Modelling Hydrogen Bond Stability by Regression Trees

Igor Chikalov, Mikhail Moshkov, Peggy Yao, and Jean-Claude Latombe

***Abstract*—Hydrogen bonds (H-bonds) play a key role in both the formation and stabilization of protein structures. However, H-bonds greatly vary in stability. Different local interactions may reinforce or weaken an H-bond. This paper describes inductive learning methods to train a protein-independent probabilistic model of H-bond stability from molecular dynamics (MD) simulation trajectories. The training data describes H-bond occurrences at successive times along these trajectories by the values of 32 attributes. A trained model is constructed in the form of a regression tree. Experimental results demonstrate that such models can predict H-bond stability quite well. In particular, their performance is roughly 20% better than that of models based on H-bond energy alone. In addition, they can accurately identify a large fraction of the least stable H-bonds in a given conformation. The paper discusses several extensions that may yield further improvements.**

***Index Terms*—Molecular dynamics, machine learning, regression tree.**

I. Introduction

A hydrogen bond (H-bond) corresponds to the attractive electrostatic interaction between a covalent pair D—H of atoms, in which the hydrogen atom H is bonded to a more electronegative *donor* atom D, and an electronegative *acceptor* atom A. Due to their strong directional character, short distance ranges, and their relatively large number in a folded protein, H-bonds play a key role in both the formation and stabilization of protein secondary and tertiary structures [1], [2], [3].

Unlike covalent bonds, H-bonds greatly vary in stability. They can form and break while a protein deforms. For instance, the transition of a folded protein from a non-functional substate to a functional (*e.g.*, binding) substate may require some H-bonds to break and others to form [4]. The intrinsic strength of an individual H-bond has been studied from an energetic viewpoint [5], [6], [7], [8], [9]. But energy alone may not be a very good predictor of H-bond stability. Other local interactions may reinforce or weaken an H-bond. Moreover, several "redundant" H-bonds may reinforce one another by rigidifying the same group of atoms. To better understand the possible deformation of proteins in their folded states, it is desirable to create models that can reliably predict the *stability* of an H-bond not just from its energy, but also from its local environment.

In this paper we apply inductive learning methods to train a protein-independent probabilistic model of H-bond stability from a training set of molecular dynamics (MD) simulation trajectories of various proteins. The input to the training procedure is a data table in which each row gives the value of several (32) attributes, called predictors, of an H-bond and its local environment at a given time t in a trajectory, as well as the measured stability of this H-bond over an interval of time $(t, t+\Delta)$. The output is a function σ of a subset of predictors that estimates the probability that an H-bond present in the conformation[1] c achieved by a protein will be present in any conformation achieved by this protein within a time interval of duration Δ. The value of Δ defines the *timescale* of the prediction.

Section 0 gives a precise statement of the problem addressed in this paper. Section III presents the machine learning approach that is used to solve this problem. Section IV describes details of the training algorithm. Section V describes our experimental setup. Section VI discusses test results obtained with models trained using our method. Section VII suggests future developments that may lead to improving trained models.

II. Problem Statement

Let c be the conformation of a protein P at some time considered (with no loss of generality) to be 0 and H be an H-bond present in c. Let $M(c)$ be the set of all physically possible trajectories of P passing through c and π be the probability distribution over this set. We define the stability of H in c over the time interval Δ by a function $\bar{\sigma}: (H, c, \Delta) \to [0,1]$:

$$\bar{\sigma}(H,c,\Delta) \square \sum_{q \in M(c)} \left[\frac{1}{\Delta} \int_0^{\Delta} I(q,H,t)dt \right] \pi(q) \quad (1)$$

where $I(q,H,t)$ is a Boolean function that takes value 1 if H is present in the conformation $q(t)$ at time t along trajectory q, and 0 otherwise. The value $\bar{\sigma}(H,c,\Delta)$ can be interpreted as the probability that H will be present in the conformation of P at any specified time $t \in (0,\Delta)$, given that P is at conformation c at time 0.

