The Development of Tools to Investigate Relationships between Atomic Contacts in Proteins

Robert W. Su

A thesis submitted in partial fulfillment of the
requirements for the degree of:

Master of Science in Bioengineering

University of Washington

2013

Committee:

Valerie Daggett

Christopher Neils

Program Authorized to Offer Degree:

Department of Bioengineering

University of Washington

Abstract

The Development of Tools to Investigate Relationships between Atomic Contacts in Proteins

Robert W. Su

Chair of the Supervisory Committee:

Valerie Daggett

Bioengineering

The time-varying states of atomic contacts reflect the dynamics of the protein. Protein dynamics are linked to protein functions. Thus, it is crucial to have automatic methods for extracting the relationships between atomic contacts. In this study, three methods were used to extract the relationships between atomic contacts, reverse engineering based on Bernoulli mixture models (ReBmm), correlated motion, and Leader Finder. The protein used in this study was Cu/Zn superoxide dismutase (SOD1), and its aggregations is associated with amyotrophic lateral sclerosis (ALS). Its wild type (WT) form and mutant type (A4V) were simulated using all-atom molecular dynamics (MD) simulations. The results from the three methods suggested m the aggregation of SOD begins with the downward movements in the electro static loop and propagated to the Zn binding loop, and several strands of the β-sheet. As a result, the buried surface area between the front sheets and the back sheets became exposed and may be the cause of aggregation. The three methods, ReBMM, correlated motion, and Leader Finder proved to be useful tools for studying the relationship between atomic contacts in protein simulations.

TABLE OF CONTENTS

1. INTRODUCTION

Proteins fold into their functional tertiary structures based on their amino acid sequences. The process is known as protein folding. Protein motion, i.e. dynamic structures over time is linked to function. Proteins can be enzymes, hormones, antibodies *etc*. Since proteins play many crucial roles in the human body, proteins can also be the cause of many diseases when they misfold or do not function properly. Many diseases are caused by protein unfolding/misfolding and aggregation [1, 2]. Protein misfolding can also occur due to a single mutation [1, 3, 4]. If protein folding and unfolding can be fully understood, then drugs could be developed to prevent protein misfolding and aggregation. Therefore, by understanding protein structure, function and dynamics, there is a chance to improve human health.

Molecular dynamics (MD) simulation is the most realistic computational method available, and it has been applied to protein folding and unfolding [5]. MD simulations provide high resolution protein structures. However, the amount of data generated hundreds of terabytes is a challenge to analyze [5, 6]. In 2007, Daggett's group generated a database called Dynameomics [6-8]. The goal of the Dynameomics database is to capture the unfolding pathways of representatives of essentially all known protein structures. Since nature reuses protein folds to construct new proteins, we gain insight into how most proteins work by studying these folds as they represent 95% of all protein folds. With such high coverage of proteins, in theory general patterns in protein folding and unfolding can be found by analyzing the data in the Dynameomics database. Software developed for this project can speed up researcher's ability to analyze MD data. With the increased efficiency in analyzing large data, more and longer simulations can be done while reducing the burden on scientists to interpret them. Thus, the heavy lifting can be transferred from scientists to computers. This speedup can generate more scientific significant results in a short time and potentially reduce the cost of drugs that target diseases caused by protein misfolding.

It is beneficial to understand how proteins fold and unfold. By understanding protein folding and unfolding, novel proteins can be engineered to give the shape, motion, and function desired that are not available in nature. Currently, there are many experimentally determined protein structures in the Protein Data Bank [9]. High resolution protein structures are obtained by X-ray crystallography and NMR spectroscopy [10]. X-ray crystallography is a common technique used to determine the arrangement of atoms in a protein. X-ray crystallographer emits x-rays to a protein and crystallographer can use software to estimate the arrangement of atoms based on the diffracted beams. [11]. NMR spectroscopy uses chemical shifts of every atom in a protein and structural constraints to construct the protein's 3D structure [12]. Both methods have their strengths and limitations. X-ray crystallography can only construct protein structure in crystal form; thus the structure obtained may be different from the protein in a biological assay [11]. NMR spectroscopy can be used to study a protein's structure in solution [12]. Both methods define the structure of mobile regions in a protein poorly [12] and are restricted by experimental conditions [11, 12]. For example, NMR spectroscopy is limited by the solution pH and temperature [12] , and X-ray crystallography cannot be used to obtain structures in solution [11].

Techniques have been developed to understand the process of protein folding and unfolding. One way is through the ϕ-value analysis [13]. For a number of small proteins, the pathway of protein folding/unfolding can be simplified to denatured state, transition state, and the native state [13]. ϕ-value analysis provides structural information about the transition state of the protein folding/unfolding pathway [13]. ϕ-value analysis indicates whether each residue in a protein has a native-state-like conformation or a denature-state-like conformation; it does this by finding the ratio between the change in the transition state energy and change in the folded state energy after a subtle mutation has been made to the target residue [13]. By knowing the ϕ-value of each residue, researchers can predict the protein's structure in the transition state ensemble, which will help in the understanding of protein folding and unfolding [13].

Another way to study protein folding and unfolding is through MD simulations. Due to the limitations of X-ray crystallography and NMR spectroscopy, protein structures can only be obtained for a limited number of states [11, 12]. MD simulation can calculate a protein's structure in a wide range of states [5, 13, 14]. MD simulation can set the solvent, temperature and pH level; this is impossible to do with X-ray crystallography and NMR spectroscopy [5, 11, 12]. Thus, MD simulation can generate more protein structures more rapidly. Another advantage of MD simulation is its sampling rate. MD simulations can sample at any rate since this can be set by the researcher [5]. However, higher sampling rates are more computationally expensive. With a high sampling rate and the freedom to set the solution temperature and pH, the process of protein folding and unfolding can be observed directly. MD simulation also provides atomic resolution going from the denatured state to the native state because it contains every atom's coordinate at each round of calculation [5]. ϕ-value analysis is often used to check the validity of MD simulations [13].

There are three main models of protein folding: diffusion-collision, nucleation-condensation, and hydrophobic collapse [13]. In diffusion-collision, secondary structures forms first and guides the unfolded structure to the native structure [13]. In hydrophobic collapse, hydrophobic interactions forms first and guides the unfolded structure to the native structure [13]. Nucleation-condensation is a hybrid version of both [13]. All three models suggest that folding occurs in a cooperative manner where some initial interactions can act as a driving force for the folding process. With MD simulation, such interactions can be described as atomic contacts. Atomic contacts can be determined by a threshold distance between two atoms. In this study, the relationships between atomic contacts are studied and used to explain protein unfolding.

2. BACKROUND

In 2012, I developed a Windows application called the Protein Dashboard for visualizing protein simulations stored in the Dynameomics database [6-8]. The Protein Dashboard was built as a plugin for the in-house software program called DIVE (Data Intensive Visualization Engine). DIVE is a software framework with smooth data streaming between plugins. As a plugin of DIVE, the Protein Dashboard benefited from the data feature of DIVE and thus was able to visualize data from the Dynameomics database in an object-orientated fashion (Figure 1). The Protein Dashboard has molecular 3D rendering capabilities that visualize the protein structure at any given frame in a protein simulation. It also has multiple charting tools for visualizing calculated properties for any simulation such as root-mean-square deviation (RMSD), root-mean-square fluctuation (RMSF), Contact Count, solvent accessible surface area (SASA), Phi Psi, Define Secondary Structure of Proteins (DSSP), Residue Contact Map, and Atom Contact Map. Users were able to interact with the charting tools such as a left click or double click and expect the 3D render to respond by zooming into the corresponding residue, atom, or time frame.

The Protein Dashboard serves as the visualization tool for all the results generated in this project. The ability to visualize custom results from this project demonstrates the potential and usability of both DIVE and the Protein Dashboard.

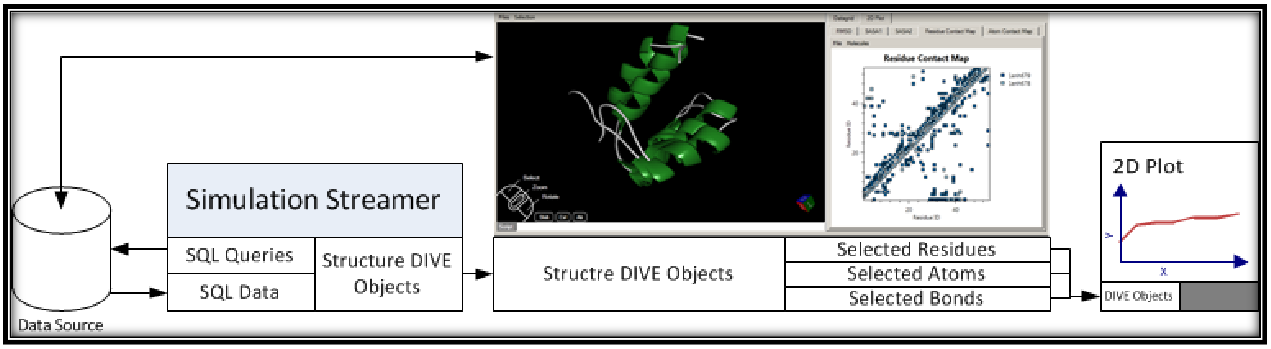


Figure 1: Workflow of Protein Dashboard. Another DIVE plugin fetch data from the Dynameomics database and sends data downstream to the Protein Dashboard where 3D structures and 2D plots are generated. Protein Dashboard can also push its processed data downstream to other DIVE plugins.

3. RELATED WORK

There has been much research that focus on extracting information from atom-atom/residue-residue contacts. Mintseris *et al.* used atomic contacts in the regions of protein-protein interactions to classify protein-protein interactions [15]. Schmidlin *et al.* used atom-atom/residue-residue contacts to study the structural differences between the wild type superoxide dismutase protein and its mutant form A4V [16, 17]. Vassura *et al.* used a residue contact map to reconstruct the protein structure in 3D space with accuracy as high as 71% [18]. There is even visualization software dedicated to visualize contact maps such as CMView [19] and CMWeb [20].

In the Daggett lab, we have been constantly developing methods to extract the relationships between atom-atom/residue-residue contacts found throughout protein simulations. A PhD student, Denny Bromley has developed an algorithm called Contact Walker that uses the depth-first-search to identify the residue-residue contacts that are significantly different between two simulations. However, to the best of our knowledge, there is no algorithm that can identify the relationships between atomic contacts at any given time in a protein simulation.

4. METHODS TESTED

4.1 Data Set

4.1.1 Data Set Background

ALS is a lethal neurodegenerative disease with a mean survival time of three to five years [21]. Patients with ALS ultimately die from respiratory failure. About 90% - 95% of ALS cases have unknown causes and are categorized as sporadic ALS (sALS). For the remaining of the cases, there is a family history of the disease and thus termed familial ALS (fALS). 20% - 25% of fALS are linked to over 100 different point mutations scattered throughout the Cu-Zn superoxide dismutase protein (SOD1) [22].

SOD1 is a copper enzyme responsible for converting superoxide radicals to oxygen and hydrogen peroxide through a two-step reaction [22]. Researchers believe that SOD1 mutants cause fALS by two possible mechanisms [23, 24]. SOD1 mutants can cause misfolding and those misfolded proteins alter the catalytic reaction to allow production of oxidants that are toxic to proteins, nucleic acids, and lipids [23]. SOD1's aggregation can also be toxic [23]. SOD1 aggregation is believed to happen in the following order, dimer dissociation of SOD protein, loss of metal ions and aggregation occurs [23]. These two mechanisms are not mutually exclusive [23].

SOD1 is a dimeric protein. Each subunit contains a copper and zinc ion. The two subunits are held together mainly by hydrophobic forces and some hydrophilic interactions. The interacting parts include the N-terminal and the C-terminal β-strands and two loops involving residues 49-54 (loop IV) and 102-114 (loop VI) [25]. The monomeric form of SOD has a classical eight-stranded Greek key β-barrel [26]. Copper ion in its reduced form is coordinated by His46, His48, and His120 [26]. Copper ion in its oxidized form has an exact contact with His63 which also coordinates the Zinc ion [26]. Zinc ion is coordinated by His63, His71, His80, and Asp83 [26].

Rakhit et al. suggested a possible energy profile of metal-catalyzed oxidation induced SOD1 aggregation. In that energy profile, SOD1 mutants were believed to lower the dimer dissociation energy barrier and thus are likely to cause SOD1 aggregation [23]. Khare et al. found low PH lowered the dimer dissociation energy barrier and make SOD1 aggregation more likely [24]. The other energy barrier in Khare et al.'s energy profile includes metal loss and aggregation [24]. There have been studies trying to explain how the most dangerous mutant A4V cause aggregation by looking at its structural changes near the dimer binding residues, Copper binding residues, and zinc binding residues both experimentally [27] and computationally [1, 28]. The A4V mutation site is close to the dimer interface. The disruption causes SOD proteins to break down to their monomeric forms, which can lead to aggregation. Thus, the analysis of this paper is focused on residues/atoms that are involved in the dimer interface, copper binding, and zinc binding.

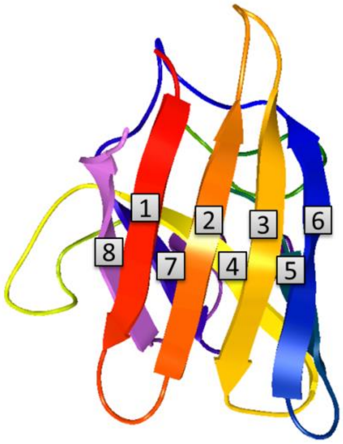


Figure 2: A cartoon view of the SOD1 structure. All 8 beta strands are labeled.

4.1.2 Data Set Source/Format

The A4V SOD1 starting structure was obtained from a 1.9 Å crystal structure of the A4V mutant of human SOD1 (1UXM Chain A) [29]. Simulations of A4V SOD1 were performed with the *in lucem* molecular mechanics (*ilmm*) [30] simulation software using protocols described elsewhere [5, 30-34]. 3 simulations for A4V SOD1 at neutral pH and 310 K were used. Three simulations for WT SOD1 at neutral pH and 310K were used. Atoms were considered to be in contact if the heavy atom distance was < 4.6Å, except for C-C, where the cutoff was 5.4 Å [1]. Atomic contacts were determined every picosecond in all 6 simulations.

