=$aniso_22[f8.3]
B33=$aniso_33[f8.3]
   display REMARK   B12=$aniso_12[f8.3] B13=$aniso_13[f8.3]
B23=$aniso_23[f8.3]
   display REMARK no B-factor correction applied to coordinate array B
 elseif ( &bscale = "isotropic" ) then
   display REMARK B-factor correction applied to coordinate array B:
$trace[f8.3]
 else
   display REMARK initial B-factor correction: none
 end if
 if ( &bulk_sol = true ) then
   display REMARK bulk solvent: density level= $sol_k_ref e/A^3, B-
factor= $sol_b_ref A^2
 else
   display REMARK bulk solvent: false
 end if
 if ( &obs_type = "intensity" ) then
   display REMARK reflections with Iobs/sigma_I < &sigma_cut rejected
   display REMARK reflections with Iobs > &obs_rms * rms(Iobs) rejected
 else
   display REMARK reflections with |Fobs|/sigma_F < &sigma_cut rejected
   display REMARK reflections with |Fobs| > &obs_rms * rms(Fobs)
rejected
 end if
 xray anomalous=? end
 if ( $result = true ) then
   display REMARK anomalous diffraction data was input
 end if
 display REMARK theoretical total number of refl. in resol. range:
$total_theor[I6] ( 100.0 % )
 display REMARK number of unobserved reflections (no entry or |F|=0):
$unobserved[I6] ( $per_unobs[f5.1] % )
 display REMARK number of reflections rejected:
$rejected[I6] ( $per_reject[f5.1] % )
 display REMARK total number of reflections used:
$total_used[I6] ( $per_used[f5.1] % )
 display REMARK number of reflections in working set:
$total_work[I6] ( $per_work[f5.1] % )
```

```
 display REMARK number of reflections in test set:
$total_test[I6] ( $per_test[f5.1] % )

 remark

 @CNS_XTALMODULE:write_pdb (pdb_o_format=&pdb_o_format;
                            coordinate_outfile=&coordinate_outfile;
                            sgparam=$sgparam;)

 stop

CNS 2Fo-Fc map

{+ file: map.inp +}
{+ directory: xtal_refine +}
{+ description: Make an electron density map using phase information
                from a model +}
{+ comment:
            choice of coefficients:
              (u m|Fo| e^(i phi_calc)) - (v D|Fc| e^(i phi_calc))
              (u  |Fo| e^(i phi_calc)) - (v k|Fc| e^(i phi_calc))
              (u m|Fo| e^(i phi_comb)) - (v D|Fc| e^(i phi_calc))
              (u m_obs|Fo| e^(i phi_obs )) - (v k m_obs|Fc| e^(i
phi_calc))
               d(target)/dFc
              where Fc is the calculated structure factor and
              m and D are derived from sigmaa
            Averaging of structure factors from several models
optional.
            Realspace R-value calculation optional. +}
{+ authors: Axel T. Brunger and Paul D. Adams +}
{+ copyright: Yale University +}

{+ reference: R.J. Read, Improved Fourier coefficients for maps using
              phases from partial structures with errors. Acta Cryst.
              A42, 140-149 (1986) +}
{+ reference: G.J. Kleywegt and A.T. Brunger, Checking your
imagination:
              Applications of the free R value, Structure 4,
              897-904 (1996) +}
{+ reference: A.T. Brunger, P.D. Adams and L.M. Rice, New applications
              of simulated annealing in X-ray crystallography and
              solution NMR, Structure 5, 325-336, (1997) +}

{- Guidelines for using this file:
   - all strings must be quoted by double-quotes
   - logical variables (true/false) must not be quoted
   - do not remove any evaluate statements from the file -}

{- begin block parameter definition -} define(

{====================== molecular structure ======================-}

{* molecular topology file *}
{===>} structure_infile="gen8.psf";
```

```
{* parameter files *}
{===>} parameter_infile_1="CNS_TOPPAR:protein_rep.param";
{===>} parameter_infile_2="CNS_TOPPAR:water_rep.param";
{===>} parameter_infile_3="cd_xplor_par.txt";
{===>} parameter_infile_4="cis_peptide.param";
{===>} parameter_infile_5="";

{* coordinate files *}
{* structure factors from multiple coordinates sets will be averaged *}
{===>} coordinate_infile_1="bref.pdb";
{===>} coordinate_infile_2="";
{===>} coordinate_infile_3="";
{===>} coordinate_infile_4="";
{===>} coordinate_infile_5="";
{===>} coordinate_infile_6="";
{===>} coordinate_infile_7="";
{===>} coordinate_infile_8="";
{===>} coordinate_infile_9="";
{===>} coordinate_infile_10="";

{====================== crystallographic data ========================}

{* space group *}
{* use International Table conventions with subscripts substituted
   by parenthesis *}
{===>} sg="P4(1)";

{* unit cell parameters in Angstroms and degrees *}
{+ table: rows=1 "cell" cols=6 "a" "b" "c" "alpha" "beta" "gamma" +}
{===>} a=73.577;
{===>} b=73.577;
{===>} c=92.298;
{===>} alpha=90;
{===>} beta=90;
{===>} gamma=90;

{* anomalous f' f'' library file *}
{* If a file is not specified, no anomalous contribution will be
included *}
{+ choice: "CNS_XRAYLIB:anom_cu.lib" "CNS_XRAYLIB:anom_mo.lib" ""
user_file +}
{===>} anom_library="";

{* reflection files *}
{* specify non-anomalous reflection files before anomalous reflection
files. *}
{* files must contain unique array names otherwise errors will occur *}
{===>} reflection_infile_1="jto2.hkl";
{===>} reflection_infile_2="";
{===>} reflection_infile_3="";