Our goal is to design a method for generating good approximations σ of $\bar{\sigma}$. We also want these approximations to be protein-independent, i.e., the argument

Manuscript received May 1, 2012; revised May 25,2012. This work was supported in part by a grant from the KAUST-Stanford Academic Excellence Alliance program.

I. Chikalov and Mikhail Moshkov are with Mathematical and CS & Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia (e-mail: igor.chikalov@kaust.edu.sa).

P. Yao and J. C. Latombe are with Computer Science Department, Stanford University, Stanford, CA 94305, USA (e-mail: latombe@cs.stanford.edu)

[1] A protein *conformation* defines the relative positions of all the atoms in the protein.

c may be a confirmation of any protein.

III. General Approach

We use machine learning methods to train a stability model σ from a given set Q of MD simulation trajectories of various proteins. Each trajectory $q \in Q$ is a discrete sequence of conformations of a protein. These conformations are reached at times $t_i = i \times \delta$, $i = 0,1,2,\ldots$, called *ticks*, where δ is typically on the order of the picoseconds.[2] We detect the H-bonds[3] which are present in each conformation $q(t_i)$ using the geometric criteria given in [11]. These criteria, shown in Fig. 1, specify conditions on distances and angles that must be satisfied by the atoms H (hydrogen), D (donor), A (acceptor), and AA (the atom covalently bonded to A) for the H-bond to be considered present. An H-bond in a given protein is uniquely identified across different conformations by its donor, acceptor, and hydrogen atoms. So, we call the presence of a specific H-bond H in a conformation $q(t_i)$ an *occurrence* of H in $q(t_i)$.

For each *occurrence* of an H-bond H in $q(t_i)$ we compute a fixed list of predictors, some numerical, others categorical. Some are time-invariant, like the types of the donor and acceptor atoms and the number of residues along the main-chain between the donor and acceptor atoms. Others are time-dependent. Among them, some describe the geometry of H in $q(t_i)$, e.g., the distance between the hydrogen and the donor atoms and the angle made by the donor, hydrogen, and acceptor atoms. Others describe the local environment of H in $q(t_i)$, e.g., the number of other H-bonds within a certain distance from the mid-point of *H*. The complete list of 32 predictors used in our work is given in Appendix.

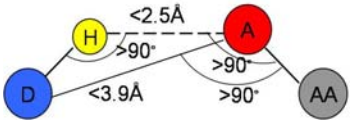

Fig. 1. Constraints on H-bond geometry

We train σ as a function of these predictors. The predictor list defines a predictor space Σ and every H-bond occurrence maps to a point in Σ. Given the input set Q of trajectories, we build a data table in which each row corresponds to an occurrence h of an H-bond present in a conformation $q(t_i)$ contained in Q. So, many rows may correspond to the same H-bond at different ticks. In our experiments, a typical data table contains several hundred thousand rows. Each column, except the last one, corresponds to a predictor p and the entry (h, p) of the table is the value of p for h. The entry in the last column is the *measured* stability y of the H-bond occurrence in conformation $q(t_i)$. More precisely, let H be the H-bond of which h is an occurrence. In addition, let $l = \Delta / \delta$, where Δ is the duration over which we wish to predict the stability of h (see Section II), and let $m \le l$ be the number of ticks t_k, $k = i+1, i+2, \ldots, i+l$, such that H is present in $q(t_k)$. The measured stability y of h is the ratio m / l. In the tests reported below we chose $l = 50$, as this value both provides a ratio m / l large enough for the measured stability to be statistically meaningful, and corresponds to an interesting prediction timescale (50ps). Typically, most H-bond occurrences are quite stable: over 25% have measured stability 1, about 50% higher than 0.8, and only 15% less than 0.3.

