Feature Selection In Biological Models

Mohammad Muneeb Sultan, *Stanford University*

Molecular dynamics(MD) of biophysical systems suffer from a lack of systematic way in understanding the resulting trajectories.In this project, I show that the combination of supervised and unsupervised learning on MD data allows us to gain both holisitic and atomistic insight about the dynamics of the system.

Introduction

One of the most difficult problems with running classical Newtonian simulations of large biophysical systems is the lack of a systematic way to understand the resulting trajectories. Most common approaches include viewing the trajectories or plotting some key parameter as a function of time. This methodology works because in most biological systems, only a few things change when going from inactive to active regimes.
However, the question arises, how do we know what features/degrees of freedom are relevant within in the system and more importantly how can we objectively ensure that we do not miss anything? This problem is a variant of feature selection and decision boundary identification within machine learning literature and has been significantly studied in the past under the umbrella of supervised machine learning. In this project, I intend to show that the Gini Importance used in building decision boundaries for tree classifiers can also be used to efficiently find important features/degrees of freedom in biological simulations.

Methods

The methods employed in this paper can be summarized as follows.

- Generation of the data(running the MD engine on the protein to generate the trajectories)
- Application of Unsupervised K-means Learning algorithm to generate protein clusters from the trajectories.
- Vectorized Representation of training and test conformations/poses of the protein in desired feature space.
- Supervised learning on the features for importance ranking.

Details of each of these steps will be presented in the following subsections. A key point here is that we are using the rmsd for the unsupervised learning algorithm but are using dihedrals(as the features) for the supervised portion. The idea being while rmsd can cluster the data efficiently, it lacks the ability to provide atomistic detail and while dihedrals can provide atomistic detail but are not able to cluster the data properly. This would in theory allow us to explain very high dimensional clusters via only a few key degrees of freedoms. A variety of supervised learning

algorithms were employed although ultimately, the Decision Tree Classifier was unmatched in terms of accuracy and relative ease of understanding the resulting model. Only the results of the decision tree will be given in this project. All unsupervised learning algorithms were employed using the MSMBuilder software[1] and supervised learning algorithms were used from the excellent Scikit Learn Python library[2].

Newtonian Dynamics

Newtonian dynamics on alanine dipeptide (figure 1) were run using the Amber99-sbildn potential energy function. 500 individual trajectories with an average length of 20 nanoseconds were generated. For the second test case,the Ubiquitin enzyme, longer trajectories(100-150 μ seconds) were run but with the same potential energy function.The data for the second protein enzyme was obtained from a colleague in the Pande lab.

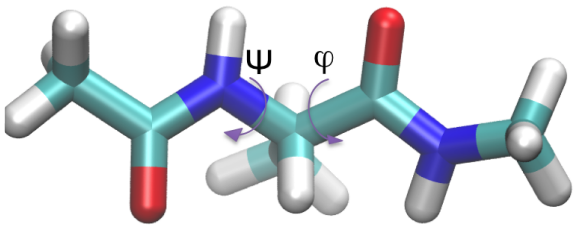

Figure 1: *The Alanine dipeptide molecule with its two main degrees of freedom ϕ & ψ.*

Unsupervised Learning

Supervised machine learning requires that we have labeled training data that can be used to train a predictive model. But, the concept of a label in a biological system is difficult to define. In order to generate these labels, a trick was employed. State models for the protein simulation data were built using an unsupervised k-means (k=4) learning algorithm. These models were built using the root mean squared deviation (RMSD) of the heavy atoms for each of the 500000 poses/conformations adopted by the alanine peptide.As shown in figure 2, the unsupervised algorithm created divided the data along natural boundaries.

Similar approach was used for the Ubiquitin enzyme but with k=3.The results from the unsupervised learning were again obtained from my colleague Gert Kiss.