{* reciprocal space array containing observed amplitudes: required *}
{===>} obs_f="fobs";
```

```
{* reciprocal space array containing sigma values for amplitudes:
required *}
{===>} obs_sigf="sigma";

{* reciprocal space array containing test set for cross-validation:
   required for calculation of cross-validated sigmaA values *}
{===>} test_set="test";

{* number for selection of test reflections: required for cross-
validation *}
{* ie. reflections with the test set array equal to this number will be
       used for cross-validation, all other reflections form the
working set *}
{===>} test_flag=1;

{* reciprocal space array containing weighting scheme for observed
   amplitudes: optional *}
{* only used for the "residual" and "vector" targets - this will
   default to a constant value of 1 if array is not present *}
{===>} obs_w="";

{* reciprocal space array containing observed intensities: optional *}
{* required for the "mli" target *}
{===>} obs_i="";

{* reciprocal space array containing sigma values for intensities:
optional *}
{* required for the "mli" target *}
{===>} obs_sigi="";

{* reciprocal space arrays with experimental phase probability
   distribution: optional *}
{* Hendrickson-Lattman coefficients A,B,C,D *}
{* required for the "mlhl" target and phase combined or observed maps
*}
{+ table: rows=1 "HL coefficients" cols=4 "A" "B" "C" "D" +}
{===>} obs_pa="";
{===>} obs_pb="";
{===>} obs_pc="";
{===>} obs_pd="";

{* complex reciprocal space array containing experimental phases:
optional *}
{* required for the "mixed" and "vector" targets *}
{===>} obs_phase="fobs";

{* reciprocal space array containing experimental figures of merit:
optional *}
{* required for the "mixed" target *}
{===>} obs_fom="fom";

{* resolution limits for data included in map calculation *}
{* all data available should be included in the map calculation *}
{+ table: rows=1 "resolution" cols=2 "lowest" "highest" +}
{===>} low_res=100;
```

```
{===>} high_res=1.8;

{* apply rejection criteria to amplitudes or intensities *}
{+ choice: "amplitude" "intensity" +}
{===>} obs_type="amplitude";

{* Observed data cutoff criteria: applied to amplitudes or intensities
*}
{* reflections with magnitude(Obs)/sigma < cutoff are rejected. *}
{===>} sigma_cut=0.0;

{* rms outlier cutoff: applied to amplitudes or intensities *}
{* reflections with magnitude(Obs) > cutoff*rms(Obs) will be rejected
*}
{===>} obs_rms=1000;

{================== non-crystallographic symmetry ==================}

{* NCS-restraints/constraints file *}
{* see auxiliary/ncs.def *}
{===>} ncs_infile="";

{============ initial B-factor and bulk solvent corrections ==========}

{* initial B-factor correction *}
{+ choice: "no" "isotropic" "anisotropic" "anisotropic_fixed_isotropic"
+}
{===>} bscale="anisotropic";

{* lower resolution limit for B-factor correction *}
{* the correction can only be calculated using data truncated at a
   lower resolution limit of between 8 and 6 A. The correction will be
   applied to all reflections *}
{===>} low_res_bscale=6.0;

{* bulk solvent correction *}
{* a mask is required around the molecule(s). The region
   outside this mask is the solvent region *}
{+ choice: true false +}
{===>} bulk_sol=true;

{* bulk solvent mask file *}
{* mask will be read from O type mask file if a name is given
   otherwise calculated from coordinates of selected atoms *}
{===>} bulk_mask_infile="";

{* solvent density level *}
{* if negative, determined automatically
   if positive, fixed at value given *}
{===>} sol_k=-1;

{* solvent B-factor *}
{* if negative, determined automatically
   if positive, fixed at value given *}
{===>} sol_b=-1;
```

```
{========================== atom selection ===========================}

{* select atoms to be included in map calculation *}
{===>} atom_select=(known and not hydrogen);

{==================== map generation parameters ======================}

{* maps are calculated u*Fo - v*Fc *}
{* eg. 2fo-fc map -> u=2 and v=1 or
       fo-fc map -> u=1 and v=1 *}

{* specify u *}
{===>} u=2;

{* specify v *}
{===>} v=1;

{* type of map *}
{+ list:   sigmaa: (u m|Fo| - v D|Fc|)^exp(i phi_calc)
                   m and D calculated from sigmaa
       unweighted: (u |Fo| - v k|Fc|)^exp(i phi_calc)
                   no figure-of-merit weighting
         combined: (u m|Fo|^exp(i phi_comb) - v D|Fc|^exp(i phi_calc))
                   model and experimental phases combined, m and D
from sigmaa
         observed: (u m|Fo|^exp(i phi_obs) - v k m|Fc|^exp(i phi_calc))
                   observed phases and fom from phase probability
distribution
         gradient: d(target)/dFc
                   gradient of the current crystallographic target wrt
Fc
       NB. experimental phases must be supplied as a phase
           probability distribution in the Hendrickson-Lattman arrays
+}
{+ choice: "sigmaa" "unweighted" "combined" "observed" "gradient" +}
{===>} map_type="sigmaa";

{* refinement target - only used for gradient maps *}
{+ list: mlf: maximum likelihood target using amplitudes
         mli: maximum likelihood target using intensities
        mlhl: maximum likelihood target using amplitudes
              and phase probability distribution
    residual: standard crystallographic residual
      vector: vector residual
       mixed: (1-fom)*residual + fom*vector
        e2e2: correlation coefficient using normalized E^2
        e1e1: correlation coefficient using normalized E
        f2f2: correlation coefficient using F^2
        f1f1: correlation coefficient using F +}
{+ choice: "mlf" "mli" "mlhl" "residual" "vector" "mixed"
           "e2e2" "e1e1" "f2f2" "f1f1" +}
{===>} reftarget="mlf";