IV. Model Training

We build σ as a binary regression tree using the CART (Classification and Regression Tree) method [12]. This well-studied machine learning approach has been one of the most successful in practice. Regression trees are often simple to interpret. Not only may this simplicity eventually lead to pertinent insights to better understand H-bond stability; it also allows us to perform many experiments, compare the generated trees, and analyze the relative importance of the predictors.

One important issue to deal with is the violation of the IID property in the training data table. The IID property would require that H-bond occurrences follow a certain fixed probability distribution, and that each row of a data table input to the learning algorithm is sampled according to this distribution, independent of the other rows. The satisfaction of this property is critical for the trained model σ to predict reliably the stability of H-bonds in new protein conformations. However, it is likely to be violated, mainly because several H-bond occurrences in a data table correspond to the same H-bond. More specifically, two occurrences of the same H-bond along the same trajectory are more likely to be similar along several dimensions of the predictor space Σ than two occurrences of distinct H-bonds, especially if these bonds belong to different proteins. This may result into correlations between predictor values and measured stability that are bond-specific and thus do not extend to other bonds.

To address this issue, we apply a two-step split calculation procedure [13]. The training data table is divided at random into three tables T_1, T_2, and T_3 so that occurrences of the same bond are not split between the tables. The split predictor p and the split value r at a node N are computed separately, using one of the two tables T_1 and T_2:

[2] MD simulation trajectories are computed by integrating the equations of motion with a time step on the order of the femtoseconds (10^{-15}s) in order to take into account high-frequency thermal vibrations. However, to reduce the amount of stored data, they are usually sub-sampled at a time step on the order of the picoseconds (10^{-12}s).

[3] We only consider H-bonds inside a protein. We ignore H-bonds between a protein and the solvent.

1) The best split value r_p^* is computed for each predictor p using T_1 : $r_p^* = \arg\max_r\{w_1(p,r)\}$, where $w_1(p,r)$ denotes the score of split (p,r) on T_1.
2) The best split predictor p^* is computed using T_2 with the best split values computed at the previous step: $p^* = \arg\max_p\{w_2(p,r_p^*)\}$, where $w_2(p,r_p^*)$ denotes the score of split (p,r_p^*) on T_2.
3) The selected split is (p,r_p^*).

Assume that the best split value computed in the first step is obtained for some predictor p'. If this best value results from a bond-specific correlation between p' and measured stability in T_1, then this correlation is unlikely to happen again in T_2, since T_1 and T_2 describe disjoint sets of H-bonds. So, in the second step, predictor p' will likely have a small score $w_2(p,r_p^*)$ and so will not be selected as the split predictor.

Finally, we reduce the complexity of the generated tree using the standard CART tree pruning procedure [12]. This procedure removes nodes from the tree iteratively and selects the pruned tree that minimizes the prediction error on table T_3.

V. Experimental Setup

A. MD Trajectories

In the experiments reported below, we used 6 MD simulation trajectories picked from different sources and generated with different force fields: *1c9oA, 1e85A, 1g9oA_1*, and *1g9oA_2* from [14], and *1eia* and *complex* from [15]. In all of these trajectories the time interval δ between two successive ticks is 1ps. Each trajectory starts from a folded conformation resolved by X-ray crystallography.

Trajectories obtained with different proteins allow us to test if a model σ trained with one protein can predict H-bond stability in another protein. Similarly, trajectories generated with different force fields allow us to test if a model σ trained with one force field can predict H-bond stability in trajectories generated with another force field.

B. Data Tables

From each trajectory we derived a separate data table in which the rows represent the detected H-bond occurrences. Table I lists the number of distinct H-bonds detected in each trajectory and the total number of H-bond occurrences extracted.