Feature Vector Generation

After this initial step, each alanine conformation was expressed as a vector using 10 features. The first two were the actual protein backbone dihedrals, which we would like to be identified as being important, while the last 8 were random Gaussian noise, which should ideally be ignored(Table 1 contains 2 training examples without the 8 normally distributed noise columns). The aim of this was to find out if an algorithm could find the right features that can explain the data while ignoring random noise. For the Ubiquitin test case, each conformation was expressed using the backbone dihedral that totaled to 140 dihedrals (i.e. 140 features). Again, only a few dihedrals are likely to be important in this dataset.

Table 1: *Example of Vectorized Representation of Conformations for Alanine dipeptide*

Index	ϕ	ψ	Label
1	-85.7	50.3	0
2	-125.1	-88.7	2

Supervised Learning & Feature Selection

The primary aim of this paper was to identify a fast method for feature selection in simulation data. For this purpose, Feature selection via Decision Tree Classification was employed. Such methods have been used in the past and have gotten quite excellent results[3].Moreover, tree classification was used because of its ability to handle various kinds of data(binary/continuous) and the simplicity of the resulting model.These trees present a very natural way of looking at simulation data where, often, a few key degrees

of freedoms contain bimodal or trimodal distributions and the aim of the classifier is to find the decision boundary capable of dividing the data along these lines efficiently. For example the ϕ distribution in alanine dipeptide has a bimodal distribution around -100 and +50 degrees which divides the data along states 0/2 and 1/3.
Decision Trees find boundaries in data by selecting the best feature capable of distinguishing between the output classes. At each node τ, the optimal split is one that maximizes the Information Gain at that node via a reduction in the Gini impurity of the subnodes. This is done by calculating the Gini Impurity for a node n and then calculating the Gini Impurity for each sub-nodes after a split if made(eq 1). The information Gain for each feature(θ) is then the weighted gain in information for that feature at that node(eq 2). For the simplest binary classification model, this can be represented as follows.

$$GiniImpurity(\tau) = G(\tau) = 1 - \sum_{k=1}^{2} [P(k|\tau)]^2 \quad (1)$$

$$Inf.Gain = IG(\theta) = G(\tau) - \frac{nl}{n} G(\tau_l) - \frac{nr}{n} G(\tau_r) \quad (2)$$

$$GiniImportance = I_G(\theta) = \sum_{\tau} IG_\theta(\tau) \quad (3)$$

Where nl is the number of training examples in the left node,nr is the number of training examples in the right node, k is the summation over all the target classes, and τ is the summation over all the times that a particular feature was used to split the data. Features that are selected more often or can cause a higher reduction in the impurity of the data have a high Gini Importance. Thus, a feature τ's importance is the normalized total reduction in the impurity brought upon by that feature. I.e if a feature(such as ψ in Alanine dipeptide) can divide the data up better than other features, then it would have a higher Gini Importance. In order to avoid over fitting, cross-validation with 30-50 % of the data was performed. In both cases, the generalization error was less than 5 % indicating that the resulting trees were generalizing quite well.

Results & Discussion

Alanine Dipeptide

As it can be seen in figure 1,the alanine dipeptide is a very simple molecule with two main degrees of freedom characterized by the backbone ϕ and ψ dihedrals. The molecule is free to rotate around these bonds but its dynamics are limited by the interaction of the side chain atoms. Running the newtonian dynamics and clustering the resulting data using RMSD identifies chemically well known 4 macro states of the alanine. These 4 states correspond to the low energy basins in our high dimensional manifold(figure 2) . Using the vectorization scheme

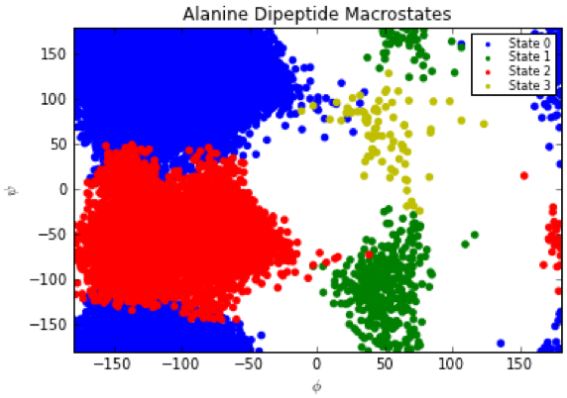

Figure 2: *Results from the unsupervised learning,using the **RMSD** metric, on the dynamics of Alanine dipeptide molecule projected onto its two main degrees of freedom ϕ & ψ.*