{* number of bins for refinement target *}
```

```
{* this will be determined automatically if a negative value is given
   otherwise the specified number of bins will be used *}
{===>} target_bins=-1;

{* use model amplitudes for unmeasured data *}
{* this will not be applied to gradient or difference maps *}
{+ choice: true false +}
{===>} fill_in=false;

{* scale map by dividing by the rms sigma of the map *}
{* otherwise map will be on an absolute fobs scale *}
{+ choice: true false +}
{===>} map_scale=true;

{* map format *}
{* choice: "cns" "ezd" *}
{===>} map_format="cns";

{* map grid size: dmin*grid *}
{* use grid=0.25 for better map appearance *}
{===>} grid=0.33;

{* memory allocation for FFT calculation *}
{* this will be determined automatically if a negative value is given
   otherwise the specified number of words will be allocated *}
{===>} fft_memory=-1;

{* extent of map *}
{+ choice: "molecule" "asymmetric" "unit" "box" "fract" +}
{===>} map_mode="molecule";

{* select atoms around which map will be written *}
{* change if different to atoms selected for map calculation *}
{===>} atom_map=(known and not hydrogen);

{* cushion (in Angstroms) around selected atoms in "molecule" mode *}
{===>} map_cushion=3.0;

{* limits in orthogonal angstroms for box mode or
   fractional coordinates for fract mode *}
{+ table: rows=3 "x" "y" "z" cols=2 "minimum" "maximum" +}
{===>} xmin=0.;
{===>} xmax=0.;
{===>} ymin=0.;
{===>} ymax=0.;
{===>} zmin=0.;
{===>} zmax=0.;

{========================= realspace R-value ========================}

{* calculate realspace R-values *}
{+ choice: true false +}
{===>} real_r=false;

{* select residues for which realspace R-values will be calculated *}
```

```
{===>} atom_real=(known and not hydrogen);

{---------------------------- output files ============================}

{* root name for output files *}
{+ list:
         map file will be in: <output_root>.map
         positive peaks in: <output_root>_positive.peaks
         negative peaks in: <output_root>_negative.peaks
         realspace R-values in: <output_root>_r.list
         Fourier coefficients will be in: <output_root>.coeff +}
{===>} output_root="bref8_2fofc";

{* write map file *}
{+ choice: true false +}
{===>} write_map=true;

{* do peak picking on map *}
{* optional - use water_pick.inp to pick waters *}
{+ choice: true false +}
{===>} peak_search=true;

{* number of peaks to pick from map *}
{===>} peak_num=30;

{* write a reflection file with the Fourier coefficients of the map *}
{+ list: arrays written:
         map_fom:   FOM weight applied to observed data
         map_phase: phases used for observed data
         map_scale: scale factor applied to calculated data
         map_coeff: Fourier map coefficients - map=ft(map_coeff) +}
{+ choice: true false +}
{===>} write_coeff=false;

{=========================================================================
=====}
{         things below this line do not normally need to be changed
}
{=========================================================================
=====}

 ) {- end block parameter definition -}

{ checkversion 0.5}

 evaluate ($log_level=quiet)

 structure @&structure_infile end

 coordinates @&coordinate_infile_1

 parameter
   if ( &BLANK%parameter_infile_1 = false ) then
      @@&parameter_infile_1
   end if
```

```
  if ( &BLANK%parameter_infile_2 = false ) then
     @@&parameter_infile_2
  end if
  if ( &BLANK%parameter_infile_3 = false ) then
     @@&parameter_infile_3
  end if
  if ( &BLANK%parameter_infile_4 = false ) then
     @@&parameter_infile_4
  end if
  if ( &BLANK%parameter_infile_5 = false ) then
     @@&parameter_infile_5
  end if
end

xray

  @CNS_XTALLIB:spacegroup.lib (sg=&sg;)

  a=&a b=&b c=&c  alpha=&alpha beta=&beta gamma=&gamma

  @CNS_XRAYLIB:scatter.lib

  binresolution &low_res &high_res
  mapresolution &high_res

  generate &low_res &high_res

  if ( &BLANK%reflection_infile_1 = false ) then
    reflection @@&reflection_infile_1 end
  end if
  if ( &BLANK%reflection_infile_2 = false ) then
    reflection @@&reflection_infile_2 end
  end if
  if ( &BLANK%reflection_infile_3 = false ) then
    reflection @@&reflection_infile_3 end
  end if

end

if ( &BLANK%anom_library = false ) then
  @@&anom_library
else
  set echo=off end
  xray anomalous=? end
  if ( $result = true ) then
    display Warning: no anomalous library has been specified
    display          no anomalous contribution will used in refinement
  end if
  set echo=on end
end if