TABLE I: Number of Distinct H-bonds and H-Bond Occurrences Detected in Each Trajectory

Trajectory	# H-bonds	# occurrences
1c9oA	263	363463
1e85A	525	1253879
1eia	757	379573
1g9oA_1	374	558761
1g9oA_2	397	544491
Complex	1825	348943

The measured stability y of an H-bond H in $q(t_i)$ is computed as described in Section III, as the ratio of the number of ticks where the bond is present in the time interval $[t_i, t_i + l\times\delta]$ in trajectory q divided by the total number of ticks l in this interval.

The values of the time-varying predictors are subject to thermal noise. Since a model σ will in general be used to predict H-bond stability in a protein conformation sampled using a kinematic model ignoring thermal noise (*e.g.*, by sampling the dihedral angles ϕ, ψ, and χ) [10], we chose to average the values of these predictors over l' ticks to remove thermal noise. More precisely, let h be an H-bond occurrence in $q(t_i)$.The value of a predictor stored in the row of the data table corresponding to h is the average value of this predictor in $q(t_{i-l'+1}), q(t_{i-l'+2}), \ldots, q(t_i)$, where $t_{i-l'+k} = t_i - (l'-k)\times\delta$. Experiments show that $l' = 50$ is near optimal.

C. Performance Measures

The performance of a regression model can be measured by the root mean square error (RMSE) of the predictions on a test dataset. Let $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ be a data table, where each x_i, $i = 1, \ldots, n$, denotes a vector of predictor values for an H-bond occurrence and y_i is the measured stability of the H-bond. For a given table *T*, the RMSE of a model σ is defined by:

$$RMSE(\sigma, T) = \sqrt{\frac{1}{n}\sum_i (y_i - \sigma(x_i))^2}$$

As RMSE depends not only on the accuracy of σ, but also on the table T, some normalization is necessary in order to compare results on different tables. So, in our tests we compute the decrease of RMSE relative to a base model σ_0. The *relative base error decrease* (or *RBED*) is then defined by:

$$RBED(\sigma, \sigma_0, T) = \frac{RMSE(\sigma_0, T) - RMSE(\sigma, T)}{RMSE(\sigma_0, T)} \times 100\%$$

In most cases, σ_0 is simply defined by $\sigma_0(x) = \frac{1}{n}\sum_i y_i$, i.e., the average measured stability of all H-bond occurrences in the dataset. In other cases, σ_0 is a model based on H-bond energy.

VI. Experiments

A. Training on Data from Multiple Trajectories

Here, we trained models on data tables obtained by mixing subsets of 5 data tables and we tested these models on the remaining data table. For each combination of 5 data tables, we trained 10 models by mixing different fractions of the 5 data tables. For each model, the mixed data table was partitioned into the three tables T_1, T_2, and T_3: 60% of the

data went to T_1, 20% to T_2, and 20% to T_3. No two tables contain occurrences of the same H-bond. Furthermore, we trained 4 groups of models varying in the tree's maximal depth (5 or 15) and in the fraction of H-bond occurrences taken from each data table (10% or 50%). So, in total, 240 models were generated in this experiment.

Table II shows the mean RBED value for each combination of data tables and each group of models. In rows 3 through 8 we indicate the data table used for testing the models trained on a combination of the 5 other data tables. Fig. 2 shows the distribution of the RBED values for the models built with the settings of in the first data column of Table II (i.e., maximal depth of 5 and 10% from each data table).

One can see that the variance of RBED values is quite small, meaning that the training process yields models that are stable in performance. The RBED values are lower for models tested on *complex*. In fact, the trajectory *complex* was generated for a complex made of a protein and a ligand,

while all other trajectories were generated for a single protein. So, it is likely that *complex* contains H-bonds in situations that did not occur in any of the other trajectories.

