MACHINE LEARNING FOR POSE SELECTION

By

Jun Pei

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Chemistry—Doctor of Philosophy

2020

ABSTRACT

MACHINE LEARNING FOR POSE SELECTION

By

Jun Pei

Scoring functions play an important role in protein related systems. In general, scoring functions were developed to connect three dimensional structures and corresponding stabilities. In protein-folding systems, scoring functions can be used to predict the most stable protein structure; in protein-ligand and protein-protein systems, scoring functions can be used to find the best ligand structure, predict the binding affinities, and identifying the correct binding modes. Potential functions make up an essential part of scoring functions. Each potential function usually represents a different interaction that exists in a protein or protein-ligand system. In many traditional scoring functions, energies calculated from individual potential functions were simply sum up to estimate the stability of the whole structure. However, it is possible that those energies cannot be directly added together. In other words, some of those potential functions might describe more important interactions, whereas other potential functions are used to represent insignificant interactions. Hence, it will be useful to construct a model, which can emphasize the important interactions, and ignore the insignificant ones.

With the development of machine learning (ML), it became possible to build up a model, which can address the importance of different interactions. In this work, we combined random forest (RF) algorithm and different potential function sets to solve the pose selection problem in protein-folding and protein-ligand systems. Chapter 3 and chapter 5 show the results of combing RF algorithm with knowledge-based potential functions and force field potential functions for protein-folding systems. Chapter 4 shows the result of combining the RF method with knowledge-based

potential functions for protein-ligand systems. As the results from chapter 3, chapter 4, and chapter 5, it is obvious that the RF model based on potential functions outperformed all of the traditional scoring functions in accuracy and native ranking tests. In order to test the importance of potential functions, scrambled and uniform artificial potential function sets were generated in chapter 3, the test results suggest that the potential function set is important in the model, and the most useful information from knowledge-base potential functions are the peak positions. In chapter 5, the importance of the RF algorithm and potential functions were tested. The results also suggest that the potential functions are important, and the RF model is also necessary to achieve the best performance.

This dissertation is dedicated to my parents, who always love, trust, and support me.

ACKNOWLEDGEMENTS

I thank my parents for the generous love. My parents have raised me and spared no effort to give me a good life and education. They encouraged me to follow my heart, and respect my decisions. They set up examples for me about good behaviors, and support me when I was stressed out. I thank my friends for their friendship and company.

I wish to express my deepest gratitude to my advisor, Professor Kenneth M. Merz Jr., for his patient love and kind support of my research. He respects my decisions and helps me to fulfill my dream. His positive attitude encouraged me when I was pessimistic; his broad thoughts show me the possibilities of research; his deep insights illuminate the research direction for me. I will never touch machine learning without him. I thank my guidance committee members: Professor Katharine C. Hunt, Professor Robert I. Cukier, and Professor Benjamin G. Levine. I thank the members of the Merz research group for their company. Among them, I specifically thank Zheng for handing me the project of constructing KECSA2 for protein systems. I thank Lin for helping me perform Amber calculations in chapter 2 and chapter 3.

Being a member of the Merz research group is a precious memory that I will never forget.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

KEY TO ABBREVIATIONS

Amber	Assisted Model Building with Energy Refinement
CASF	Comparative Assessment of Scoring Functions
CHAID	Chi-square automatic interaction detection
ff	Force Field
FFENCODER	Force Field Encoded Program
FN	False Negative
FP	False Positive
KECSA2	Knowledge-Based and Empirical Combined Scoring Algorithm 2
ML	Machine Learning
RF	Random Forest
RMSD	Root-mean-square deviation
SVM	Support Vector Machine
TN	Ture Negative
TP	Ture Positive