4.2 ReBMM

Table 1: Parameter definitions for ReBmm algorithm.

N	Number of atomic contacts
M	Number of time points
K	Number of parents
B	Number of Bernoulli mixtures
θ	Probability distribution for a Bernoulli mixture
S	A time series vector
s	A contact's on/off state
$\rho(s\|\theta)$	Expected value for one Bernoulli mixture
$E(s)$	Expected value for multiple Bernoulli mixtures
Z	Hidden variable
π_b	Prior for the k_{th}distribution
γ	Regularization constant
α	Threshold used to select parents

Kauffman was the first person to model genetic data with Boolean network in the sixties [35]. A Boolean network $G(V,F)$ is defined by a set of nodes $V = \{x_1, \dots, x_n\}$ and a list of Boolean functions $F = (f_1, \dots, f_n)$. Each node can have two states, on or off. A Boolean function $f_i(x_{i1}, \dots, x_{ik})$ can have up to k number of parent nodes. The number of parent nodes for every function does not need to be the same.

A Boolean network, unlike a continuous model, has a finite search space of 2^{2k}. Also, the deterministic nature of Boolean network fits the in/out contact nature of atomic contact data. A search space minimal sets algorithm f 2^{2k} can be daunting. However, there have been several studies that suggest the biological data represents a scale-free network, which means k is roughly $\log_2 n$. [35-37].

Boolean network is often used in genetic studies, in this project the genetic data is replaced with atomic contact data. Both types of data have only two states. To the best of our knowledge, we are the first one to model atomic contact networks with Boolean network. There have been many attempts to infer genetic regulatory networks from microarray time series data [38-43]. REVerse Engineering ALgorithm (REVEAL) proposed by Liang [40] was the first systematic method to infer a complete Boolean network. Given a fixed number k, REVEAL finds the most probable parents by calculating the Shannon entropy for all possible combination of parents. It is remarkably robust but has an extraordinarily high running time of $O(2^{2k})$. After REVEAL, there have been many new algorithms for extracting Boolean networks from time series data such as Best-fit proposed by Lähdesmäki [41], the predictor chooser method [44], Monte-Carlo type randomized algorithm [45]. All of them have k, the number of parent nodes in the exponential term.

In 2011, Maucher *et al.* proposed the correlation method to extract parent nodes from time series data. [42] The simplicity of correlation calculation gave the parent node identification process linear running time. However, they did not offer any methods for inferring rules form the identified parents. In 2012, Saeed *et al.* proposed an algorithm called reverse engineering of Boolean networks using Bernoulli mixture models (ReBMM) [43]. Their method is based on learning Bernoulli mixture models from time series data, and these mixtures are then used for determining the network structure. ReBMM is the first method that has a quadratic running time.

ReBMM was chosen among the algorithms researched to extract Boolean networks from atomic contact data because of its much improved running time.

Bernoulli Model

A Bernoulli mixture model is an extension of a simple Bernoulli distribution. A Bernoulli distribution is used to model binary data. A single univariate Bernoulli distribution has only one parameter, i.e., the probability ρ that the value of the random variable is true. The probability ρ that the value of the variable is false is then $1 - \rho$. The expected probability of seeing the given binary data is then described in Equation 1.

$$p(s|\theta) = \sum_{j=1}^{N} (p_j)^{s_j} * (1 - p_j)^{1-s_j}$$

Equation 1: The expected value for one Bernoulli mixture. p_j is the probability of seeing 1 for sample j. s_j is the value of contact S at jth time point. θ is the probability table for all S. The probability table is randomly generated in the beginning unless given.

In a multivariate Bernoulli distribution, each distribution has its own set of parameters. Thus Equation 1 derives to Equation 2. The π_b term is the prior knowledge of the b^{th}distribution.

$$p(s) = \sum_{B=1}^{B} \pi_b * \rho(s|\theta_b)$$

Equation 2: The expected value of all Bernoulli mixtures. π_b is the prior for the b^{th} mixture. The priors are evenly distributed and sum up to 1.

Calculating the probability vectors

The probability vector of each Bernoulli mixture was learned through a well-known optimization technique called expectation maximization EM. When calculating the probability vector of a target contact, its first data point was omitted because of the lag of 1 assumption. A lag of 1 meant that a target contact's on/off state was determined by its parents' on/off states from exactly one time frame ago. The probability vectors were then used to extract rules for the target contact.

The goal of the E-Step was to calculate the hidden Z matrix given the priors of each Bernoulli mixture, probability matrix, and time series data. The Z matrix is defined in Equation 3. The M-step then re-estimated the priors (Equation 4) and probability matrix (Equation 5) for each Bernoulli distribution given the newly calculated Z matrix.

$$Z_{ij} = \frac{\pi_b * p(s_i|\theta_b)}{\sum_{b=1}^{B} \pi_b * p(s_i|\theta_b)}$$

Equation 3: Definition of the Z hidden variable. Z matrix is an M by B matrix.

$$\pi(b) = \frac{\sum_{i=1}^{M} Z_{ib} * (1 + \gamma ln Z_{ib})}{\sum_{b=1}^{B} \sum_{i=1}^{M} Z_{ib} * (1 + \gamma ln Z_{ib})}$$

Equation 4: The definition of the bth prior. γ is the regularization term.

$$p_{(b)}= \frac{\sum_{i=1}^{M} Z_{ib} * (1 + \gamma ln Z_{ib}) * Si}{\sum_{b=1}^{B} \sum_{i=1}^{M} Z_{ib} * (1 + \gamma ln Z_{ib})}$$

Equation 5: The definition of a probability table for a Bernoulli mixture.

Inferring Rules

Once the EM step was completed, the calculated probability matrix transformed to main matrix by Equation 6. α is a threshold parameter and it had a strong effect on the rule extracted. Main matrix was then used to extract Boolean rules for the target contact. Main matrix was simplified by examining all its columns. The columns that had all 1s or 0s were removed because it did not play a rule in the determination of target contact's Boolean state.

$$V_j \begin{cases} 1\ if\ p_j > \ \alpha \\ 0\ if p_j < 1 - \ \alpha \\ \emptyset\ othereise \end{cases}$$

Equation 6: Categorization of probability. $\emptyset$ is a don't-care value. α is a threshold parameter.

ReBMM method is summarized in algorithm 1 and 2. ReBMM was only applied to the A4V runs 1 and 2, and WT run 3 at 0-1 nanosecond and 0-2 nanosecond intervals. For each window size, two different EM iterations were used, 100 and 200. In this case, the residue-residue contacts were used instead of atom-atom contacts. Residue-residue contact is considered to be in contact as long as there is at least one atomic contact between them. Residue contact between 46 and 63 was chosen as the target of interest because residue 63 is involved in both Zn and Cu binding. The two different time ranges were used to compare the effect of having longer data points. ReBMM was repeated 30 times because ReBMM is not a deterministic method; multiple runs were required to ensure convergence.

Atomic contacts and longer time intervals were not considered because of the speed concern. Even though ReBMM already runs in polynomial time, it still took many hours to finish a one-nanosecond interval for residue-residue contacts. Using atomic contacts would dramatically extend the time requirement for this project.

Algorithm1: ReBMM

INPUT: $S^i, \dots, S^N$,time series data in 0 or 1 format, probability vector cutoff α, maximum number of parents k, and the target S.

OUTPUT: Boolean rule for target S.

```
for each time series $S^i$ do
    if ($S^i$ is all 1s or all 0s) continue;
    Partition the data into two matrices. One associated with 0 output value ($M_0^i$) and the other associated with 1 output value ($M_1^i$);
    Calculate the probability matrix, $P_0^i$ <- $EM(M_0^i)$ and $P_1^i$ <- $EM(M_1^i)$;
    Ignore the node that has all the same values in probability matrix and nodes that have the same values in both $P_0^i$ and $P_1^i$;
    Simplify a rule.
    Generate two Boolean rules from $V_0^i$ and $V_1^i$;
    Check accuracy of both Boolean rules;
    return Boolean rule with the highest accuracy;
```

Algorithm2: EM

INPUT: Number of iterations itr, Matrix data in Boolean format M , probability vector cutoff α, and maximum number of parents k.

```
OUTPUT: Probability matrix
Initialize prior and probability matrices if not given.
for $i = 0\ to\ itr$ do
    Calculate the hidden Z variables by Equation 3;
    Re-estimate the prior and probability matrices by Equation 4 and Equation 5;
end for
return the probability matrix;
```

4.3 Correlated Motion

Correlated motions between alpha carbons highlight the backbones of the protein that move in the same or opposite directions [50, 51]. Correlated motions between any two atom pair within a fixed time window can be calculated by Equation 7. Correlated motion for α-carbons, β-carbons, and γ-carbons were all calculated for all 6 simulations of SOD1 available on the Dynameomics database with a window size of 10ns. The correlated motion heat maps were visualized with the Protein Dashboard.

$$C_{i,j} = \frac{< \Delta r_i \, \Delta r_j >}{\sqrt{(< \Delta_{r_i}{}^2 \Delta_{r_j}{}^2 >)}}$$

Equation 7: Correlated motion between two atoms over a time window. $\Delta r_i$is the displacement of the ith atom away from its mean position over the time window. $\Delta r_j$is the displacement of the jth atom away from its mean position over the time window

4.4 Leader Finder

The Boolean networks extracted from atom-atom contact data or residue-residue contact data can allow researchers to predict the system's behavior. However, a Boolean network does not provide a ranking among the identified parent nodes. Identify high ranking leaders can guide researcher to the important nodes in complicated Boolean networks. Identifying popular leaders is a popular topic across many disciplines such as economic, sociology, and metrology [46, 47]. In this project, we adopted method from Kitagawa *et al.* [47] to identify popular leaders in our test set.

Our method for finding popular leaders was derived from Kitagawa and co-workers [47]. The main alteration was to calculate lag correlations with Boolean values instead of floating numbers [48]. Normally the correlation between two floating number vectors is calculated using Pearson correlation (Equation 8). However, when the size of atomic contact data becomes large with different lags considered, the Pearson correlation is simply not fast enough. There have been many studies on increasing the speed of Pearson calculation. Zhu *et al.* proposed of translating raw data into the Fourier domain [49]. Since Discrete Fourier Transform (DFT) is a linear transformation, the Euclidean distance is preserved. Also, for many real-world datasets, first few DFT coefficients capture most information of the original sequences. Thus, the correlation coefficient can be estimated with fewer calculations. Sakurai *et al.* proposed a speed-up method to calculate lag correlation by using data smoothing, and log-scale probing [46]. Data smoothing sacrifices accuracy to gain speed by decreasing the number of calculation required when calculating correlation coefficients. Log-scale probing decreases the numbers of lag correlation calculation needed. By only calculating lag correlation at 2^i time points up to$\frac{N}{2}$, the numbers of calculations decrease dramatically with some sacrifices of accuracy. The real lag correlations can then be estimated fitting a cubic curve the probed results. Zhu *et al.* and Sakurai *et al.* ‘s methods were not used because our data size of 30,000 atomic contacts by 60,000 time points would require a 7.2 gb ram. Boolean correlation calculation was chosen because of its speed, small memory usage, and the fact that our time series data already exist in binary format. The correlation of two Boolean vectors can be calculated by Equation 9. The correlation of two Boolean vectors with a lag l can be calculated by Equation 10.

$$R\left(S^i,\, S^j\right) = \frac{\sum_{i=1}^{n}(S^i - \overline{S^i})(S_j - \overline{S_j})}{\sqrt{\sum_{i=1}^{n}(S^i - \overline{S^i})^2}\,\sqrt{\sum_{i=1}^{n}\left(S_j - \overline{S_j}\right)^2}}$$

Equation 8: Pearson correlation calculation between two time series, S^i and S^j.

$$R\left(S^i,\, S^j\right) = \left(0.5 - \frac{\sum_{k=1}^{n} s_k^i \oplus\ s_k^j}{n}\right) * 2$$

Equation 9: Correlation calculation for two Boolean vectors.

$$R\left(S^i,\, S^j, l\right) = \left(0.5 - \frac{\sum_{k=1}^{n-l} s_k^i \oplus\ s_{(k+l)}^j}{n}\right) * 2$$

Equation 10: Correlation calculation for two Boolean vectors with lag l.

The leader finder method is summarized in algorithms 3, 4, and 5. Correlation calculations were carried out for every pair of S^i and S^j atomic contact data with lag 1 to 10. Among the correlations calculated with lag 1 to 10, the highest correlation value was stored if its absolute value was higher than the cutoff of 0.8. A high cutoff such as 0.8 can increase reliability of the leaders identified at the end by pruning out noises. A temporary graph was then constructed, where the nodes were the atomic contacts time series and the edges represent the causal relationships between each atomic contact. Leader scores were calculated for every leader identified by Equation 11. Time series were then sorted in descending order by the leader scores calculated. Remove descendants from every time series recursively from the graph. The remaining time series in graph were treated as the popular leaders. Leader finder algorithm was only applied to A4V simulation run 1 and 2 between 40 nanosecond and 50 nanosecond, WT simulation 2 between 30 nanosecond and 40 nanosecond, and WT simulation 3 between 40 nanosecond and 50 nanosecond. All calculations were performed at 100ps interval. The specific time windows were chosen based on the results from the correlated motion. Leader results were translated to be by-Residue by Algorithms 6 and 7, summed, normalized, and visualized using the Protein Dashboard.

$$score^{j} = \sum_{S^{i} \in L_{S^{j}}} \frac{score^{i} * R(S^{j}, S^{i})}{d_{out}(S^{i})}$$

Equation 11: Formula for calculating a time series S^j's leader score.

Algorithm3: DiscoverLeaders

INPUT: $S^i, \ldots, S^n$,time series data in 0 or 1 format, maximum lag m, correlation threshold γ.