TSBLE II: MEAN RBED VALUES OBTAINED IN EXPERIMENT A

Max tree depth	**5**		**15**	
Fraction of data	***0.1***	***0.5***	***0.1***	***0.5***
1c9oA	46.92	47.07	47.24	46.87
1e85A	59.37	59.59	59.03	59.04
1eia	42.6	43.15	43.35	43.46
1g9oA_1	50.93	50.69	51.42	51.38
1g9oA_2	45.29	45.45	45.65	45.89
complex	37.9	38.08	38.07	38.38
Average	47.17	47.34	47.46	47.5

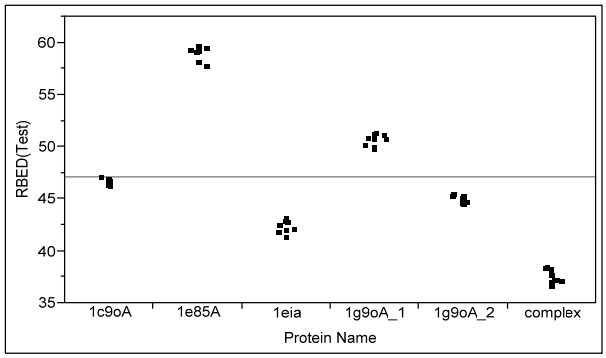

Fig. 2. Distribution RBED values for the models built with settings specified in the first data column of Table II

These results suggest that we should try to train models with a larger set of trajectories. We actually did some experiments using a few additional trajectories, but with no noticeable improvement. Most likely these trajectories did not contain enough H-bonds in situations that did not already occur in the trajectories of Table I.

Another observation is that deeper trees and larger data fractions tend to improve model accuracy, but the very small gain is not worth the additional model or computation complexity.

B. Comparison with FIRST-Energy Model

Here, the models are the same as those generated in Experiment A in the first data column of Table II (maximal depth of 5 and 10% from each data table). But we now compare them to a regression tree σ_0 built from the same training data using FIRST_energy as the only predictor (predictor #32 in Appendix). FIRST_energy is the value of the function used in FIRST [8] to evaluate the energy of an H-bond occurrence; it is a slightly modified version of the Mayo energy [5]. We compute RBED values as defined in Section V.C, where σ_0 is the regression tree based on H-bond energy only.

Table III shows the mean RBED values. Tests on all 6 data tables show that the more complex models are significantly more accurate than the model based on FIRST_energy only. Overall, these results confirm that the stability of an H-bond occurrence depends not only on its energy, but also on other parameters.

TABLE III: MEAN VALUES OF RBED COMPUTED IN EXPERIMENT B

1c9oA	***1e85A***	***1eia***	***1g9oA_1***	***1g9oA_2***	***complex***
26.36	27.95	5.65	22.63	19.63	19.42

C. Identification of Least Stable H-Bonds

Most H-bond occurrences tend to be stable. So, accurately identifying the weakest ones is important if one wishes to predict the possible deformation of a protein [10].

Here, we measure how well the models generated in Experiment A (again, in the first data column of Table II) identify the least stable H-bonds occurrences in the test data table. In each test table *T*, we first identify the subset *S* of the 10% least stable H-bond occurrences (i.e., the H-bond occurrences with the smallest *measured* stability). Using a regression tree σ trained with a combination of data from the 5 other tables, we then sort the H-bond occurrences in *T* in ascending order of predicted stability and we compute the fraction $w \in [0,1]$ of S that is contained in the first $100 \times u\%$ occurrences in this sorted list, for successive values of $u \in [0,1]$. We call the function $w(u)$ the *identification curve* of the least stable H-bonds for σ.

Fig. 3 plots identification curves for 1c9oA table: the dotted curve is the ideal identification curve (the one that would be obtained with a model that perfectly predict the 10% least stable H-bonds), the solid curve is obtained with one (randomly picked) regression tree computed in Experiment A, and the dashed curve is obtained by sorting H-bond occurrences in decreasing values of FIRST_energy.