{- copy define parameters of optional arrays into symbols so
   we can redefine them -}

evaluate ($obs_i=&obs_i)
```

```
evaluate ($obs_sigi=&obs_sigi)
evaluate ($obs_w=&obs_w)
xray
  @@CNS_XTALMODULE:checkrefinput (
                                  reftarget=&reftarget;
                                  obs_f=&obs_f;
                                  obs_sigf=&obs_sigf;
                                  test_set=&test_set;
                                  obs_pa=&obs_pa;
                                  obs_pb=&obs_pb;
                                  obs_pc=&obs_pc;
                                  obs_pd=&obs_pd;
                                  obs_phase=&obs_phase;
                                  obs_fom=&obs_fom;
                                  obs_w=$obs_w;
                                  obs_i=$obs_i;
                                  obs_sigi=$obs_sigi;
                                  )

  query name=fcalc domain=reciprocal end
  if ( $object_exist = false ) then
     declare name=fcalc domain=reciprocal type=complex end
  end if
  declare name=fbulk domain=reciprocal type=complex end
  do (fbulk=0) ( all )

  query name=&STRIP%obs_f domain=reciprocal end
  declare name=fobs_orig domain=reciprocal type=$object_type end
  declare name=sigma_orig domain=reciprocal type=real end

  do (fobs_orig=&STRIP%obs_f) (all)
  do (sigma_orig=&STRIP%obs_sigf) (all)

  if ( &BLANK%obs_i = false ) then
    query name=&STRIP%obs_i domain=reciprocal end
    declare name=iobs_orig domain=reciprocal type=$object_type end
    declare name=sigi_orig domain=reciprocal type=real end
    do (iobs_orig=&STRIP%obs_i) (all)
    do (sigi_orig=&STRIP%obs_sigi) (all)
  end if

  if ( &obs_type = "intensity" ) then
    if ( &BLANK%obs_i = true ) then
      display  Error: observed intensity array is undefined
      display         aborting script
      abort
    end if
    evaluate ($reject_obs=&obs_i)
    evaluate ($reject_sig=&obs_sigi)
    show min (amplitude(&STRIP%obs_i)) (all)
    evaluate ($obs_lower_limit=$result-0.1)
  else
    evaluate ($reject_obs=&obs_f)
    evaluate ($reject_sig=&obs_sigf)
    evaluate ($obs_lower_limit=0)
```

```
   end if

   declare name=ref_active domain=reciprocal type=integer end
   declare name=tst_active domain=reciprocal type=integer end

   do (ref_active=0) ( all )
   do (ref_active=1) ( ( amplitude($STRIP%reject_obs) >
$obs_lower_limit ) and
                       ( &low_res >= d >= &high_res ) )

   statistics overall
     completeness
     selection=( ref_active=1 )
   end
   evaluate ($total_compl=$expression1)

   show sum(1) ( ref_active=1 )
   evaluate ($total_read=$select)
   evaluate ($total_theor=int(1./$total_compl * $total_read))

   show rms (amplitude($STRIP%reject_obs)) ( ref_active=1 )
   evaluate ($obs_high=$result*&obs_rms)
   show min (amplitude($STRIP%reject_obs)) ( ref_active=1 )
   evaluate ($obs_low=$result)

   do (ref_active=0) ( all )
   do (ref_active=1)
                  ( ( amplitude($STRIP%reject_obs) >=
&sigma_cut*$STRIP%reject_sig ) and
                    ( $STRIP%reject_sig # 0 ) and
                    ( $obs_low <= amplitude($STRIP%reject_obs) <=
$obs_high ) and
                    ( &low_res >= d >= &high_res ) )

   do (tst_active=0) (all)
   if ( &BLANK%test_set = false ) then
     do (tst_active=1) (ref_active=1 and &STRIP%test_set=&test_flag)
   end if

   show sum(1) ( ref_active=1 and tst_active=0 )
   evaluate ($total_work=$select)
   show sum(1) ( ref_active=1 and tst_active=1 )
   evaluate ($total_test=$select)
   evaluate ($total_used=$total_work+$total_test)

   evaluate ($unobserved=$total_theor-$total_read)
   evaluate ($rejected=$total_read-$total_used)
   evaluate ($per_unobs=100*($unobserved/$total_theor))
   evaluate ($per_reject=100*($rejected/$total_theor))
   evaluate ($per_used=100*($total_used/$total_theor))
   evaluate ($per_work=100*($total_work/$total_theor))
   evaluate ($per_test=100*($total_test/$total_theor))

   associate fcalc ( &atom_select )
```

```
  tselection=( ref_active=1 )

  cvselection=( tst_active=1 )

  method=FFT

  fft
    grid=&grid
    if ( &fft_memory < 0 ) then
      automemory=true
    else
      memory=&fft_memory
    end if
  end

  tolerance=0.0 lookup=false

end

if ( &map_type = "observed" ) then
  evalaute ($test_hl=true)
elseif ( &map_type = "combined" ) then
  evalaute ($test_hl=true)
else
  evalaute ($test_hl=false)
end if

if ( $test_hl = true ) then
  xray
    @@CNS_XTALMODULE:check_abcd (pa=&obs_pa;
                                 pb=&obs_pb;
                                 pc=&obs_pc;
                                 pd=&obs_pd;)
  end
end if

if ( &BLANK%ncs_infile = false ) then
   inline @&ncs_infile
end if

evaluate ($struct=1)
evaluate ($done=false)
while ( $done = false ) loop struct
  if ( &exist_coordinate_infile_$struct = true ) then
    if ( &BLANK%coordinate_infile_$struct = true ) then
      evaluate ($done=true)
      evaluate ($struct=$struct-1)
    else
      evaluate ($struct=$struct+1)
    end if
  else
    evaluate ($done=true)
    evaluate ($struct=$struct-1)
  end if
end loop struct
```