OUTPUT: Leaders.

```
for every pair of time series $S^i$ and $S^j$ do
    Compute $R(S^i, S^j, l)$ for $|l| \leq m$;
    Store the highest $R(S^i, S^j, l)$;
end for
Construct graph $G$ with respect to $\gamma$;
$\pi$ <- Compute leader score vector by Equation 11;
$L$ <- $ExtractLeaders(G, \pi)$;
Return $L$;
```

Algorithm4: ExtractLeaders

INPUT: Graph G, score vector π

OUTPUT: Leaders

```
$L$ <- Sort time series in descending order by leader score vector.
for each time $S^i$ in $L$ do
    $RemoveDescendants(L, G, S^i)$;
end for
Return $L$;
end for
```

Algorithm5: RemoveDescendants

INPUT: Sorted time series L, graph G, time series S^j

```
for each time series $S^i$ in $L$ after $S^j$ do
    if($S^i, S^j$) is an edge in $G$ then
        $RemoveDescendants(L, G, S^i)$;
        Removes $S^i$ from $L$;
    end if
end for
```

Algorithm6: TranslateToResidueContact

```
INPUT: Sorted time series $L$, score vector $\pi$
OUTPUT: Translated time series $L$, score vector $\pi$
for each time series $S^i$ in $L$ do
    translate atom number to its parent residue's number;
end for
Combine duplicated translated time series and uses the sum of their score as their new score;
return translated time series and score vector $\pi$;
```

Algorithm7: TranslateToResidue

```
INPUT: Translated time series $L$, translated score vector $\pi$
Initialize a new score vector $\pi_R$for each residue;
for each time series $S^i$ in $L$ do
    Locate the indices in $\pi_R$ that holds the score for the two involving residues in $S^i$. Add
    The score for $S^i$ to the two indices in $\pi_R$;
end for
Normalize the score vector $\pi_R$;
for each score in score vector $\pi_R$ do
    Divide the score by the number of atomic contacts the corresponding residue holds;
return $\pi_R$;
```

5. EXPERIMENTAL VALIDATION OF METHODS

5.1 Boolean Network

ReBMM was applied to A4V run 1, A4V run 2, and WT run 3. For each simulation, two different time windows were investigated. The first time window was 0 to 1 nanosecond. The second time window was 0 to 2 nanoseconds. For each time interval, two different EM iterations were used, 100 and 200. Each analysis was repeated 30 times to ensure convergence.

The results shown in Table 2 and Table 3 suggest that the increase of EM iterations did not increase the accuracy. This implied that 100 EM iterations were enough to reach the optimal accuracy. The best rule had accuracy of 0.7 (Table 2), which meant the rule could only explain 70% of the data. This level of accuracy although better than randomly generated rules, was not high enough to convince researchers that the state of the 46-63 residue contacts could be explained by 57-60 and 137-140 residue contacts alone.

The results shown in Table 4 and Table 5 also suggest that 100 EM iterations were enough to reach rules with optimal accuracy. Accuracy topped around 65 % and was lower than the results found from the 0 to 1 nanosecond time window. The slight decrease in accuracy in the 0 to 2 nanosecond time window suggested that the network may be dynamic [52].

Table 2: Top five rules for A4V run 1. Time window was between 0 and 1 ns. EM iterations were 100. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	not ((not 57_60 and not 137_140))	0.7	0.36	0.9
46-63	not ((not 124_138 and not 137_140))	0.69	0.34	0.9
46-63	not ((35_93) or (not 90_93))	0.66	0.13	0.98
46-63	not ((90_94) or (not 90_94))	0.62	0	1
46-63	(33_95 and 90_93)	0.59	0.62	0.57

Table 3: Top five rules for A4V run 1. Time window was between 0 and 1 ns. EM iterations were 200. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	not ((35_93) or (not 90_93))	0.66	0.13	0.98
46-63	(33_95 and 90_93 and 90_94)	0.59	0.62	0.57
46-63	(33_95 and 90_93)	0.59	0.62	0.57
46-63	(33_95 and 95_97 and 90_94)	0.57	0.68	0.65
46-63	(14_37 and 33_95 and 36_93 and 37_93 and 38_93 and 93_95 and 95_97 and 90_93 and 90_94) or (not 14_37 and not 33_95 and not 35_93 and not 36_93 and not 37_93 and not 38_93 and not 93_95 and not 95_97 and not 90_93 and not 90_94)	0.54	0.75	0.42

Table 4: Top five rules for A4V run 1. Time window was between 0 and 2 ns. EM iterations were 100. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	(34_94 and 124_139 and 9_53) or (not 34_94 and not 124_139 and not 139_141 and not 127_129 and not 9_53) or (34_94 and 124_139 and 139_141 and 127_129 and not 9_53)	0.65	0.85	0.44

46-63	not ((not 9_53))	0.65	0.73	0.56
46-63	(9_53)	0.65	0.73	0.56
46-63	(34_94 and 44_120 and 124_139 and 139_141 and 122_139 and 127_129 and 9_53) or (34_94 and 44_120 and 124_139 and not 122_139) or (not 34_94 and not 124_139 and not 139_141 and not 122_139 and not 127_129 and not 9_53)	0.64	0.83	0.42
46-63	not ((85_124 and not 122_139 and not 3_23))	0.63	0.8	0.46

Table 5: Top five rules for A4V run 1. Time window was between 0 and 2 ns. EM iterations were 200. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	not ((not 57_60 and not 122_139 and not 3_23 and not 1_153))	0.65	0.6	0.7
46-63	not ((not 57_60 and not 122_139 and not 3_23))	0.65	0.6	0.7
46-63	not ((not 57_60 and not 122_139))	0.64	0.57	0.73
46-63	(34_94 and 44_120 and 124_139 and 139_141 and 122_139 and 127_129 and 9_53) or (34_94 and 44_120 and 124_139 and not 122_139) or (not 34_94 and not 124_139 and not 139_141 and not 122_139 and not 127_129 and not 9_53)	0.64	0.84	0.42
46-63	not ((85_124 and not 122_139 and not 3_23))	0.63	0.8	0.46

The results from A4V run 2 and WT run 3 (Table 6 -Table 13) all showed the same pattern as results from A4V run 1. 100 EM iterations were sufficient. Rules from the longer time windows had lower accuracies than the results from the shorter time windows, which suggested the contact network was dynamic. When the target contact spent most of the time in one state, ReBMM tend to generate dummy rules. Dummy rules were rules that always produce the same outcome. For example, the best rule from Table 6 was a dummy rule and it always generated the same state. It scored high 0.96 accuracy, but had a dismally low sensitivity score of 0. This kind of dummy rule had high accuracy, but did not offer any biological insights. It’s worth noting that the rules found from the same time intervals across different simulations were really different (Table 2 -Table 13). These differences suggested that proteins from different simulations move very differently at the early stages of MD simulations.

Overall, ReBMM generated rules with 60% to 70% accuracy. The low accuracy may be improved by using even smaller time windows, so the ReBMM algorithm can have a better chance capturing the contact network while it is static [52]. To use atom-atom contacts instead of residue-residue contacts can also boost up the accuracy. A residue to residue contact is considered to be in contact as long as there is one atomic contact between them. As a result, the effects of losing the majority of the atomic contacts between two residues are omitted when we use residue-residue contacts. Atomic contacts were not explored with ReBMM in this project due to its high computational cost. However, it will be a fascinating topic for future research.

Table 6: Top five rules for A4V run 2. Time window was between 0 and 1 ns. EM iterations were 100. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	(2_21) or (not 2_21)	0.96	1	0
46-63	not ((not 47_49 and not 58_143 and not 113_150 and not 114_150) or (47_49 and 58_143 and 113_150 and not 90_95))	0.74	0.74	0.69
46-63	not ((not 114_150))	0.76	0.76	0.79
46-63	not ((not 47_49 and not 58_143 and not 113_150 and not 117_119 and not 114_150) or (47_49 and 58_143 and 113_150 and 117_119))	0.505	0.49	0.72
46-63	not ((not 47_49 and not 58_143 and not 113_150 and not 114_150) or (47_49 and 58_143 and 113_150))	0.39	0.37	0.79

Table 7: Top five rules for A4V run 2. Time window was between 0 and 1 ns. EM iterations were 200. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	(2_21) or (not 2_21)	0.96	1	0
46-63	not ((not 117_119 and not 114_150))	0.77	0.77	0.74
46-63	not ((not 113_150 and not 117_119 and not 114_150))	0.79	0.79	0.66
46-63	not ((not 47_49 and not 58_143 and not 113_150 and not 114_150) or (47_49 and 58_143 and 113_150 and not 90_95))	0.744	0.74	0.69
46-63	not ((not 33_95 and not 47_49 and not 58_143 and not 113_150 and not 90_94 and not 114_150 and not 90_95) or (33_95 and 34_95 and 47_49	0.39	0.38	0.69

	and 58_143 and 95_97 and 90_94 and 90_95) or (47_49 and 58_143 and 113_150 and not 90_95))			

Table 8: Top five rules for A4V run 2. Time window was between 0 and 2 ns. EM iterations were 100. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	(not 117_119) or (47_49 and 113_151 and 117_119 and 114_150)	0.89	0.92	0.1
46-63	not ((not 25_27 and not 31_98 and not 113_150 and not 113_151 and not 114_150 and not 124_134 and not 125_134) or (31_98 and not 113_150 and not 114_150 and 124_134 and 125_134))	0.79	0.8	0.67
46-63	not ((not 25_27 and not 31_98 and not 113_150 and not 113_151 and not 114_150 and not 124_134 and not 125_134) or (25_27 and 31_98 and not 113_150 and not 113_151 and not 114_150 and 124_134 and 125_134) or (25_27 and 31_98 and 113_150 and 113_151 and 114_150 and not 124_134))	0.73	0.74	0.57
46-63	not ((not 47_49 and not 113_150 and not 113_151 and not 117_119 and not 1_151 and not 114_150))	0.7	0.7	0.67
46-63	not ((not 113_151 and not 114_150))	0.64	0.63	0.76

Table 9: Top five rules for A4V run 2. Time window was between 0 and 2 ns. EM iterations were 200. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	(1_22) or (not 1_22)	0.96	1	0
46-63	not ((not 47_49 and not 113_150 and not 113_151 and not 117_119 and not 114_150 and not 124_134 and not 125_134) or (not 47_49 and not 113_150 and not 113_151 and not 117_119 and 105_108 and not 114_150 and 125_134))	0.73	0.74	0.6
46-63	not ((31_98 and 32_97 and 32_98 and 125_127 and 134_139 and not 114_150 and 125_134) or (not 31_98 and not 32_97 and not 32_98 and not 113_151 and not 125_127 and not 134_139 and not 114_150 and not 125_134))	0.71	0.71	0.7
46-63	(32_97 and not 125_134) or (32_97 and 113_151 and 134_139 and 125_134) or (113_151 and not 134_139)	0.71	0.71	0.67
46-63	not ((not 47_49 and not 113_150 and not 113_151 and not 117_119 and not 114_150))	0.7	0.7	0.67

Table 10: Top five rules for WT run 3. Time window was between 0 and 1 ns. EM iterations were 100. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	(2_22 and 2_106 and 113_151 and 42_88 and 1_106) or (2_22 and 2_106 and 6_17 and 17_33 and 113_151 and 42_88 and not 1_106) or (2_22 and 2_106 and 6_17 and 17_33 and 113_151 and not 42_88 and not 1_106)	0.56	0.68	0.45
46-63	(not 15_36 and not 28_101 and not 29_101 and not 36_94 and not 70_135 and not 71_134 and not 71_137 and not 80_83 and not 101_104 and not 124_137) or (15_36 and 28_101 and 29_101 and 36_94 and 70_135 and 71_134 and 71_137 and 80_83 and 101_104 and 124_137)	0.56	0.65	0.46
46-63	(2_21 and 2_22 and 2_106 and 113_151 and 1_106) or (2_22 and 2_106 and 113_151 and not 1_106)	0.56	0.66	0.56
46-63	not ((not 16_35 and not 48_118))	0.55	0.4	0.7
46-63	(2_21 and 2_22 and 2_106 and 113_151 and 1_106) or (2_22 and 2_106 and 113_151 and not 1_106) or (not 2_21 and not 2_22 and not 2_106 and not 113_151 and not 1_106)	0.53	0.71	0.34

Table 11: Top five rules for WT run 3. Time window was between 0 and 1 ns. EM iterations were 200. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	(2_22 and 2_106 and 113_151 and 42_88 and 1_106) or (2_22 and 2_106 and 4_106 and 4_113 and 20_22 and 113_151 and 42_88 and not 1_106) or (2_22 and 2_106 and 4_106 and 4_113 and 20_22 and 113_151 and not 42_88 and not 1_106)	0.57	0.69	0.44
46-63	(2_21 and 2_22 and 2_106 and 113_151 and 1_106) or (2_22 and 2_106 and 6_17 and 17_33 and 113_151 and not 1_106)	0.56	0.63	0.48
46-63	(2_21 and 2_22 and 2_106 and 113_151 and 1_106) or (2_22 and 2_106 and 113_151 and not 1_106)	0.56	0.66	0.46
46-63	(2_21 and 2_22 and 2_106 and 113_149 and 113_151 and 1_106) or (2_22 and 2_106 and 113_149 and 113_151 and not 1_106) or (not 2_21 and not 2_22 and not 2_106 and not 113_149 and not 113_151 and not 1_106)	0.54	0.64	0.42
46-63	(2_21 and 2_22 and 2_106 and 113_151 and 1_106) or (2_22 and 2_106 and 17_33 and 113_151 and 1_106) or (2_22 and 2_106 and 6_17 and 17_33 and 113_151 and not 1_106) or (not	0.53	0.7	035

	2_21 and not 2_22 and not 2_106 and not 6_17 and not 17_33 and not 113_151 and not 1_106)			

Table 12: Top five rules for WT run 3. Time window was between 0 and 2 ns. EM iterations were 100. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	(9_145 and 9_146 and 48_62 and 50_116 and 87_89 and 87_99 and 128_134) or (9_145 and 9_146 and 47_64 and 48_62 and 48_64 and 50_116 and not 87_89 and not 87_99 and 128_134) or (9_145 and 9_146 and 47_64 and 48_62 and 48_64 and 87_89 and not 87_99 and not 128_134)	0.66	0.55	0.72
46-63	(9_145 and 9_146 and 48_62 and 50_116 and 87_89 and 87_99) or (9_145 and 9_146 and 47_64 and 48_62 and 48_64 and 50_116 and not 87_99)	0.66	0.6	0.69
46-63	(9_145 and 9_146 and 50_116 and 87_89 and 128_134) or (9_145 and 9_146 and 10_145 and 47_64 and 48_64 and 50_116 and not 87_89 and 128_134 and 10_144) or (9_145 and 9_146 and 10_145 and 47_64 and 48_64 and 87_89 and not 128_134)	0.66	0.63	0.67
46-63	(9_145 and 9_146 and 48_62 and 50_116 and 87_89 and 87_99) or (9_145 and 9_146 and 10_145 and 47_64 and 48_62 and 48_64 and 50_116 and not 87_99 and 10_144)	0.65	0.57	0.7
46-63	(9_145 and 9_146 and 50_116)	0.65	0.62	0.67

Table 13: Top five rules for WT run 3. Time window was between 0 and 2 ns. EM iterations were 200. 30 repetitions were carried to ensure convergence.