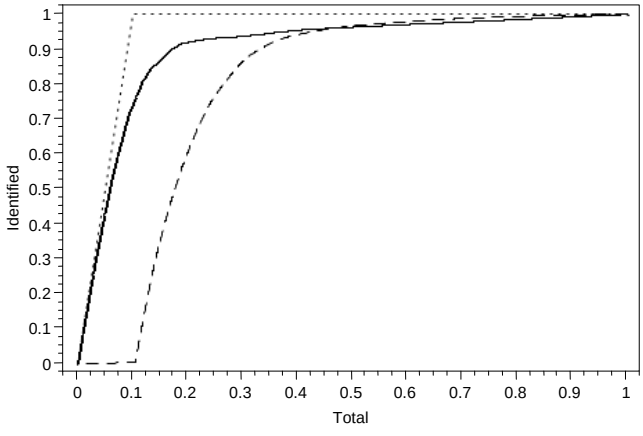

Fig. 3. Identification curves of the least stable bonds for 1c9Oa

Table IV shows $w(0.1)$ value for each of the 6 test tables. One can see that the models computed in Experiment A perform well in general. For models tested on data tables other than *complex*, about 70% of the 10% truly least stable H-bond occurrences are actually among the 10% predicted as the least stable. However, several curves show a rather long tail of poorly ranked unstable bonds. For example, the set of the 50% least stable bonds predicted by the model tested on *1eia* still misses about 5% of the truly least stable bonds.

TABLE IV: IDENTIFICATION CURVE VALUE AT 0.1

Protein	1c9oA	1e85A	1eia	1g9oA_1	1g9oA_2	complex
Model	0.73	0.77	0.65	0.77	0.68	0.50
FIRST	0.006	0.46	0.65	0.62	0.58	0.45

Not surprisingly, the results for *complex* are much less satisfactory. The regression models generated in Experiment A perform consistently better than the FIRST_energy-only models, but for *1eia* the difference is small.

VII. CONCLUSION AND FUTURE WORK

In this paper we have described machine learning methods to train regression trees modeling H-bond stability in a protein. The training and test data are in the form of tables whose rows describe H-bond occurrences at successive times along Molecular Dynamics simulation trajectories and columns give the values of various predictors.

Test results demonstrate that trained models can predict H-bond stability quite well. In particular, we have shown that their performance is significantly better (roughly 20% better) than that of a model based on H-bond energy alone. We have also shown that they can accurately identify a large fraction of the least stable H-bonds in a given conformation. However, our results also suggest that better models could be obtained with a richer set of MD simulation trajectories. In particular, the trajectories used in our experiments might be too short to characterize the stability of H-bonds that break and form during a transition between sub states.

We believe that the training methods could be improved in several ways:

- To eliminate thermal noise, predictor values are averaged over time windows of 50 ticks, independent of the elapsed time between two ticks. It would be better to average predictor values before sub-sampling MD simulation trajectories (see Footnote 2). This would result in a much shorter averaging window, hence it would greatly reduce the risk of filtering out changes in predictor values that are important for H-bond stability. Unfortunately, in our trajectories we only had access to the data after sub-sampling.
- More sophisticated learning techniques could be used. For example, instead of generating a single tree, we could generate an ensemble of trees, such as Gradient Boosting Trees [16] or Random Forests [17]. A regression tree could also be enriched by using splits on linear combinations of predictors and by fitting linear regression models at the leaves.
- We could use rigidity analysis methods such as those described in [10] to decompose a protein into rigid groups of atoms (based on distance constraints imposed by covalent and hydrogen bonds present in the current conformation). This would allow us to apply Bayesian techniques to align the predicted stability of individual H-bonds in the same rigid group. By doing so, we could better predict the collective behavior of related H-bonds and avoid solitary incorrect predictions.
- Finally, the notion of stability itself could be refined, for example by distinguishing between the case where an H-bond frequently switches on and off during a prediction window and the case where it rarely switches.

Overall, we believe that considerable progress can still be made in learning more accurate and robust models of H-bond stability.