```
evaluate ($ave_cycle=$struct)

evaluate ($cycle=1)

xray
  declare name=dtarg domain=reciprocal type=complex end
  declare name=fcave domain=reciprocal type=complex end
  declare name=fpave domain=reciprocal type=complex end
  query name=&STRIP%obs_f domain=reciprocal end
  declare name=foave domain=reciprocal type=$object_type end
  declare name=dave domain=reciprocal type=complex end
  declare name=total domain=reciprocal type=complex end
  declare name=fmap domain=reciprocal type=complex end
  do (fcave=0) (all)
  do (fpave=0) (all)
  do (foave=0) (all)
  do (dave=0) (all)
end

while ( $cycle <= $ave_cycle ) loop main

  coord init end
  coord @@&coordinate_infile_$cycle

  xray
    do (&STRIP%obs_f=fobs_orig) (all)
    do (&STRIP%obs_sigf=sigma_orig) (all)
    if ( &BLANK%obs_i = false ) then
      do (&STRIP%obs_i=iobs_orig) (all)
      do (&STRIP%obs_sigi=sigi_orig) (all)
    end if
  end

  xray
     predict
       mode=reciprocal
       to=fcalc
       selection=(all)
       atomselection=( &atom_select )
     end
  end

  @@CNS_XTALMODULE:scalenbulk (bscale=&bscale;
                               sel=( ref_active=1 );
                               sel_test=( tst_active=1 );
                               res_bscale=&low_res_bscale;
                               atom_select=( &atom_select );
                               bulk_sol=&bulk_sol;
                               bulk_mask=&bulk_mask_infile;
                               bulk_atoms=( &atom_select );
                               sol_k=&sol_k;
                               sol_b=&sol_b;
                               fcalc=fcalc;
                               fobs=&STRIP%obs_f;
```

```
                                    sigma=&STRIP%obs_sigf;
                                    iobs=$STRIP%obs_i;
                                    sigi=$STRIP%obs_sigi;
                                    fpart=fbulk;
                                    B2_11=$aniso_11;
                                    B2_22=$aniso_22;
                                    B2_33=$aniso_33;
                                    B2_12=$aniso_12;
                                    B2_13=$aniso_13;
                                    B2_23=$aniso_23;
                                    trace=$trace;
                                    sol_k_ref=$sol_k_ref;
                                    sol_b_ref=$sol_b_ref;
                                    b_too_low=$low_b_flag;)

  xray
    if ( &map_type = "gradient" ) then
      @@CNS_XTALMODULE:refinementtarget (target=&reftarget;
                                        sig_sigacv=0.07;
                                        mbins=&target_bins;
                                        fobs=&STRIP%obs_f;
                                        sigma=&STRIP%obs_sigf;
                                        weight=$STRIP%obs_w;
                                        iobs=$STRIP%obs_i;
                                        sigi=$STRIP%obs_sigi;
                                        test=tst_active;
                                        fcalc=fcalc;
                                        fpart=fbulk;
                                        pa=&STRIP%obs_pa;
                                        pb=&STRIP%obs_pb;
                                        pc=&STRIP%obs_pc;
                                        pd=&STRIP%obs_pd;
                                        phase=&STRIP%obs_phase;
                                        fom=&STRIP%obs_fom;
                                        sel=(ref_active=1);
                                        sel_test=(tst_active=1);
                                        statistics=true;)
      predict
         mode=dtarget(fcalc)
         to=dtarg
         selection=(ref_active=1)
         atomselection=( &atom_select )
      end
      do (dave=dave+dtarg) (all)
    end if
    do (fcave=fcave+fcalc) (all)
    do (fpave=fpave+fbulk) (all)
    do (foave=foave+&STRIP%obs_f) (all)
  end

  evaluate ($cycle=$cycle+1)

end loop main

xray
```

```
  if ( &map_type = "gradient" ) then
    do (dave=dave/$ave_cycle) (all)
  end if
  do (fcalc=fcave/$ave_cycle) (all)
  do (fbulk=fpave/$ave_cycle) (all)
  do (&STRIP%obs_f=foave/$ave_cycle) (all)
  @@CNS_XTALMODULE:calculate_r (fobs=&STRIP%obs_f;
                                fcalc=fcalc;
                                fpart=fbulk;
                                sel=(ref_active=1);
                                sel_test=(tst_active=1);
                                print=true;
                                output=OUTPUT;
                                r=$map_r;
                                test_r=$map_free_r;)
end

xray
  declare name=map_phase domain=reciprocal type=real end
  declare name=map_fom   domain=reciprocal type=real end
  declare name=map_scale domain=reciprocal type=real end
end

if ( &map_type = "unweighted" ) then

  xray

    do (map_phase=phase(fcalc+fbulk)) (all)

    do (total=fcalc+fbulk) (all)
    multiscale
      bfmin=-40 bfmax=40
      set1=&STRIP%obs_f  k1=-1 b1=0
      set2=total          b2=0
      selection=(ref_active=1 and d <= &low_res_bscale)
    end
    do (map_scale=$k2) (all)

    do (map_fom=1.0) (all)

  end

elseif ( &map_type = "sigmaa" ) then

  xray

    do (map_phase=phase(fcalc+fbulk)) (all)

    declare name=m          domain=reciprocal type=complex end
    declare name=mod_fom    domain=reciprocal type=real end
    declare name=mod_x      domain=reciprocal type=real end
    declare name=mod_pa     domain=reciprocal type=real end
    declare name=mod_pb     domain=reciprocal type=real end
    declare name=mod_pc     domain=reciprocal type=real end
    declare name=mod_pd     domain=reciprocal type=real end
```