Target Contact	Rule Found	Percent Match	Sensitivity	Specificity
46-63	(9_145 and 9_146 and 48_62 and 87_89 and 87_99) or (9_145 and 9_146 and 48_62 and not 87_99)	0.65	0.65	0.65
46-63	(9_145 and 9_146 and 48_62 and 50_116 and 87_89 and 87_99 and 128_134) or (9_145 and 9_146 and 47_64 and 48_62 and 48_64 and 50_116 and not 87_89 and not 87_99 and 128_134) or (9_145 and 9_146 and 47_64 and 48_62 and 48_64 and 87_89 and not 87_99 and not 128_134)	0.66	0.56	0.72
46-63	(9_145 and 9_146 and 48_62 and 50_116 and 87_89 and 87_99) or (9_145 and 9_146 and 47_64 and 48_62 and 48_64 and 50_116 and not 87_99)	0.66	0.6	0.7
46-63	(9_145 and 9_146 and 50_116 and 128_134) or (9_145 and 9_146 and 10_145 and 47_64 and	0.66	0.64	0.67

	48_64 and 110_112 and not 128_134)			
46-63	(9_145 and 9_146 and 50_116 and 142_144)	0.64	0.53	0.7

5.2 Correlated Motion

Correlated motion between Cα-Cα, Cβ-Cβ, and C*γ*-C*γ* were calculated for all 6 simulations throughout the entire 60ns with window size of 10 ns. Cβ-Cβ and C*γ*-C*γ* were originally designed to capture movements between side chains. However, their heat maps turned out to be extremely similar to Cα- Cα's heat maps. The correlation value for each residue-residue pair in Cα- Cα and Cβ-Cβ had an average correlation around 0.9 (Figure 3). The correlation value for each residue-residue pair in Cα-Cα and C*γ*-C*γ* had an average correlation around 0.72 (Figure 4). Since many residue-residue pair correlations would be lost if C*γ*-C*γ* was used because not every residue had C*γ* atom. Also, Cα- Cα highly represented the results for C*γ*-C*γ*. Therefore, we only used Cα- Cα for studying correlated motion.

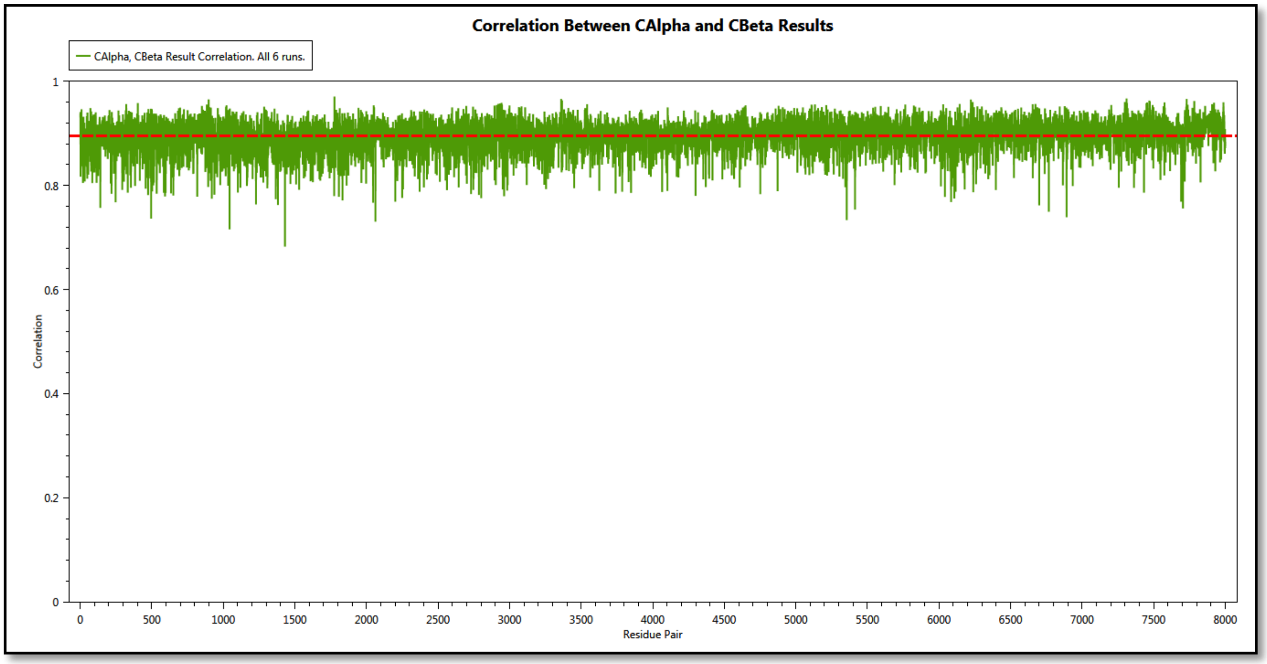


Figure 3: The correlation plot between results for Cα-Cα and Cβ-Cβ. The results from Cα-Cα and Cβ-Cβ turned out to have an average correlation with 0.9. This implied that Cα-Cα was very representative of Cβ-Cβ.

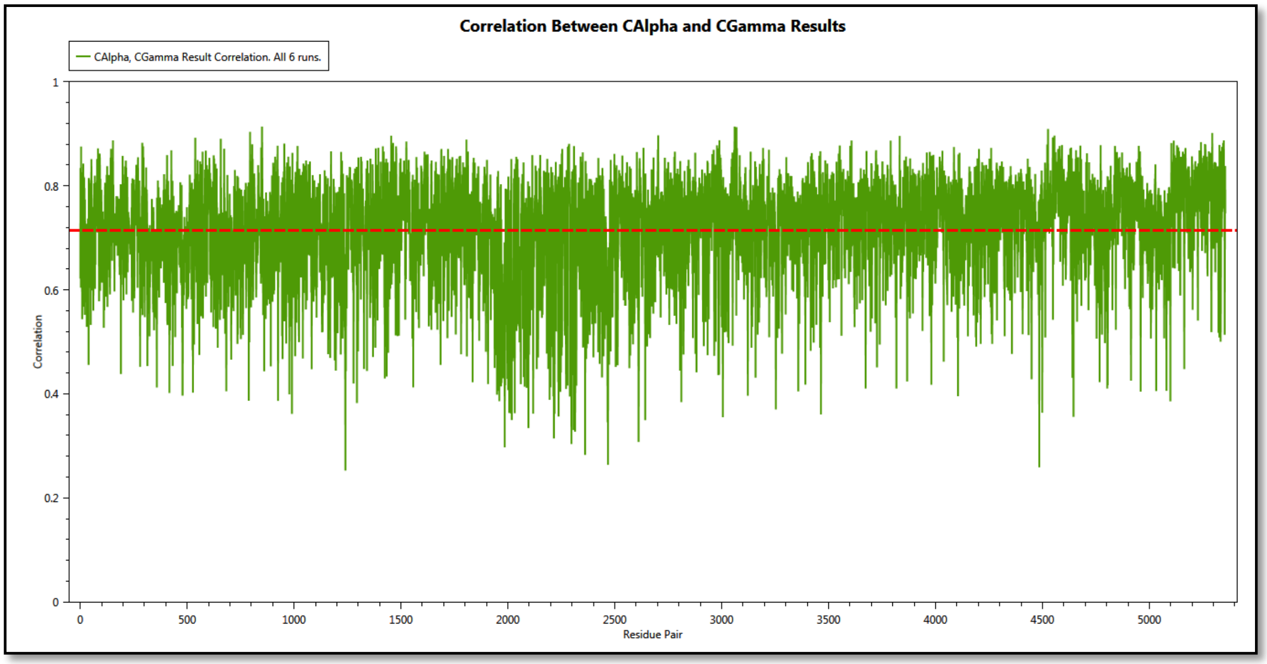


Figure 4: The correlation plot between results for Cα-Cα and Cγ-Cγ. The results from Cα-Cα and Cγ-Cγ turned out to have an average correlation with 0.72. This implied that Cα-Cα was also very representative of Cγ-Cγ.

For most of the 10ns intervals, there were very little negative correlations and off-diagonal correlations except for the first 10ns interval. However, the first 10ns intervals from all 6 simulations were very different. Four distinct 10ns intervals from 6 simulations were identified as interesting because they all shared a common pattern (Figure 5). A4V run 1's 40ns to 50ns, A4V run 2's 40ns to 50ns, WT run 2's 30ns to 40ns, and WT run 3's 40ns to 50ns showed strong positive correlation within the electrostatic loop (124-142). All those electrostatic loops had negative correlations with B4, B5, B7 and the Zn loop (67-79). The results here were used to narrow down the search windows for the Leader Finder algorithm.

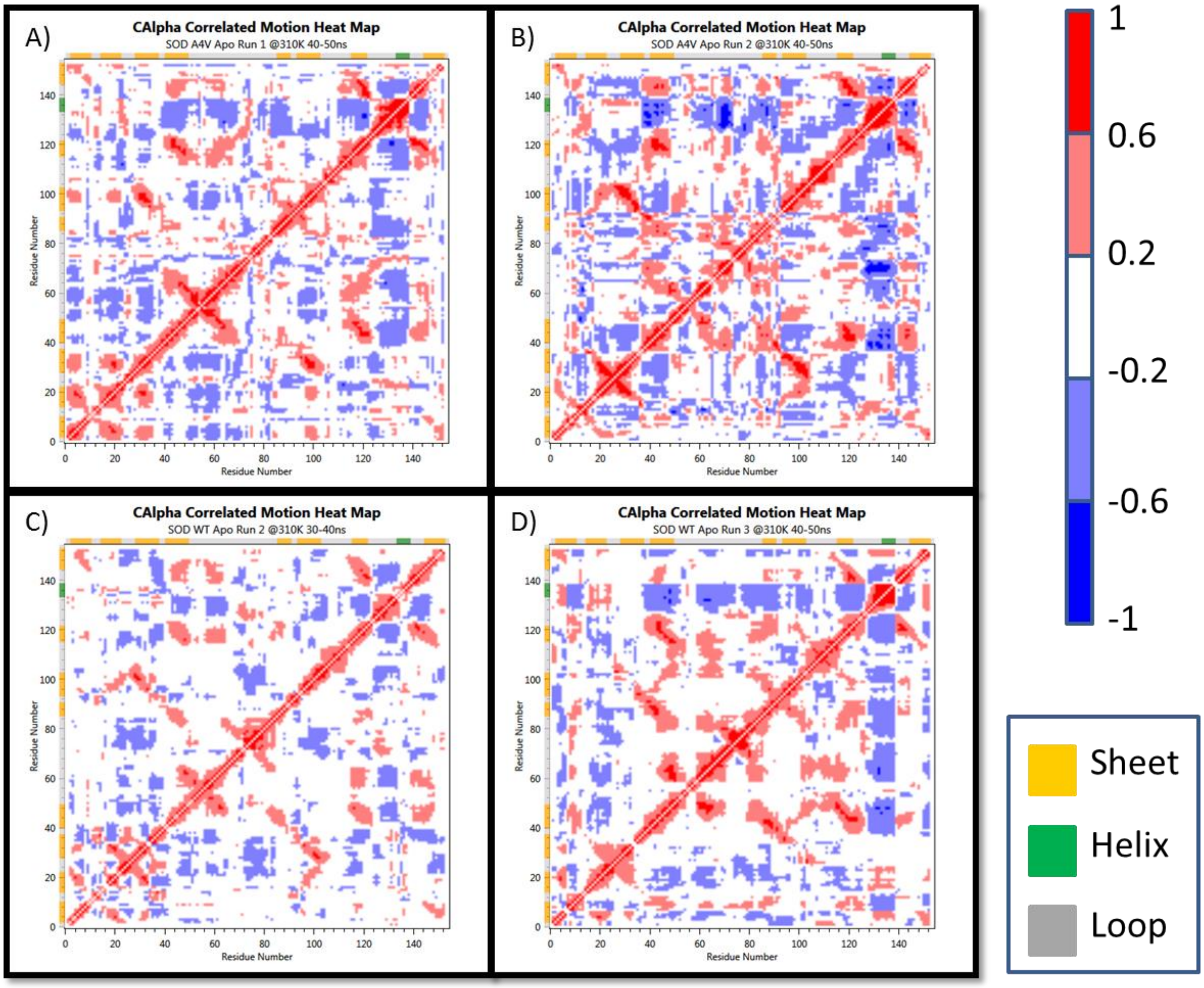


Figure 5: Alpha carbon correlated motion heat maps with 10ns window size. Correlation coefficient's color scale is shown on the right from red to blue. Secondary structure's legend is shown on the bottom right. Those four 10ns intervals were chosen from the results because they showed a similar trend. In all four plots, the electros static loop 124-142 correlates well with itself and showed a negative correlation with B4, B5, B7 and the Zn loop (67-79). A) Alpha carbon correlated motion heat map for- A4V run 1 between 40ns and 50ns. B) Alpha carbon correlated motion heat map for A4V run 2 between 40ns and 50ns. C) Alpha carbon correlated motion heat map for WT run 2 between 30ns and 40ns. D) Alpha carbon correlated motion heat map for A4V run 3 between 40ns and 50ns.