detailed above, a training set was fed into the a decision tree classifier. The aim being to find the features capable of distinguishing between the different macro states. Cross validation using 30% of the original data was performed and the results,projected back onto ϕ & ψ dihedrals are shown in figure 3. As show, the decision tree was able to learn the boundaries in our data quite well even though it was never given the RMSD as an input feature.

More importantly, the Gini Importance correctly identified the ϕ & ψ as being the only two features(out of ten) as being important in explaining the variation in the data. More importantly, the random Gaussian noise features are being completely ignored.

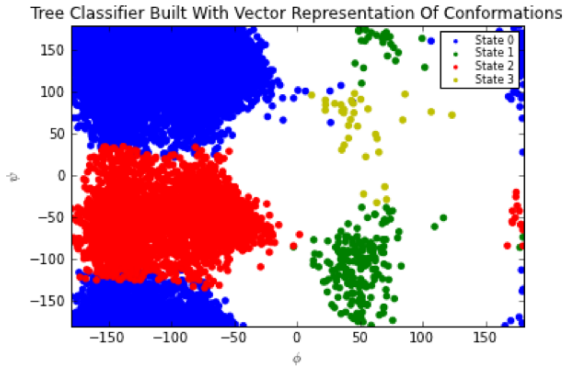

Figure 3: *Results from feeding 30% of the dataset into the trained classifier. The Decision tree algorithm correctly identifies the boundaries in the high dimensional dihedral space and correctly assigns the macro states despite not being given the rmsd for each case.The test classification error was less than 3 % in this particular case.*

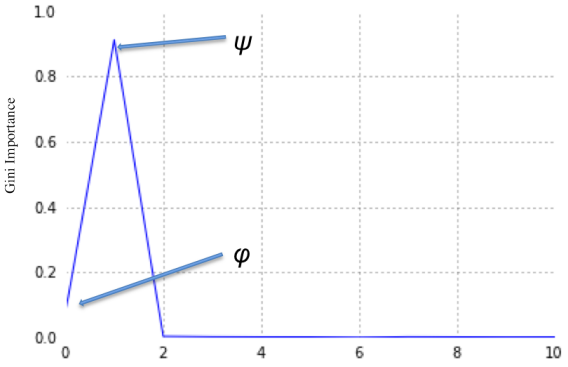

Figure 4: *The Gini Importance of each feature in the case of alanine dipeptide. As it can be see supervised learning correctly identifes the ϕ & ψ as being the most important features.The value of ϕ is less than that of ψ because of the difference in the number of training examples.*

Ubiquitin Enzyme

For the next test, the dynamics of the Ubiquitin enzyme were broken down via this analysis. Clustering the Ubiquitin enzyme trajectories led to the 3 macrostates as depicted in figure 5. Visual inspection of the clusters reveals that the system follows a three step model in locking the turn.

1 The α helix kinks up and towards the left.

2 While the α helix is up, the loop moves down.

3 The α helix kinks back down locking the loop in place.

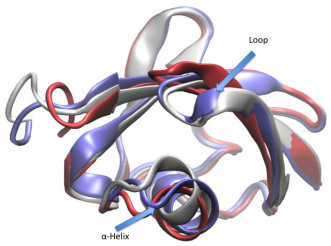

Figure 5: *The Ubiquitin enzyme with representative structures from the three states identified via unsupervised learning. The system moves from red to white to blue and its dynamics are characterized by the movement of turn and the α helix.*

But again, we are lacking important atomic level information about this movement. Similar to the alanine dipeptide, the conformations in Ubiquitin were represented as a 140 dimensional vector consisting of the backbone dihedrals of the proteins. 70-80% of the data was then fed into a decision tree classification system followed by cross-validation on the remaining 20 %. The Gini Importance of each of those dihedrals was then calculated and plotted(figure 6).