APPENDIX: LIST OF PREDICTORS

#	Feature Name	Feature Meaning	Type[4]
		Distance-related	
1	Dist_H_D	Distance between H and donor (covalent bond length)	N
2	Dist_H_A	Distance between H and acceptor (H-bond length)	N
3	Dist_A_AA	Distance between acceptor and the atom it is covalently bonded to	N
4	Dist_D_A	Distance between donor and acceptor	N
5	Dist_D_AA	Distance between donor and AA	N
6	Dist_H_AA	Distance between H and AA	N
		Angle-related	
7	Ang_D_H_A	Angle Donor-H-Acceptor	N
8	Ang_H_A_AA	Angle H-acceptor-the atom the acceptor covalently bonded to	N
9	Ang_D_A_AA	Angle donor-acceptor-the atom the acceptor covalently bonded to	N
10	Ang_planar	Angle between plane D-H-A and H-A-AA	N
		Atom	
11	Atom_type_D	Donor atom type (e.g., O, N, S, C)	C
12	Atom_type_A	Acceptor atom type (e.g., N, O, S)	C
13	Atom_type_AA	AA atom type (e.g, P, C, S)	C
		Residue	
14	Resi_name_H	Donor residue name (3 letter code)	C
15	Resi_name_A	Acceptor residue name (3 letter code)	C
16	Resi_type_H	Donor residue type. Nonpolar (Ala, Val, Leu, Ile, Trp, Met, Pro), Polar_acidic (Asp, Glu), Polar_uncharged (Gly,	C

[4]N designates a numerical predictor and C a categorical predictor.

		Ser, Thr, Cys, Tyr, Asn, Gln), Polar_basic (Lys, Arg, His)	
17	Resi_type_A	Acceptor residue type	C
18	Resi_sch_size_H	Donor residue side-chain size, i.e., number of atoms in the side-chain	N
19	Resi_sch_size_A	Acceptor residue side-chain size	N
Bond structure type			
20	Sec_type	Secondary structure of the H-bond. MA (H-atom and O-atom are in same helix, middle portion), MB (same strand, middle), EA (same helix, end), EB (same strand, end), AL (helix-loop), BL (helix-loop), DA (different helices), SL (same loop), DL (different loops). Don't have DB (different strands) because it's hard to know which strand pairs with which strand to form the sheet.	C
21	Ch_type	H and O are on mch or sch: MM (mch-mch), MS (mch-sch), SS (sch-sch)	C
22	Rgd_type	SR (H and A are in the same rigid body), DR (different rigid body)	C
23	Range	Difference in the residue numbers of donor and acceptor, i.e., abs($Resi_{donor}$-$Resi_{acceptor}$)	N
24	Hybrid_state	Hybridization state (sp^2-sp^2, sp^2-sp^3, sp^3-sp^2, sp^3-sp^3)	C
25	Num_furcated_H	Number of H-bonds share the H-atom as this H-bond	N
26	Num_furcated_A	Number of H-bonds share the acceptor as this H-bond	N
Environment			
27	Num_potential_As	Number of potential acceptors (N, O, or S) in 3Å of H (but not covalently bonded to it) besides the current acceptor	N
28	Num_hb_seqNbr	Number of sequence-neighboring H-bonds, i.e., number of H-bonds of residues ±2 of $Resi_{donor}$ and $Resi_{acceptor}$	N
29	Num_hb_spaceNbr	Number of space-neighboring H-bonds, i.e., number of H-bonds within 5Å of the mid-point of this H-bond	N
30	Num_hb_spaceRgdNbr	Number of space-neighboring H-bonds in the same rigid-body, i.e., number of Num_hb_spaceNbr in the same rigid-body as this H-bond (cross-rigid = -100)[5]	N
31	Surface	Average surface percentage of the H atom and acceptor	N
Energy			
32	FIRST_energy	Modified Mayo potential implemented in FIRST [8]	N

[5]Here, we first use the FIRST software [TLR+01] to decompose the protein into rigid groups of atoms based on distance constraints imposed by covalent and hydrogen bonds present in the current conformation. Num_hb_spaceRgdNbr is the number of H-bonds located within 5Å of the mid-point of the analyzed H-bond in the same rigid component.