Strange *et al.* hypothesized that the large spatial and temporal fluctuations of the ES and Zn loops exposed beta 4 (residues 41-48), and beta 5 (residues 85-89), and allowed it to bind to other SOD1 monomers and caused aggregation [53]. The results from correlated motions match Strange *et al.*'s hypothesis extremely well. The correlated motions suggest that the ES loop move in opposite directions as beta 4, beta 5, beta 7, and the Zn loop. More evidence supporting Strange's hypothesis was revealed by examining the structures (Figure 6 - Figure 9). In A4V run1, A4V run2, WT run2, and WT run3, the ES loop tended to move in a downward fashion. The beta 4, beta 5, beta 7, and the Zn loop that had negative correlated motions with the ES loop tended to move upward, thus exposed the area between the front beta sheets (beta 1,2,3,6) and the back sheets (beta 4,5,7,8). The correlated motion calculation was able to pick out significant structural changes identified by other researchers. Thus, it potentially is an adequate tool for researchers to investigate for significant structural changes in protein simulations. The significant regions identified with correlated motion were explored further with the Leader Finder algorithm.

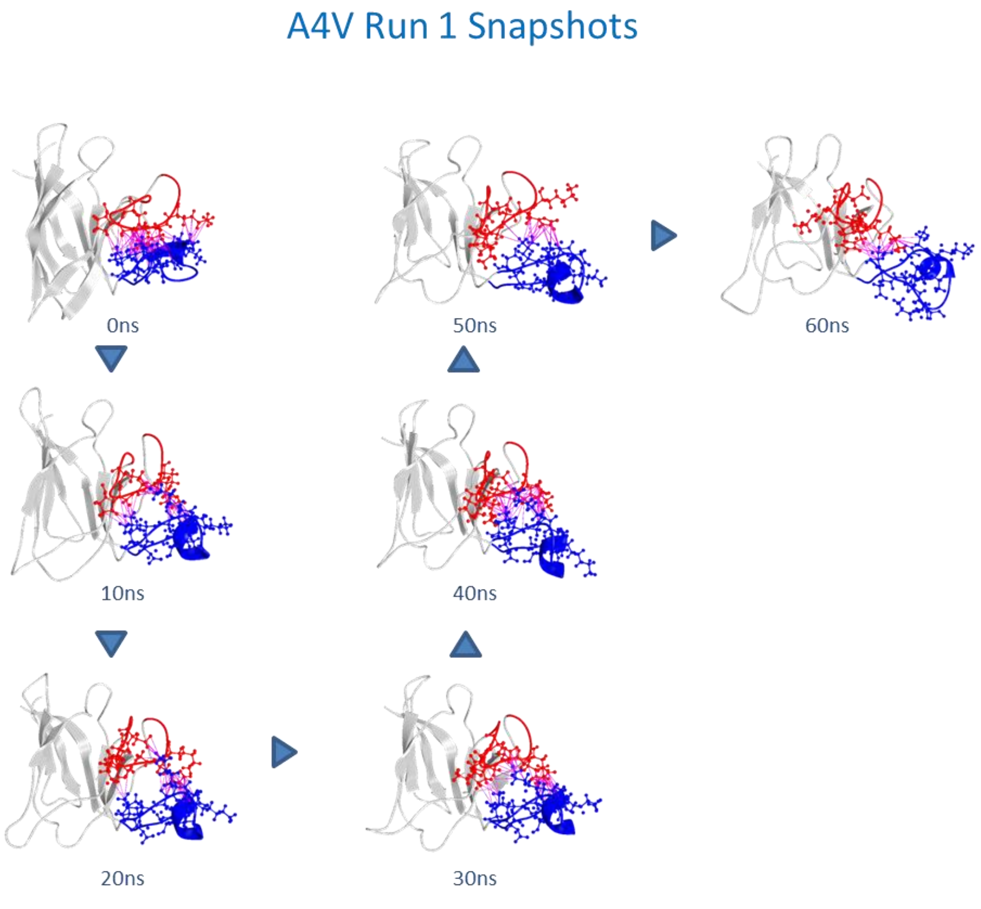


Figure 6: Snapshots of the A4V SOD1 structures throughout the A4V run 1 simulation. The Zn loop was colored red and the ES loop was colored blue. All atomic contacts that were in contact were shown in pink.

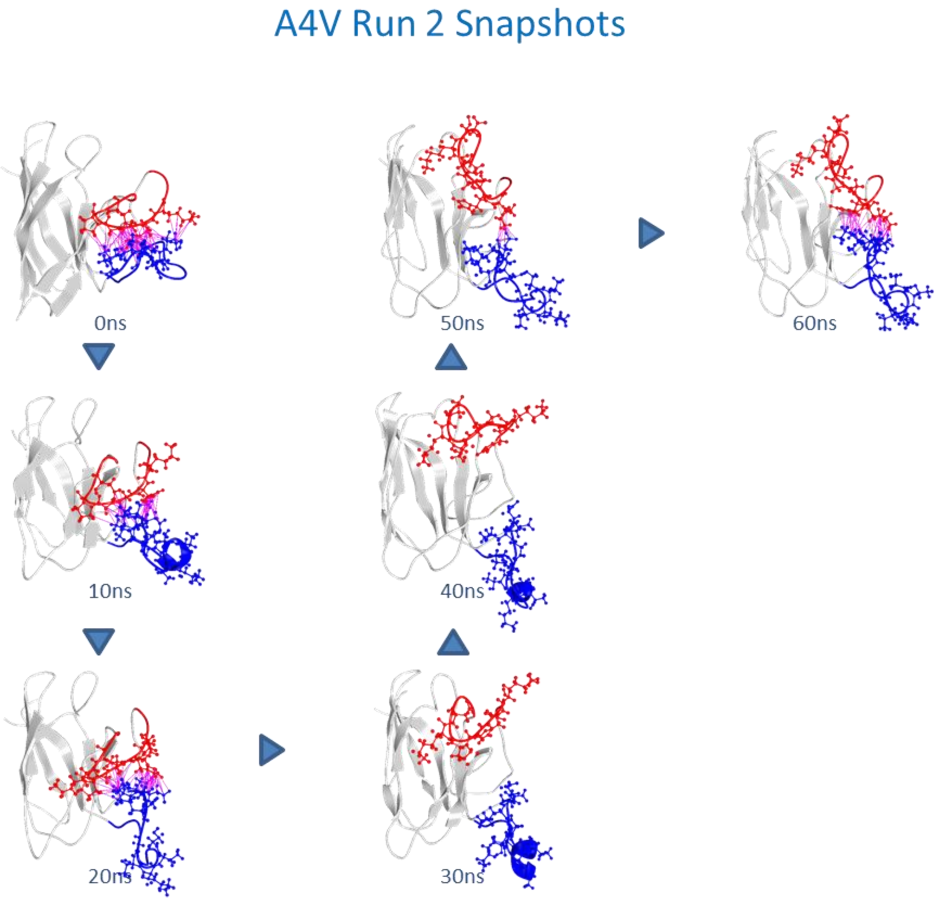


Figure 7: Snapshots of the A4V SOD1 structures throughout the A4V run 2 simulation. The Zn loop was colored red and the ES loop was colored blue. All atomic contacts that were in contact were shown in pink.

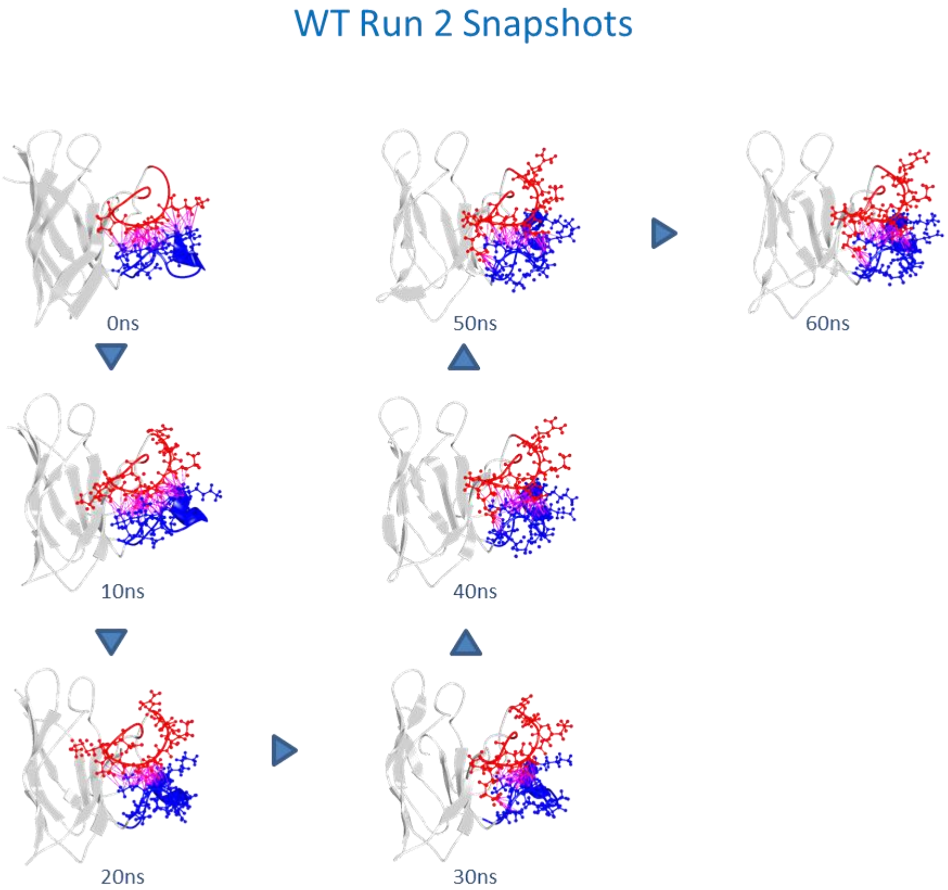


Figure 8: Snapshots of the WT SOD1 structures throughout the WT run 2 simulation. The Zn loop was colored red and the ES loop was colored blue. All atomic contacts that were in contact were shown in pink.

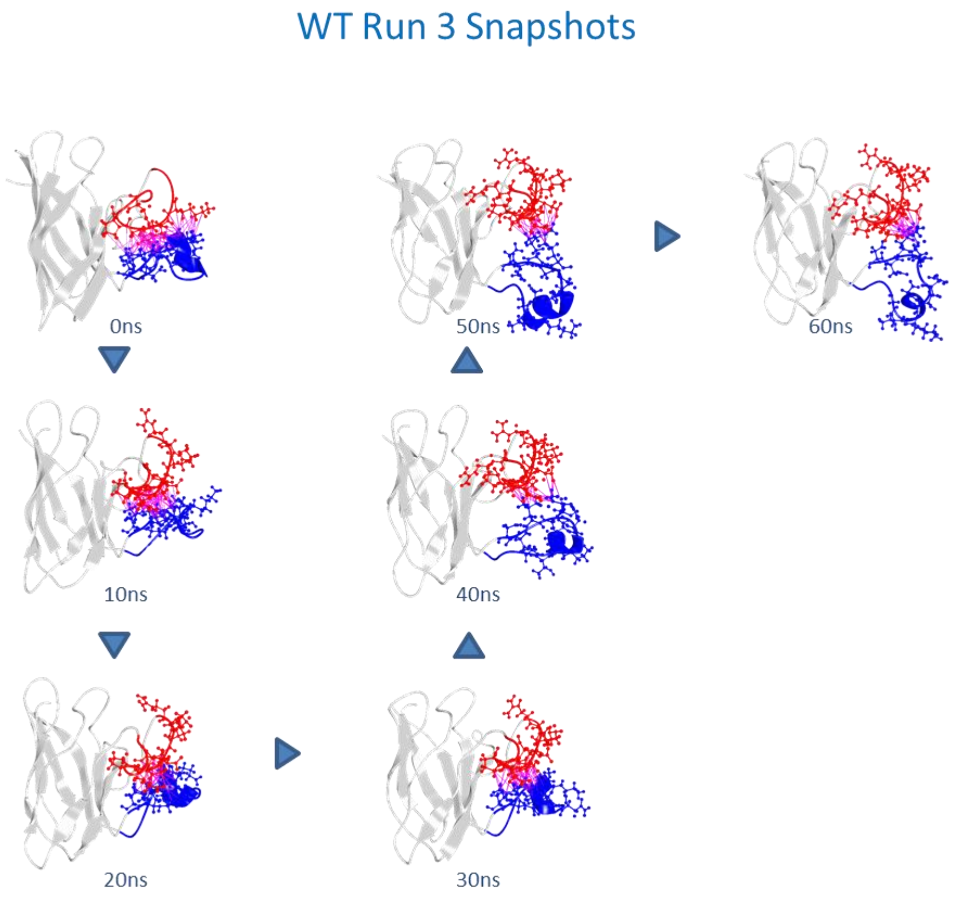


Figure 9: Snapshots of the WT SOD1 structures throughout the WT run 3 simulation. The Zn loop was colored red and the ES loop was colored blue. All atomic contacts that were in contact were shown in pink.

Correlated motion analysis from the WT and A4V both suggested that the ES loop moved in opposite directions as beta 4, 5, 7 and Zn. However, A4V has been known to be less stable than the WT and more toxic than the WT. The correlated heat maps were not enough to offer information about the difference between the WT and A4V proteins. To explore the differences between the WT and A4V, the numbers of in-contact atomic contacts were calculated for WT and A4V runs (Figure 10 - Figure 13). The numbers of in-contact atomic contacts in the A4V runs reached 0 at some point during the simulations and stayed slightly above 0 afterwards. For the WT runs, the number of in-contact atomic contacts decreased, but it never reached 0. The higher number of in-contact atomic contacts in the WT runs may explain the higher stability of the WT relative the A4V. The next step is to figure out the cause for the dramatic decrease in the numbers of in-contact atomic contacts between the Zn loop and the ES loop in the A4V runs. This should be an interesting topic for future research.