```
    declare name=mod_dd     domain=reciprocal type=real end

    @CNS_XTALMODULE:fomsigmaacv ( sig_sigacv=0.07;
                                  mbins=&target_bins;
                                  statistics=true;
                                  fobs=&STRIP%obs_f;
                                  fcalc=fcalc;
                                  fpart=fbulk;
                                  test=tst_active;
                                  sel=(ref_active=1);
                                  sel_test=(tst_active=1);
                                  fom=mod_fom;
                                  x=mod_x;
                                  pa=mod_pa;
                                  pb=mod_pb;
                                  pc=mod_pc;
                                  pd=mod_pd;
                                  dd=mod_dd; )


    do (map_fom=mod_fom) (all)
    do (map_scale=distribute(mod_dd)) (&low_res >= d >= &high_res)

    undeclare name=m          domain=reciprocal end
    undeclare name=mod_fom    domain=reciprocal end
    undeclare name=mod_x      domain=reciprocal end
    undeclare name=mod_pa     domain=reciprocal end
    undeclare name=mod_pb     domain=reciprocal end
    undeclare name=mod_pc     domain=reciprocal end
    undeclare name=mod_pd     domain=reciprocal end
    undeclare name=mod_dd     domain=reciprocal end

  end

elseif ( &map_type = "combined" ) then

  xray
    declare name=m          domain=reciprocal type=complex end
    declare name=mod_fom    domain=reciprocal type=real end
    declare name=mod_x      domain=reciprocal type=real end
    declare name=mod_pa     domain=reciprocal type=real end
    declare name=mod_pb     domain=reciprocal type=real end
    declare name=mod_pc     domain=reciprocal type=real end
    declare name=mod_pd     domain=reciprocal type=real end
    declare name=mod_dd     domain=reciprocal type=real end

    @CNS_XTALMODULE:fomsigmaacv ( sig_sigacv=0.07;
                                  mbins=&target_bins;
                                  statistics=true;
                                  fobs=&STRIP%obs_f;
                                  fcalc=fcalc;
                                  fpart=fbulk;
                                  test=tst_active;
                                  sel=(ref_active=1);
                                  sel_test=(tst_active=1);
```

```
                                    fom=mod_fom;
                                    x=mod_x;
                                    pa=mod_pa;
                                    pb=mod_pb;
                                    pc=mod_pc;
                                    pd=mod_pd;
                                    dd=mod_dd; )

     @CNS_XTALMODULE:combineprobability ( messages="off";
                                          addname="model phases";
                                          pa=mod_pa;
                                          pb=mod_pb;
                                          pc=mod_pc;
                                          pd=mod_pd;
                                          w=1;
                                          addname="experimental
phases";
                                          adda=&STRIP%obs_pa;
                                          addb=&STRIP%obs_pb;
                                          addc=&STRIP%obs_pc;
                                          addd=&STRIP%obs_pd;
                                          addw=1;)

     @CNS_XTALMODULE:getfom ( pa=mod_pa;
                              pb=mod_pb;
                              pc=mod_pc;
                              pd=mod_pd;
                              m=m;
                              phistep=5; )

     do (map_phase=phase(m)) (all)
     do (map_fom=amplitude(m)) (all)
     do (map_scale=distribute(mod_dd)) (&low_res >= d >= &high_res)

     undeclare name=m          domain=reciprocal end
     undeclare name=mod_fom    domain=reciprocal end
     undeclare name=mod_x      domain=reciprocal end
     undeclare name=mod_pa     domain=reciprocal end
     undeclare name=mod_pb     domain=reciprocal end
     undeclare name=mod_pc     domain=reciprocal end
     undeclare name=mod_pd     domain=reciprocal end
     undeclare name=mod_dd     domain=reciprocal end
   end

 elseif ( &map_type = "observed" ) then

   xray

     do (total=fcalc+fbulk) (all)
     multiscale
       bfmin=-40 bfmax=40
       set1=&STRIP%obs_f  k1=-1 b1=0
       set2=total          b2=0
       selection=(ref_active=1 and d <= &low_res_bscale)
     end
```

```
    do (map_scale=$k2) (all)

    declare name=m          domain=reciprocal type=complex end

    @CNS_XTALMODULE:getfom ( pa=&STRIP%obs_pa;
                             pb=&STRIP%obs_pb;
                             pc=&STRIP%obs_pc;
                             pd=&STRIP%obs_pd;
                             m=m;
                             phistep=5; )

    do (map_phase=phase(m)) (all)
    do (map_fom=amplitude(m)) (all)
    do (map_scale=map_scale*map_fom) (all)
    undeclare name=m domain=reciprocal end
  end

 end if

 if ( &map_type = "gradient" ) then

   xray
     {- take the negative of the gradient so the map is the same sign
        as a standard difference map -}
     do (fmap=-dave) (ref_active=1)
     do (map_fom=0.0) (ref_active=1)
     do (map_scale=0.0) (ref_active=1)
     do (map_phase=0.0) (ref_active=1)
   end

 else

   xray
     if ( &u = &v ) then
       do (fmap= 2 ((&u   map_fom
combine(amplitude(&STRIP%obs_f),map_phase)) -
                    (&v map_scale        (fcalc+fbulk))))
          (ref_active=1 and acentric)
       do (fmap=    (&u   map_fom
combine(amplitude(&STRIP%obs_f),map_phase)) -
                    (&v map_scale        (fcalc+fbulk)))
          (ref_active=1 and centric)
     else
       do (fmap=(&u             map_fom
combine(amplitude(&STRIP%obs_f),map_phase)) -
                (&v           map_scale        (fcalc+fbulk)))
          (ref_active=1 and acentric)
       do (fmap=(max((&u-1),0)   map_fom
combine(amplitude(&STRIP%obs_f),map_phase)) -
                (max((&v-1),0) map_scale        (fcalc+fbulk)))
          (ref_active=1 and centric)
       if ( &fill_in = true ) then
         do (fmap=(&u-&v) map_scale (fcalc+fbulk))
            (&low_res >= d >= &high_res and ref_active # 1)
       end if
```