ACKNOWLEDGMENT

The authors thank L. Kavraki (Rice University), V. Pande (Stanford), M. Levitt (Stanford), and J. Wang Tsai (University of the Pacific) for providing us MD simulation trajectories and for useful comments during our work.

REFERENCES

[1] E. N. Baker, "Hydrogen bonding in biological macromolecules," *International Tables for Crystallography*, vol. F, chapter 22.2, pp. 546-552, 2006.

[2] A. R. Fersht and L. Serrano, "Principles in protein stability derived from protein engineering experiments," *Curr. Opin. Struct. Biol.*, vol. 3, pp. 75-83, 1993.

[3] D. Schell, J. Tsai, J. M. Scholtz, and C. N. Pace, "Hydrogen bonding increases packing density in the protein interior," *PROTEINS: Structure, Function, and Bioinformatics*, vol. 63, 2006, pp. 278-282.

[4] Z. Bikadi, L. Demko, and E. Hazai, "Functional and structural characterization of a protein based on analysis of its hydrogen bonding network by hydrogen bonding plot," *Archives of Biochemistry and Biophysics*, vol. 461, pp. 225-234, 2007.

[5] B. I. Dahiyat, D. B. Gordon, and S. L. Mayo, "Automated design of the surface positions of protein helices," *Protein Science*, vol. 6, pp. 1333-1337, 1997.

[6] M. Levitt, "Molecular dynamics of hydrogen bonds in bovine pancreatic trypsin unhibitor protein," *Nature*, vol. 294, pp. 379-380, 1981.

[7] K. Morokuma, "Why do molecules interact? The origin of electron donor-acceptor complexes, hydrogen bonding, and proton affinity," *Accounts of Chemical Research*, vol. 10, pp. 294-300, 1977.

[8] A. J. Rader, B. M. Hespenhelde, L. A. Kuhn, and M. F. Thorpe, "Protein unfolding: rigidity lost," *Proceedings of the National Academy of Sciences*, vol. 99, pp. 3540-3545, 2002.

[9] M. A. Spackman, "A simple quantitative model of hydrogen bonding," *J. Chem. Phys.*, vol. 85, pp. 6587-6601, 1986.

[10] M. F. Thorpe, M. Lei, A. J. Rader, D. J. Jacobs, and L. A. Kuhn, "Protein flexibility and dynamics using constraint theory," *Journal of Molecular Graphics and Modeling*, vol. 19, pp. 60-69, 2001.

[11] I. K. McDonald and J. M. Thornton, "Satisfying hydrogen bonding potential in proteins," *J. Mol. Biol.*, vol. 238, pp. 777-793, 1994.

[12] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, CRC Press, 1984.

[13] E. Tuv, A. Borisov, and K. Torkokola, "Best subset feature selection for massive mixed-type problems," *Proc. Intelligent Data Engineering and Automated Learning (IDEAL'06)*, Lecture Notes in Computer Science, Springer, vol. 4224, pp. 1048-1056, 2006.

[14] H. Joo, X. Qu, R. Swanson, C. M. McCallum, and J. Tsai, "Modeling the dependency of residue packing upon backbone conformation using molecular dynamics simulation," *Comput. Biol. Chem.*, to be published.

[15] N. Haspel, D. Ricklin, B. Geisbrecht, J. D. Lambris, and E. K. Lydia, "Electrostatic contributions drive the interaction between staphylococcus aureus protein Efb-C and its complement target C3d," *Protein Science*, vol. 17, pp. 1894-1906, 2008.

[16] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, pp. 1189-1232, 2000.

[17] L. Breiman and E. Schapire, "Random forests," *Machine Learning*, vol. 45, pp. 5-32, 2001.