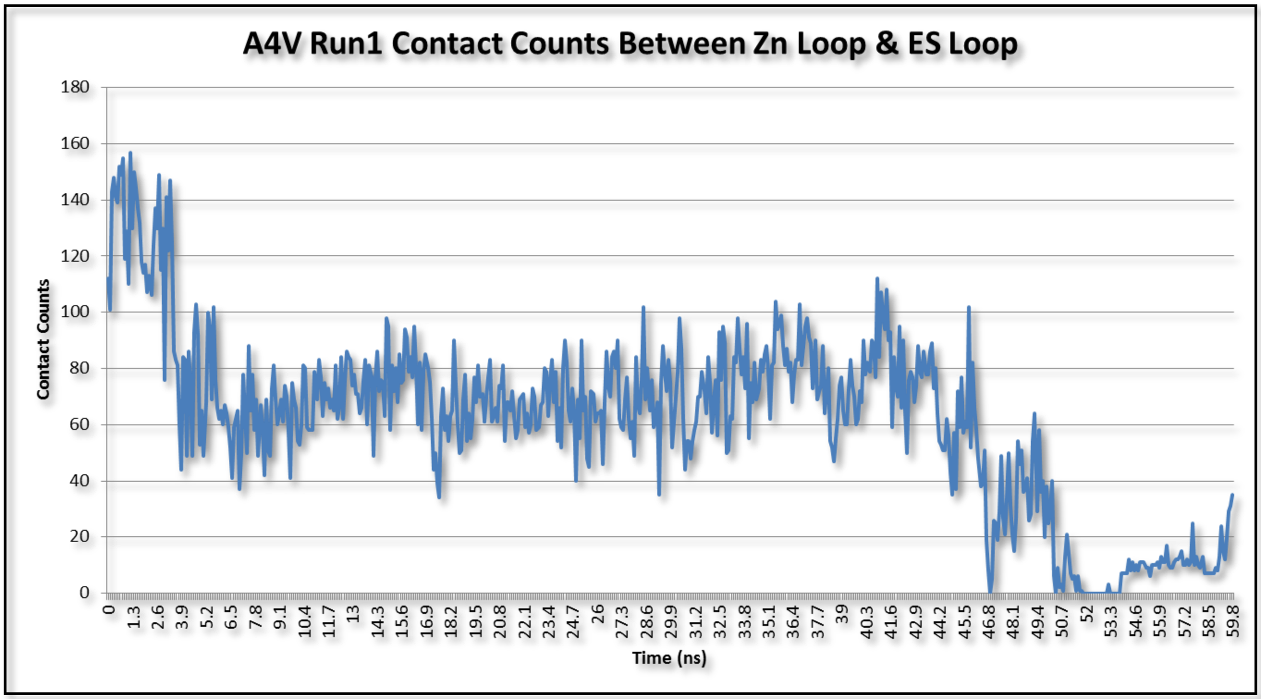


Figure 10: The numbers of in-contact atomic contacts between the Zn loop and the ES loop were plotted through time for A4V run 1. The number of contacts decreased to near 0 at the end of the simulation.

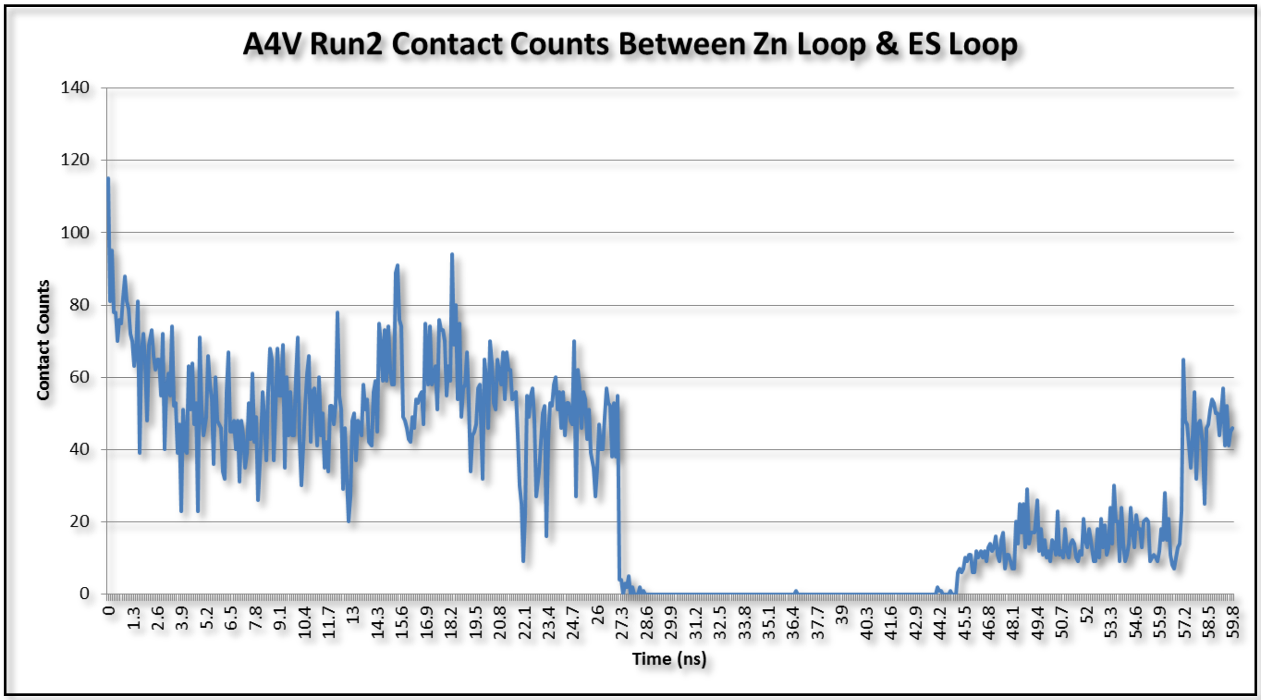


Figure 11: The numbers of in-contact atomic contacts between the Zn loop and the ES loop were plotted through time for A4V run 2. The number of contacts decreased to near 0 half way throughout the simulations and then formed some new non-native contacts after 44 ns.

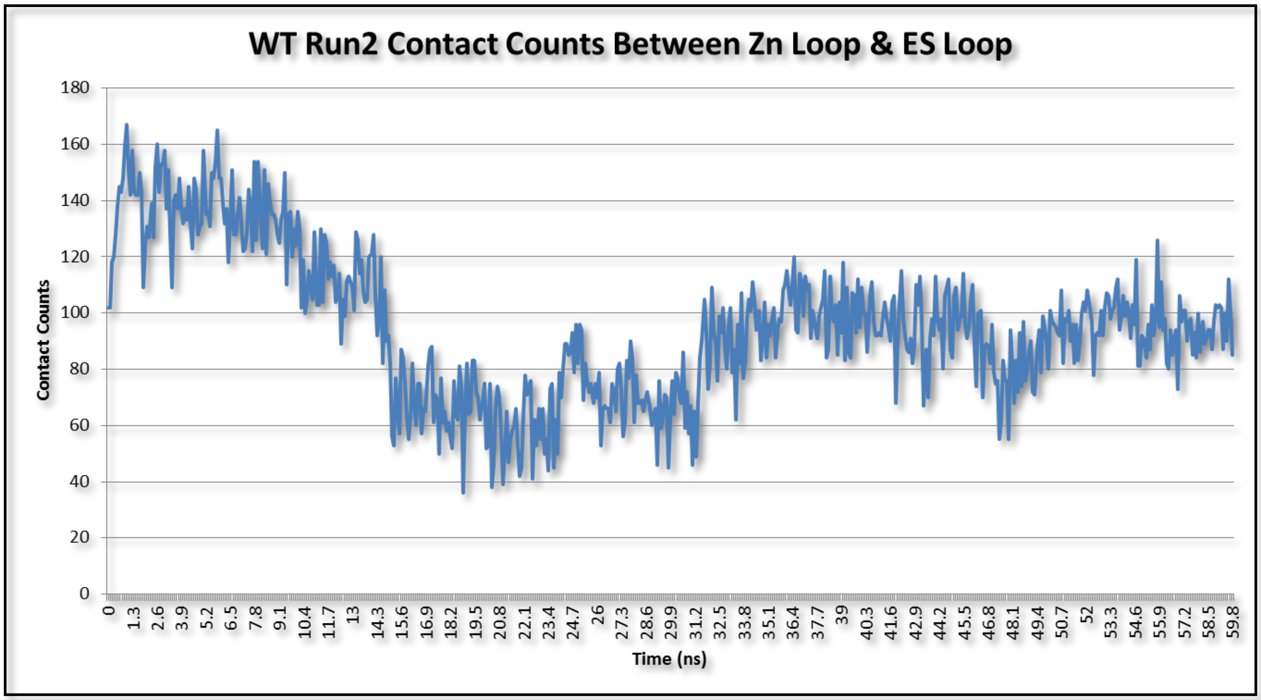


Figure 12: The numbers of in-contact atomic contacts between the Zn loop and the ES loop were plotted through time for WT run 2. The number of contacts stayed around 80 throughout the simulation unlike the A4V runs.

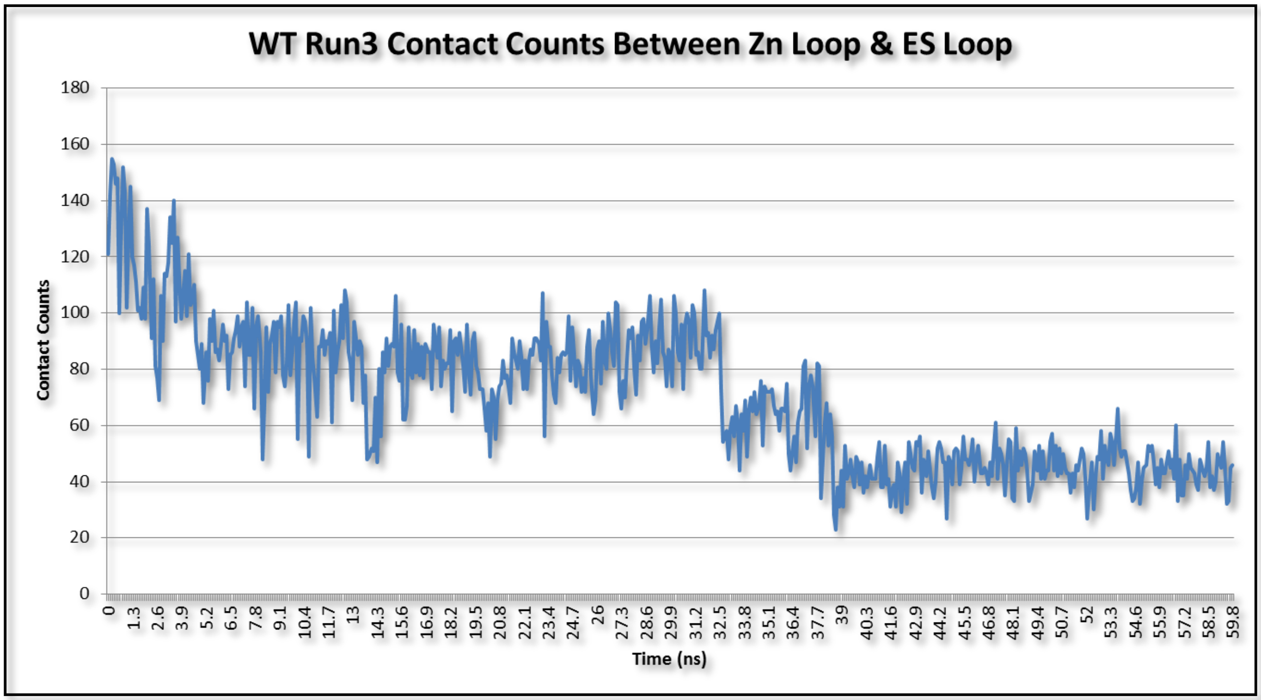


Figure 13: The numbers of in-contact atomic contacts between the Zn loop and the ES loop were plotted through time for WT run 3. The number of contacts stayed around 40 throughout the simulation. The numbers of in-contact atomic contacts were lower than the WT run 2's. However, the results were still much higher than A4V's results.

5.3 Leader Finder

The Leader Finder algorithm was applied to A4V run 1 between 40 and 50 ns, A4V run 2 between 40 and 50 ns, WT run 2 between 30 and 40 ns, and WT run 3 between 40 and 50 ns. The four time ranges were chosen based on the results from correlated motion (Section 5.3). Atomic contact leaders were identified using the Leader Finder algorithm, translated to residue-residue contact leaders using Algorithm 6, then translated to by-residue basis using Algorithm 7. The results were then summed, normalized, and visualized on to the SOD1 starting structure (Figure 14).

For A4V runs, most of the residues on the beta sheets had low scores. Low score was defined as lower than 0.6. High score was defined as higher than 0.6. The residues in the loop region such as the Zn loop and ES loop had high scores. The beta plug regions (37-43 and 89-95) also had higher scores than the beta sheet regions. The results suggested that the structural changes originated at the Zn loop, ES loop, and the beta plug regions. The structural changes in the beta sheets were probably caused by Zn loop, ES loop, or the beta regions. This finding matched well with Strange *et al.'s* proposed mechanisms of SOD aggregation [54]. Strange *et al.* hypothesized that the large spatial and temporal fluctuations of the ES and Zn loops exposed beta 4 (residues 41-48), and beta 5 (residues 85-89), and allowed it to bind to other SOD1 monomers and caused aggregation. For WT runs, the residues that had high scores were more spread out. More residues on the beta sheet regions had high scores. The results from the WT runs suggested that residues on the beta sheets were also responsible for the structural changes. The interactions between residues in the beta sheets may suggest that the proteins in the WT runs were held tighter together by the interactions between residues in the beta sheets than the proteins in the A4V runs. This could potentially explain the higher stability of WT SOD1 structure relative to the A4V SOD1 structure.

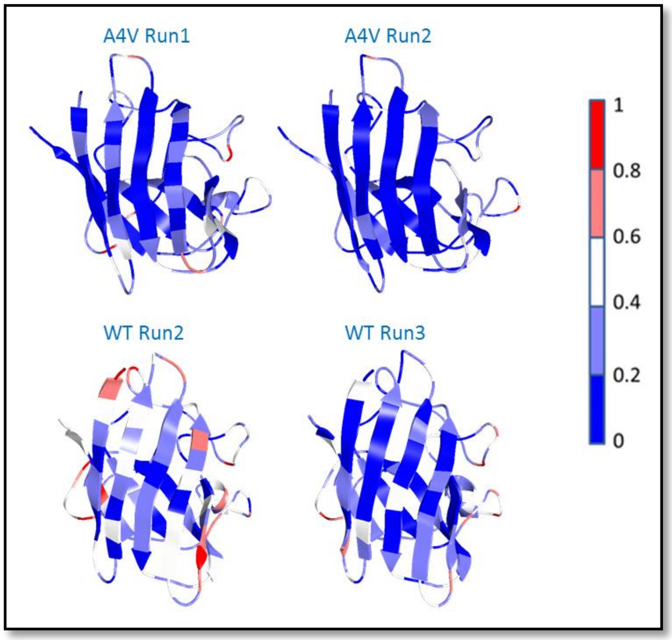


Figure 14 Leader scores for A4V run1, A4V run 2, WT run2, and WT run3. Residue with a high value meant that it was involved in many residue-residue contacts that were defined as leaders. For A4V runs, most of the residues on the beta sheets had low values. The loop region such as the Zn loop and ES loop had high scores. The beta plug regions also had

higher scores than the beta sheet regions. For WT runs, the residues that had high scores were more spread out. More residues on the beta sheet regions had high scores.