```
    end if
  end

end if

xray
  declare name=map domain=real end
  if ( &map_type = "gradient" ) then
    do (map=ft(fmap)) (ref_active=1)
  elseif ( &u = &v ) then
    do (map=ft(fmap)) (ref_active=1)
  elseif ( &fill_in = true ) then
    do (map=ft(fmap)) (&low_res >= d >= &high_res)
  else
    do (map=ft(fmap)) (ref_active=1)
  end if
end

if ( &write_coeff = true ) then
  evaluate ($coeff_out=&output_root + ".coeff")
  xray
    declare name=map_coeff domain=reciprocal type=complex end
    do (map_coeff=fmap) (all)
    write reflection
      output=$coeff_out
      if ( &map_type = "gradient" ) then
        sele=(ref_active=1)
      elseif ( &fill_in = true ) then
        sele=(&low_res >= d >= &high_res)
      else
        sele=(ref_active=1)
      end if
      map_fom map_phase map_scale map_coeff
    end
    undeclare name=map_coeff domain=reciprocal end
  end
end if

xray
  undeclare name=map_phase domain=reciprocal end
  undeclare name=map_fom   domain=reciprocal end
  undeclare name=map_scale domain=reciprocal end
end

xray
  undeclare name=dtarg domain=reciprocal end
  undeclare name=fcave domain=reciprocal end
  undeclare name=fpave domain=reciprocal end
  undeclare name=foave domain=reciprocal end
  undeclare name=dave  domain=reciprocal end
  undeclare name=total domain=reciprocal end
  undeclare name=fmap  domain=reciprocal end
end

if (&map_scale=true) then
```

```
  xray
    show rms (real(map)) ( all )
    do (map=map/$result) ( all )
  end
end if

if ( &real_r = true ) then
  xray
    declare name=mod_map domain=real end

    do (mod_map=ft(fcalc+fbulk)) (&low_res >= d >= &high_res)

    if (&map_scale=true) then
      show rms (real(mod_map)) ( all )
      do (mod_map=mod_map/$result) ( all )
    end if

    declare name=corr_map domain=real end
    declare name=map11    domain=real end
    declare name=map12    domain=real end
    declare name=map22    domain=real end
    declare name=prop     domain=real end
    declare name=dist     domain=real end
    declare name=mask     domain=real end
  end

  do (store9=0) (all)
  evaluate ($counter=1)
  for $id in id ( tag and &atom_real ) loop main
    do ( store9=$counter ) ( byres( id $id) )
    evaluate ($counter=$counter+1)
  end loop main

  xray
    mask
      mode=vdw
      solrad=0.1
      shrink=0.1
      nshell=1
      to=mask
      selection=( &atom_select )
    end

    do (prop=0) (all)
    proximity
      from=store9
      distance_map=dist
      property_map=prop
      cutoff=3.0
      selection=( &atom_real )
    end

    do (map11=gave(real(map) * real(map), real(prop)) )
                              ( real(prop) > 0 and real(mask) <= 0 )
    do (map12=gave(real(map) * real(mod_map), real(prop)) )
```

```
                              ( real(prop) > 0 and real(mask) <= 0 )
     do (map22=gave(real(mod_map) * real(mod_map), real(prop)) )
                              ( real(prop) > 0 and real(mask) <= 0 )
     do (corr_map=real(map12)/ sqrt ( real(map11) * real(map22) ) )
                              ( real(prop) > 0 and real(mask) <= 0 )
   end

   evaluate ($display=&output_root + "_r.list")
   set display=$display end

   display    #    resid    segid    CC    R-value

   evaluate ($counter=1)
   for $id in id ( tag and &atom_real ) loop real
     xray
       show ave (real(corr_map)) (real(prop)=$counter and real(mask) <=
0)
     end
     evaluate ($corr=$result)
     evaluate ($realr=1-$corr)
     show (resid) (id $id)
     evaluate ($resid=$result)
     show (segid) (id $id)
     evaluate ($segid=$result)
     display $counter[i6]   $resid[a4]   $segid[a4] $corr[f6.3]
$realr[f6.3]
     evaluate ($counter=$counter+1)
   end loop real

   xray
     undeclare name=mod_map  domain=real end
     undeclare name=corr_map domain=real end
     undeclare name=map11    domain=real end
     undeclare name=map12    domain=real end
     undeclare name=map22    domain=real end
     undeclare name=prop     domain=real end
     undeclare name=dist     domain=real end
     undeclare name=mask     domain=real end
   end
 end if

 set remarks=reset end
 set remarks=accumulate end

 xray
   show sum (1) (tst_active=1)
   if ( $result > 0 ) then
     evaluate ($test_exist=true)
   else
     evaluate ($test_exist=false)
   end if
 end

 evaluate ($remark="")
 if ( $struct > 1 ) then
```

```
   evaluate ($remark="averaged ")
 end if
 if ( &map_type = "unweighted" ) then
   evaluate ($remark=$remark + "(" + encode(&u) +  " |Fo| - " +
                                     encode(&v) + " k|Fc|)e^(i
phi_calc)")
 elseif ( &map_type = "sigmaa" ) then
   evaluate ($remark=$remark + "(" + encode(&u) + " m|Fo| - " +
                                     encode(&v) + " D|Fc|)e^(i
phi_calc)")
   if ( $test_exist = true ) then
     evaluate ($remark=$remark + " cross-val.")
   end if
   evaluate ($remark=$remark + " sigmaa")
 elseif ( &map_type = "combined" ) then
   evaluate ($remark=$remark + "(" + encode(&u) + " m|Fo|)e^(i
phi_comb) - " +
                                "(" + encode(&v) + " D|Fc|)e^(i
phi_calc)")
   if ( $test_exist = true ) then
     evaluate ($remark=$remark + " cross-val.")
   end if
   evaluate ($remark=$remark + " sigmaa")
 elseif ( &map_type = "observed" ) then
   evaluate ($remark=$remark + "(" + encode(&u) +  " m|Fo|)e^(i
phi_obs) - " +
                                  "(" + encode(&v) + " k m|Fc|)e^(i
phi_calc))")
 elseif ( &map_type = "gradient" ) then
   evaluate ($remark=$remark + "( d(" + &reftarget + ")/dFc )")
 end if