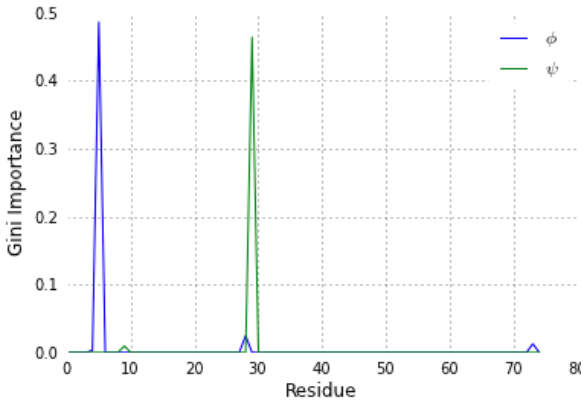

Figure 6: *Gini Importance for the backbone dihedrals in the Ubiquitin enzyme. The $6^{th}\phi$ corresponds to the loop movement and the $30^{th}\psi$ corresponds to the α helix kinking up and down.*

As shown above, the entire Ubiquitin motion can be broken down using the $6^{th}\phi$ and $30^{th}\psi$. The former corresponds to the loop moving down while the latter explains th helix kink. A time average of these two dihedrals(figure 7) confirms what was already suspected from the looking at the macrostate models,namely the three step process. However, the advantage of this analysis is that we can now systematically eliminate features that do not meaningfully explain the data.

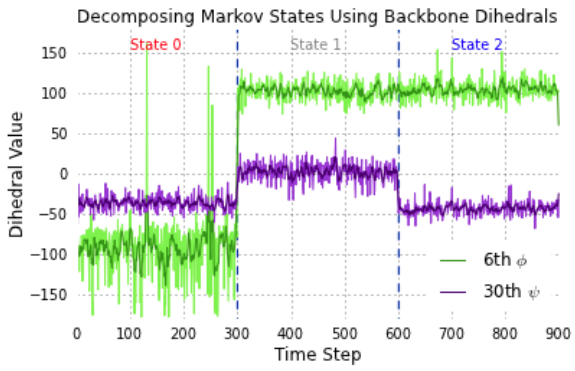

Figure 7: *Moving average plot for the dihedrals identified in figure 6.These plots follow what we observe via visual inspection. Namely that the helix(purple) kinks up followed by the loop(green) moving down and the helix moving back to lock the loop in place.*

Conclusions

The aim of this project was to develop a method for feature selection in MD trajectories. To accomplish this, a two step process was employed.The RMSD of heavy atoms was used to generate high dimensional clusters via the k-means algorithms. These cluster labels were then used in supervised learning and feature selection steps. To this end, cross-validated decision trees were trained on the vectorized representation of these trajectories and the Gini Importance of the resulting features was used as a feature selection criterion in understanding these trajectories.The two test cases,alanine dipeptide and Ubiquitin, highlighted the strengths of this approach and its ability to break down very high dimensional and highly correlated data. Future work will extend this method to larger and more complex systems such as kinases and G-protein coupled receptors(GPCRs).

Acknowledgments

The author would like to thank his colleague Gert Kiss for providing the trajectories and the unsupervised clustering results for the Ubquitin test case.

References

[1] KA Beauchamp, GR Bowman, TJ Lane, L Maibaum, IS Haque, VS Pande. JCTC 2011. MSMBuilder2: Modeling Conformational Dynamics at the Picosecond to Millisecond Timescale

[2] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[3] Tahir, N.M.; Hussain, A.; Samad, S.A.; Ishak, K.A.; Halim, R.A., "Feature Selection for Classification Using Decision Tree," Research and Development, 2006. SCOReD 2006. 4th Student Conference on , vol., no., pp.99,102, 27-28 June 2006 doi: 10.1109/SCORED.2006.4339317