More high scoring residues were found in the WT runs than the A4V runs in general (Table 14 - Table 17). This suggested that in the WT runs, the structural changes were more even distributed throughout the protein than the A4V runs. When more residues were responsible for the structural changes, they might constrain each other and caused more stable structures. All the high scoring residues in the A4V runs were located in loop regions. Some of the high scoring residues in the WT runs were located in beta sheet regions, but most were still located in loop regions. Many glycine residues were identified as influential residues. It might have happened because glycine is likely to exist in loop structure. It might have happened because there were simply more glycine than other amino acids in the SOD1 structure (Table 18). There is no way to be certain, unless the Leader Finder algorithm is applied to different protein simulations, which will be left' for future research.

Table 14: Residues with leader scores high than 0.6 for A4V run 1

A4V Run 1. 40 ns – 50 ns			
Residue Number	**Residue Name**	**Score (Normalized)**	**DSSP At 0ns**
26	**ASN**	**0.602**	**Loop**
56	**GLY**	**0.97**	**Loop**
60	**ALA**	**0.63**	**Loop**
61	**GLY**	**1**	**Loop**
72	**GLY**	**0.85**	**Loop**
73	**GLY**	**0.80**	**Loop**
92	**ASP**	**0.6**	**Loop**
93	**GLY**	**0.79**	**Loop**
139	**ASN**	**0.66**	**Loop**

Table 15: Residues with leader scores high than 0.6 for A4V run 2

A4V Run 2. 40 ns – 50 ns			
Residue Number	**Residue Name**	**Score (Normalized)**	**DSSP At 0ns**
25	**SER**	**0.65**	**Loop**
130	**GLY**	**1**	**Loop**

Table 16: Residues with leader scores high than 0.6 for WT run 2.

WT Run 2. 30 ns – 40 ns			
Residue Number	**Residue Name**	**Score (Normalized)**	**DSSP At 0ns**
1	**ALA**	**0.80**	**Loop**

2	**THR**	**0.62**	**Sheet**
26	**ASN**	**0.77**	**Loop**
27	**GLY**	**0.66**	**Loop**
51	**GLY**	**0.85**	**Loop**
52	**ASP**	**0.65**	**Loop**
56	**GLY**	**0.86**	**Loop**
57	**CYS**	**0.62**	**Loop**
58	**THR**	**0.62**	**Loop**
72	**GLY**	**0.90**	**Loop**
88	**THR**	**0.64**	**Sheet**
89	**ALA**	**0.95**	**Sheet**
90	**ASP**	**0.73**	**Loop**
100	**GLU**	**0.65**	**Sheet**
107	**SER**	**0.74**	**Loop**
108	**GLY**	**1**	**Loop**
123	**ALA**	**0.65**	**Loop**
124	**ASP**	**0.60**	**Loop**

Table 17: Residues with leader scores high than 0.6 for WT run 3.

WT Run 3. 40 ns – 50 ns			
Residue Number	**Residue Name**	**Score (Normalized)**	**DSSP At 0ns**
10	**GLY**	**0.67**	**Sheet**
55	**ALA**	**0.88**	**Loop**
56	**GLY**	**0.76**	**Loop**
58	**THR**	**0.66**	**Loop**
72	**GLY**	**0.63**	**Loop**
73	**GLY**	**0.83**	**Loop**
76	**ASP**	**0.76**	**Loop**
90	**ASP**	**0.77**	**Loop**
124	**ASP**	**0.72**	**Loop**
127	**GLY**	**0.65**	**Loop**
130	**GLY**	**1**	**Loop**

Table 18: The three most occurring amino acids in SOD1 structure

Amino Acid	**Count**
GLY	**25**
VAL	**15**
ASP	**11**

The Leader Finder algorithm extracted the causal relationships between atomic contacts through correlation calculation. Correlation calculations were carried out with lag 1 to 10. However, there was no mathematical explanation for the choice of lags used. Lag of 10 was chosen as a high cutoff because it allowed for a fast calculation. Most lag correlation calculation algorithms suggested a lag up to N/2 [46, 47]. Ideally, a lag of 10 can always calculate the maximal correlation between samples with 100 time points. However, it turned out to be the opposite.

Figure 15 shows that the maximal correlations between samples with 100 time points did not always fall within a lag of 1 to 10. This finding suggests that the results may not be the same if a lag of 1 to 50 was used. Leader Finder with longer lag may increase the reliability of the results and should be practiced in the future.

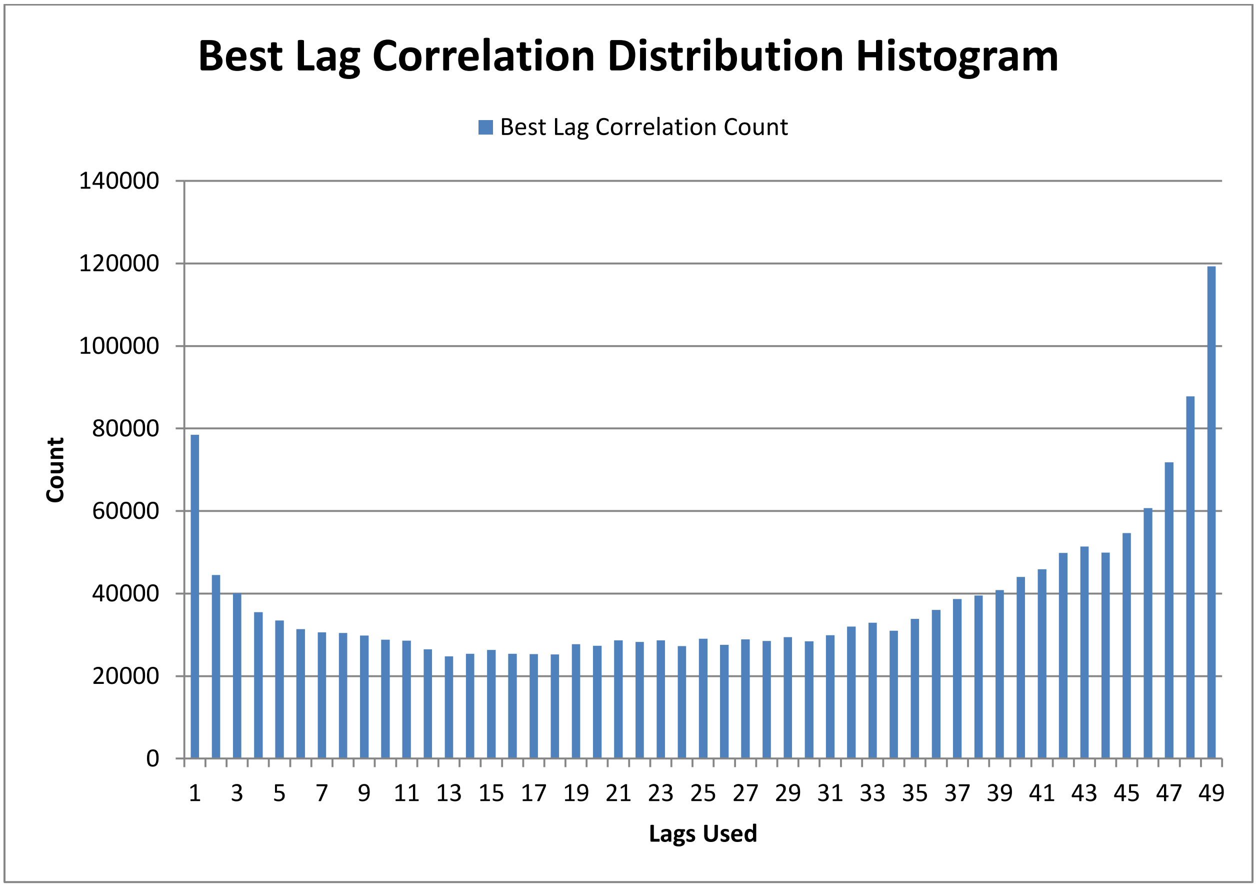
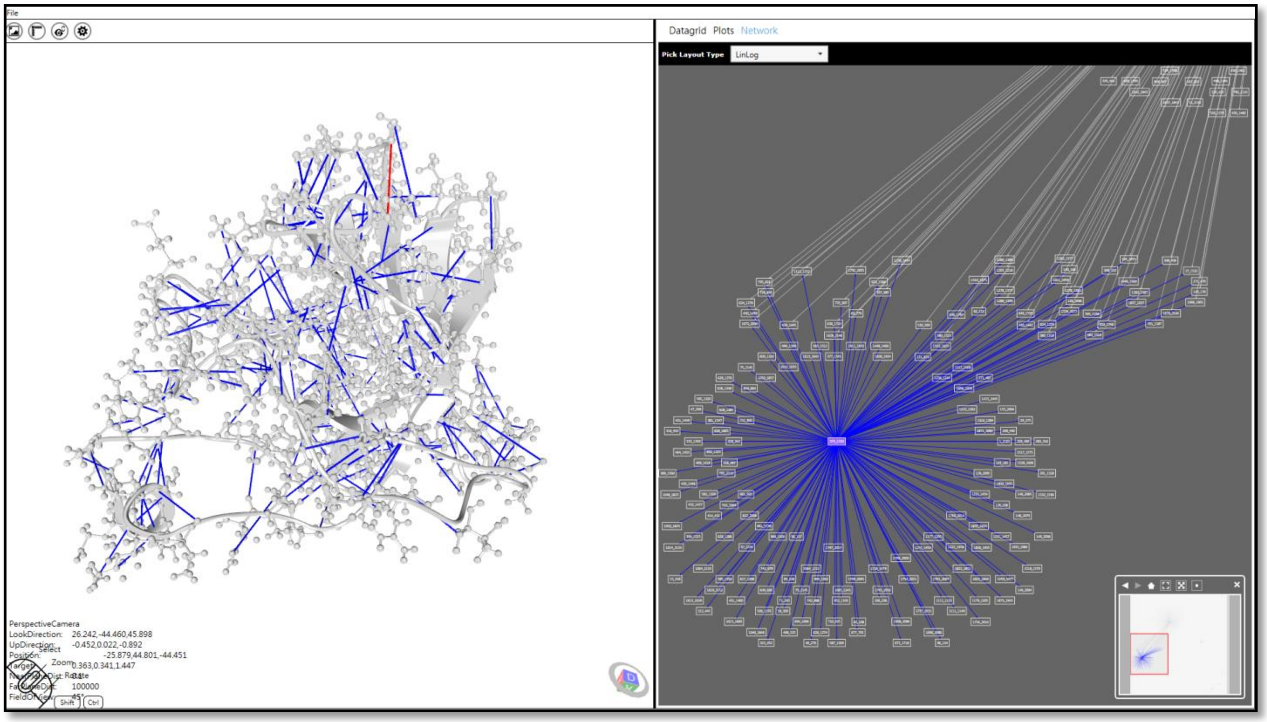


Figure 15: The correlations between every possible pair of atomic contacts from A4V run 1 were calculated with a lag of 1 to 49. The number of counts in each lag bin indicated the number of maximal correlations that were found with that lag. The results suggested that a lag of 1-10 was not sufficient. Lag 1-10 only covered a small portion of the overall results.

The Protein Dashboard was enhanced with the ability of display results from the Leader Finder algorithm. The interactive nature of Protein Dashboard allows researchers to quickly visualize their results in 3D (Figure 16). The Protein Dashboard had a network tab added to its graphic user interface. Under the network tab, users could visualize any causal network in 2D. The visualized 2D network was also interactive. As users hover over a node in the network, all the outgoing edges are shown in blue, and all the incoming edges would be shown in red. As users hover over an edge, the weight associated with the edge is shown as a tool tip. As users click on any node in the network, the clicked atomic contact becomes visible on the 3D structure in red, and all its descendants re shown on the 3D structure in blue. This new feature in the Protein Dashboard allows for quick visualization of the causal network found and should be useful for other researchers.



Pick Layout Type LinLog

```
PerspectiveCamera
LookDirection:   26.242,-44.460,45.898
UpDirection:     -0.452,0.022,-0.892
Position:              -25.879,44.801,-44.451
Target:          0.363,0.341,1.447
NearPlaneDist:   0.1
FarPlaneDist:    100000
FieldOfView:     45°
```

Figure 16: A snapshot of the Protein Dashboard. The network tab on the right hand side showed the casual relationships between atomic contacts discovered with the Leader Finder algorithm. A leader contact was clicked on the graph viewer under the network tab, the 3D render responded by displaying the leader contact in red and all its descendants in blue.

6. DISCUSSION

Atomic contacts determine protein structure and their changes over time reflect protein dynamics. The dynamic nature of protein structures is often tied to its function, thus by understanding the relationship between every atomic contact, researchers can obtain full the story of protein dynamics. In this project, three methods were attempted to extract information about atomic contacts in SOD1 protein simulations. The first method was to extract a full Boolean network from the residue-residue contacts in SOD1 simulations using the ReBMM algorithm [43]. However, ReBmm was not capable of modeling the contact network as a Boolean work with high accuracy suggesting that the contact network is likely to be time-varying. A time-varying network is more commonly known as adaptive network, where the topology of the network changes as the state of the nodes changes [52]. Unfortunately there is not any viable solution at this point. Using atomic contacts and smaller time windows may offer a good chance to extract a decent Boolean network from simulations. It may be a good research for the future.