 evaluate ($remark=$remark + " map")

 if ( $total_test > 0 ) then
   remark $remark  r= $map_r[f6.4] free_r= $map_free_r[f6.4]
 else
   remark $remark  r= $map_r[f6.4]
 end if

 if ( &obs_type = "intensity" ) then
   remark reflections with Iobs/sigma_I < &sigma_cut rejected
   remark reflections with Iobs > &obs_rms * rms(Iobs) rejected
 else
   remark reflections with |Fobs|/sigma_F < &sigma_cut rejected
   remark reflections with |Fobs| > &obs_rms * rms(Fobs) rejected
 end if
 xray anomalous=? end
 if ( $result = true ) then
   remark anomalous diffraction data was input
 end if
 remark theoretical total number of refl. in resol. range:
$total_theor[I6] ( 100.0 % )
 remark number of unobserved reflections (no entry or |F|=0):
$unobserved[I6] ( $per_unobs[f5.1] % )
```

```
 remark number of reflections rejected:
$rejected[I6] ( $per_reject[f5.1] % )
 remark total number of reflections used:
$total_used[I6] ( $per_used[f5.1] % )
 remark number of reflections in working set:
$total_work[I6] ( $per_work[f5.1] % )
 remark number of reflections in test set:
$total_test[I6] ( $per_test[f5.1] % )

 if ( &bulk_sol = true ) then
   remark bulk solvent: density level= $sol_k_ref e/A^3, B-factor=
$sol_b_ref A^2
 else
   remark bulk solvent: false
 end if

 if ( &bscale # "no" ) then
   remark B-correction resolution: &low_res_bscale - &high_res
 else
   remark initial B-factor correction: none
 end if

 if ( &bscale = "anisotropic" ) then
   remark initial B-factor correction applied to &STRIP%obs_f :
   remark      B11= $aniso_11 B22= $aniso_22 B33= $aniso_33
   remark      B12= $aniso_12 B13= $aniso_13 B23= $aniso_23
   remark B-factor correction applied to coordinate array B:
$trace[f8.3]
 elseif ( &bscale = "anisotropic_fixed_anisotropic" ) then
   remark initial B-factor correction applied to &STRIP%obs_f :
   remark      B11= $aniso_11 B22= $aniso_22 B33= $aniso_33
   remark      B12= $aniso_12 B13= $aniso_13 B23= $aniso_23
   remark no B-factor correction applied to coordinate array B
 elseif ( &bscale = "isotropic" ) then
   remark B-factor correction applied to coordinate array B:
$trace[f8.3]
 end if

 if ( &write_map = true ) then

   evaluate ($filename=&output_root + ".map")

   if ( &map_mode = "asymmetric" ) then
     evaluate ($map_mode_string=ASYM)
   elseif ( &map_mode = "unit" ) then
     evaluate ($map_mode_string=UNIT)
   elseif ( &map_mode = "box" ) then
     evaluate ($map_mode_string=BOX)
   elseif ( &map_mode = "fract" ) then
     evaluate ($map_mode_string=FRAC)
   else
     evaluate ($map_mode_string=MOLE)
   end if

   xray
```

```
    write map
      if ( &map_format = "ezd" ) then
        type=ezd
      else
        type=cns
      end if
      automatic=false
      from=map
      output=$filename
      cushion=&map_cushion
      selection=&atom_map
      extend=$map_mode_string
      if ( &map_mode = "box" ) then
        xmin=&xmin xmax=&xmax
        ymin=&ymin ymax=&ymax
        zmin=&zmin zmax=&zmax
      end if
      if ( &map_mode = "fract" ) then
        xmin=&xmin xmax=&xmax
        ymin=&ymin ymax=&ymax
        zmin=&zmin zmax=&zmax
      end if
    end
  end
end if

if ( &peak_search = true ) then
  evaluate ($filename=&output_root + "_positive.peaks")
  xray
    peakpik
      from=map
      mpeak=&peak_num
      selection=( all )
      atom=true
      proximity=(&atom_map)
    end
  end

  write coor output=$filename selection=(segid=PEAK) end

  delete sele=(segid=PEAK) end

  evaluate ($filename=&output_root + "_negative.peaks")
  xray
    do (map=-map) ( all )
    peakpik
      from=map
      mpeak=&peak_num
      selection=( all )
      atom=true
      proximity=(&atom_map)
    end
  end

  write coor output=$filename selection=(segid=PEAK) end
```

Vita

Matthew Alton Wilkerson is an East Tennessee native. He was born in Morristown, Tennessee on August 1, 1977 to Dennis and Alta Wilkerson. He graduated from Morristown Hamblen County High School West in 1995. Afterwards he attended Carson Newman College where he received a B.A. in Biology. In 1999, he was married to Karen Wilkerson. Also in 1999, Matt entered the department of Biochemistry, Cellular, and Molecular Biology at the University of Tennessee. He graduated with a Masters of Science degree in December 2003. Matt is now a professor of Biology at Walter State Community College in his hometown of Morristown.