The second method was to examine the motions in different regions of the protein directly by calculating the correlated motion between each pair of residues. Correlated motion successfully identified significant structural changes that matched extremely well with Strange *et al.*'s findings [54]. The correlated motion revealed that the ES loop moved in opposite direction as the beta 4, beta 5, beta 7, and the Zn loop. By further examining the structures through simulations, the ES loop turned out to always move downward and the beta 4, beta 5, beta 7, and the Zn loop always moved in the opposite direction. As a result, the area between the front sheets (beta 1, 2, 3, 6) and the back sheets (beta 4, 5, 7, 8) became exposed and may be the reason of SOD aggregation. The higher number of in-contact atomic contacts in the WT runs relative to the A4V runs may explain the higher stability in the WT than the A4V. The ability to automatically capture significant structural changes in protein simulation proved that the correlated motion can be a useful tool to narrow down the search windows for finding the relationships between atomic contacts.

The time intervals identified as significant in the correlated motion section were further examined with the Leader Finder algorithm. The Leader finder algorithm does not extract a

Boolean network. It extracts the causal network. Residues in the Zn loop and the ES loop tend to have high scores in the A4V runs. This suggests that the structural changes in the A4V simulations originated from the Zn loop and the ES loop. In the WT runs, the residues with high scores were more spread out. More residues were responsible for the structural changes may explain for the higher stability of the WT relative to the A4V because they may constraint each other. Leader Finder algorithm found results that matches the results from the correlated motion and it also gave information about which residues may be the cause of it. The Leader Finder algorithm has been shown to be useful for studying atomic contacts and its results are consistent with those of the correlated motion analysis and it also provided further information about the possible causes. Also, the Protein Dashboard was made to visualize the causal network in an interactive fashion, thus it should be a valuable tool researchers who wish to study atomic contacts in simulations.

7. REFERENCES

[1] T. Schmidlin, B. K. Kennedy, and V. Daggett, "Structural changes to monomeric CuZn superoxide dismutase caused by the familial amyotrophic lateral sclerosis-associated mutation A4V.," *Biophys J,* vol. 97, pp. 1709-18, Sep 2009.

[2] W. Chen, M. W. van der Kamp, and V. Daggett, "Diverse effects on the native β-sheet of the human prion protein due to disease-associated mutations.," *Biochemistry,* vol. 49, pp. 9874-81, Nov 2010.

[3] S. Calhoun and V. Daggett, "Structural Effects of the L145Q, V157F, and R282W Cancer-Associated Mutations in the p53 DNA-Binding Core Domain.," *Biochemistry,* May 2011.

[4] P. C. Anderson and V. Daggett, "The R46Q, R131Q and R154H polymorphs of human DNA glycosylase/beta-lyase hOgg1 severely distort the active site and DNA recognition site but do not cause unfolding.," *J Am Chem Soc,* vol. 131, pp. 9506-15, Jul 2009.

[5] D. A. Beck and V. Daggett, "Methods for molecular dynamics simulations of protein folding/unfolding in solution.," *Methods,* vol. 34, pp. 112-20, Sep 2004.

[6] A. M. Simms, R. D. Toofanny, C. Kehl, N. C. Benson, and V. Daggett, "Dynameomics: design of a computational lab workflow and scientific data repository for protein simulations.," *Protein Eng Des Sel,* vol. 21, pp. 369-77, Jun 2008.

[7] C. Kehl, A. M. Simms, R. D. Toofanny, and V. Daggett, "Dynameomics: a multi-dimensional analysis-optimized database for dynamic protein data.," *Protein Eng Des Sel,* vol. 21, pp. 379-86, Jun 2008.

[8] M. W. van der Kamp, R. D. Schaeffer, A. L. Jonsson, A. D. Scouras, A. M. Simms, R. D. Toofanny, *et al..*, "Dynameomics: a comprehensive database of protein dynamics.," *Structure,* vol. 18, pp. 423-35, Mar 2010.

[9] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig*, et al.*., " The Protein Data Bank," *Nucleic Acids Research,* vol. 28, pp. 235-242, 2000.

[10] A. A. Yee, A. Savchenko, A. Ignachenko, J. Lukin, X. Xu, T. Skarina*, et al.*., "NMR and X-ray crystallography, complementary tools in structural proteomics of small proteins.," *J Am Chem Soc,* vol. 127, pp. 16512-7, Nov 2005.

[11] A. M. Davis, S. J. Teague, and G. J. Kleywegt, "Application and limitations of X-ray crystallographic data in structure-based ligand and drug design.," *Angew Chem Int Ed Engl,* vol. 42, pp. 2718-36, Jun 2003.

[12] A. H. Kwan, M. Mobli, P. R. Gooley, G. F. King, and J. P. Mackay, "Macromolecular NMR spectroscopy for the non-spectroscopist.," *FEBS J,* vol. 278, pp. 687-703, Mar 2011.

[13] A. R. Fersht and V. Daggett, "Protein folding and unfolding at atomic resolution.," *Cell,* vol. 108, pp. 573-82, Feb 2002.

[14] V. Daggett and M. Levitt, "Protein unfolding pathways explored through molecular dynamics simulations.," *J Mol Biol,* vol. 232, pp. 600-19, Jul 1993.

[15] J. Mintseris and Z. Weng, "Atomic contact vectors in protein-protein recognition," *Proteins,* vol. 53, pp. 629-39, Nov 2003.

[16] T. Schmidlin, B. K. Kennedy, and V. Daggett, "Structural changes to monomeric CuZn superoxide dismutase caused by the familial amyotrophic lateral sclerosis-associated mutation A4V," *Biophys J,* vol. 97, pp. 1709-18, Sep 2009.

[17] T. Schmidlin, K. Ploeger, A. L. Jonsson, and V. Daggett, "Early steps in thermal unfolding of superoxide dismutase 1 are similar to the conformational changes associated with the ALS-associated A4V mutation," *Protein Eng Des Sel,* vol. 26, pp. 503-13, Aug 2013.

[18] M. Vassura, L. Margara, P. Di Lena, F. Medri, P. Fariselli, and R. Casadio, "Reconstruction of 3D structures from protein contact maps," *IEEE/ACM Trans Comput Biol Bioinform,* vol. 5, pp. 357-67, 2008 Jul-Sep 2008.

[19] S. H. Vehlow C, Winkelmann M, Duarte JM, Petzold L, Dinse J, Lappe M., "CMView: Interactive contact map visualization and analysis.," *Bioinformatics,* 2011.

[20] D. Kozma, I. Simon, and G. E. Tusnády, "CMWeb: an interactive on-line tool for analysing residue-residue contacts and contact prediction methods," *Nucleic Acids Res,* vol. 40, pp. W329-33, Jul 2012.

[21] L. P. Rowland and N. A. Shneider, "Amyotrophic lateral sclerosis," *N Engl J Med,* vol. 344, pp. 1688-700, May 2001.

[22] J. S. Valentine, P. A. Doucette, and S. Zittin Potter, "Copper-zinc superoxide dismutase and amyotrophic lateral sclerosis," *Annu Rev Biochem,* vol. 74, pp. 563-93, 2005.

[23] R. Rakhit, J. P. Crow, J. R. Lepock, L. H. Kondejewski, N. R. Cashman, and A. Chakrabartty, "Monomeric Cu,Zn-superoxide dismutase is a common misfolding intermediate in the oxidation models of sporadic and familial amyotrophic lateral sclerosis," *J Biol Chem,* vol. 279, pp. 15499-504, Apr 2004.

[24] S. D. Khare, M. Caplow, and N. V. Dokholyan, "The rate and equilibrium constants for a multistep reaction sequence for the aggregation of superoxide dismutase in amyotrophic lateral sclerosis," *Proc Natl Acad Sci U S A,* vol. 101, pp. 15094-9, Oct 2004.

[25] L. Banci, I. Bertini, F. Cantini, M. D'Onofrio, and M. S. Viezzoli, "Structure and dynamics of copper-free SOD: The protein before binding copper," *Protein Sci,* vol. 11, pp. 2479-92, Oct 2002.

[26] L. Banci, M. Benedetto, I. Bertini, R. Del Conte, M. Piccioli, and M. S. Viezzoli, "Solution structure of reduced monomeric Q133M2 copper, zinc superoxide dismutase (SOD). Why is SOD a dimeric enzyme?," *Biochemistry,* vol. 37, pp. 11780-91, Aug 1998.

[27] R. M. Cardoso, M. M. Thayer, M. DiDonato, T. P. Lo, C. K. Bruns, E. D. Getzoff*, et al.*., "Insights into Lou Gehrig's disease from the structure and instability of the A4V mutant of human Cu,Zn superoxide dismutase," *J Mol Biol,* vol. 324, pp. 247-56, Nov 2002.

[28] S. D. Khare and N. V. Dokholyan, "Common dynamical signatures of familial amyotrophic lateral sclerosis-associated structurally diverse Cu, Zn superoxide dismutase mutants," *Proc Natl Acad Sci U S A*, vol. 103, pp. 3147-52, Feb 2006.

[29] M. A. Hough, J. G. Grossmann, S. V. Antonyuk, R. W. Strange, P. A. Doucette, J. A. Rodriguez, *et al.*., "Dimer destabilization in superoxide dismutase may result in disease-causing properties: structures of motor neuron disease mutants," *Proc Natl Acad Sci U S A*, vol. 101, pp. 5976-81, Apr 2004.

[30] D. A. C. Beck, D. O. V. Alonso, M. E. McCully, and V. Daggett, "*in lucem Molecular Mechanics (ilmm)*," ed: University of Washington, Settle, WA., 2000-2013.

[31] M. Levitt, M. Hirshberg, R. Sharon, and V. Daggett, "Potential energy function and parameters for simulations of the molecular dynamics of proteins and nucleic acids in solution," *Computer Physics Communications*, vol. 91, pp. 215-231, 1995.

[32] M. Levitt, M. Hirshberg, R. Sharon, K. E. Laidig, and V. Daggett, "Calibration and Testing of a Water Model for Simulation of the Molecular Dynamics of Proteins and Nucleic Acids in Solution," *The Journal of Physical Chemistry B*, vol. 101, pp. 5051-5061, 1997/06/01 1997.

[33] D. A. C. Beck, R. S. Armen, and V. Daggett, "Cutoff Size Need Not Strongly Influence Molecular Dynamics Results for Solvated Polypeptides†," *Biochemistry*, vol. 44, pp. 609-616, 2005/01/01 2004.

[34] R. S. Armen, B. M. Bernard, R. Day, D. O. V. Alonso, and V. Daggett, "Characterization of a possible amyloidogenic precursor in glutamine-repeat neurodegenerative diseases," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 13433-13438, 2005.

[35] A. L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509-12, Oct 1999.

[36] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651-4, Oct 2000.

[37] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, pp. 41-2, May 2001.

[38] S. Martin, Z. Zhang, A. Martino, and J. L. Faulon, "Boolean dynamics of genetic regulatory networks inferred from microarray time series data," *Bioinformatics*, vol. 23, pp. 866-74, Apr 2007.

[39] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, pp. 261-74, Feb 2002.

[40] S. Liang, S. Fuhrman, and R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," *Pac Symp Biocomput*, pp. 18-29, 1998.

[41] H. Lähdesmäki, I. Shmulevich, O. Yli-Harja, and J. Astola, "Inference of Genetic Regulatory Networks via Best-Fit Extensions," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., ed: Springer US, 2006, pp. 259-278.

[42] M. Maucher, B. Kracher, M. Kühl, and H. A. Kestler, "Inferring Boolean network structure via correlation," *Bioinformatics*, vol. 27, pp. 1529-36, Jun 2011.

[43] M. Saeed, M. Ijaz, K. Javed, and H. A. Babri, "Reverse engineering Boolean networks: from Bernoulli mixture models to rule based systems," *PLoS One*, vol. 7, p. e51006, 2012.

[44] T. E. Ideker, V. Thorsson, and R. M. Karp, "Discovery of regulatory interactions through perturbation: inference and experimental design," *Pac Symp Biocomput*, pp. 305-16, 2000.

[45] T. Akutsu, S. Miyano, and S. Kuhara, "Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function," *J Comput Biol*, vol. 7, pp. 331-43, 2000.

[46] Y. Sakurai, S. Papadimitriou, and C. Faloutsos, "BRAID: stream mining through group lag correlations," presented at the Proceedings of the 2005 ACM SIGMOD international conference on Management of data, Baltimore, Maryland, 2005.

[47] H. Kitagawa, Y. Ishikawa, Q. Li, C. Watanabe, D. Wu, Y. Ke, *et al.*., "Detecting Leaders from Correlated Time Series," in *Database Systems for Advanced Applications*. vol. 5981, ed: Springer Berlin Heidelberg, 2010, pp. 352-367.

[48] T. Zhang, D. Yue, Y. Gu, and G. Yu, "Boolean representation based data-adaptive correlation analysis over time series streams," presented at the Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, Lisbon, Portugal, 2007.

[49] Y. Zhu and D. Shasha, "StatStream: statistical monitoring of thousands of data streams in real time," presented at the Proceedings of the 28th international conference on Very Large Data Bases, Hong Kong, China, 2002.

[50] M. Levitt, C. Sander, and P. S. Stern, "Protein normal-mode dynamics: trypsin inhibitor, crambin, ribonuclease and lysozyme," *J Mol Biol,* vol. 181, pp. 423-47, Feb 1985.

[51] V. Daggett and M. Levitt, "Protein unfolding pathways explored through molecular dynamics simulations," *J Mol Biol,* vol. 232, pp. 600-19, Jul 1993.

[52] T. Gross and B. Blasius, "Adaptive coevolutionary networks: a review," *J R Soc Interface,* vol. 5, pp. 259-71, Mar 2008.

[53] R. W. Strange, S. Antonyuk, M. A. Hough, P. A. Doucette, J. A. Rodriguez, P. J. Hart, *et al.*., "The structure of holo and metal-deficient wild-type human Cu, Zn superoxide dismutase and its relevance to familial amyotrophic lateral sclerosis," *J Mol Biol,* vol. 328, pp. 877-91, May 2003.

[54] R. W. Strange, C. W. Yong, W. Smith, and S. S. Hasnain, "Molecular dynamics using atomic-resolution structure reveal structural fluctuations that may lead to polymerization of human Cu-Zn superoxide dismutase," *Proc Natl Acad Sci U S A,* vol. 104, pp. 10040-4, Jun 2007.