CHAPTER 1: INTRODUCTION

1.1 Introduction for protein-folding pose selection

According to the "thermodynamic hypothesis", the native protein in its preferred chemical environment should have a structure with the lowest Gibbs free energy.[1] Identifying the three dimensional protein structure with the lowest Gibbs free energy is important in many applications to protein systems, including protein folding,[2-40] protein structure prediction,[41-52] and protein design problems.[53-61] It is a challenge to understand the relationship between the three dimensional structure of a protein and the corresponding stability. For example, in the protein design field, it is important to predict the native structure of a protein (most stable structure) to tailor the properties of that protein. Scoring functions were developed to connect three dimensional structures and corresponding stabilities. Currently, there are three broad categories of scoring functions for protein-folding. (i) Physics-based scoring functions,[62-68] this kind of scoring functions usually employ relatively simple equations to represent bond, angle, dihedral, van der Waals, and electrostatic interactions and calculate the score of a protein at atomic level. (ii) Knowledge-based scoring functions,[2-33] also referred to as statistical-based scoring functions. Those functions use crystal structures of proteins as the data source and extract the radial distribution functions of atom/residue pairs based on protein crystal structures. Then, the reference state can be constructed based on different statistical models to generate the "pure" interactions between different atom/residue pairs. Hence, those scoring functions can generate scores of proteins at atomic/residual level. (iii) Machine learning-based scoring functions (ML-based scoring functions),[34-40] those functions usually include more features of a protein, which is hard to be

involved in physics-based and knowledge-based scoring functions. ML-based scoring functions utilize different ML algorithms and information from protein structures to predict the stabilities of protein structures.

1.2 Introduction for protein-ligand pose selection

Similar to protein-folding pose selection described in section 1.1, scoring functions are also needed for protein-ligand pose selection problems. In drug discovery, it is important to identify the binding mode of a small ligand (identify the most stable binding pose). Without the information of the correct binding mode, it will be hard to optimize the lead structures. The existing scoring functions for protein-ligand systems can be classified as four groups: physics-based,[69-79] knowledge-based,[80-91] empirical based,[92-98] and ML-based scoring functions.[99-110] The physics-based, knowledge-based and ML-based scoring functions are similar to those functions discussed in protein-folding pose selection. Empirical based scoring functions were constructed upon an assumption that the total binding affinity between a protein and ligand can be decomposed into basic components with different coefficients. The coefficients can be determined with a multivariate regression model and a benchmark contains experimentally determined protein-ligand structures and corresponding binding affinities.

1.3 Combine machine learning and conventional scoring function

Currently, with the success of ML in the computer vision area, there are more ML-based scoring functions were developed for protein-folding and protein-ligand systems.[34-40, 99-110] With the flexibility of ML algorithms, a large variety of information was employed as features in ML-based scoring functions. For example, in protein-folding systems, the secondary surface area of proteins and solvent accessible surface area were used as input features.[35, 38] In protein-ligand systems, topological fingerprint and three dimensional graph of a protein-ligand complex were used as inputs.[100-101] And some of the information used in ML-based scoring functions cannot be used in other traditional scoring functions. Although those ML-based scoring functions achieved higher accuracy and lower computational cost than conventional scoring functions, some important information might be lost when the ML-based scoring functions were constructed.

As we know, there are some well performed scoring functions in protein-folding and protein-ligand systems. Hence, important information might be buried in those conventional scoring functions. Physics-based, knowledge-based, and empirical based scoring functions share a common character that, all of them employed potential functions to describe interactions between atoms/residues. Potential functions can be used to denote three dimensional structures as a series of pair wise energies. In conventional scoring functions, the importance of all pair wise energies were treated as the same. However, the importance of each pair wise energy might be different. Due to date and algorithm limitations, it is hard to address the importance of different pair wise interactions in the past. With the development of ML algorithms, it became possible to utilize ML models to address the importance of different interactions.

It is important to understand basic ML algorithms in order to select the suitable ML models, which can emphasize more important pair wise interactions and ignore insignificant ones. In general, most machine learning problems can be classified into two groups: supervised and unsupervised learning problems. Supervised learning refers to the cases when a model is fitted to predict values or categories. Unsupervised learning refers to the cases when there are no real values or classes to be predicted. In supervised learning, there are two categories of models, regression and classification. Regression models are usually used to predict continuous values, and classification models tend to predict the category of each input. In this study, the goal is to build up a ML model that can correctly identify the most stable structure in protein-folding and protein-ligand systems. Hence, supervised models can be used to solve the problem. On the other hand, it is hard to find experimentally determined values to describe different protein-folding and protein-ligand poses, therefore, classification models should be used to solve the problem.

Considering ML classification models, logistic regression, decision tree, support vector machine, and random forest classifiers are basic models. Here, brief introductions about those methods will be discussed.[111]

1.4 Logistic regression

In statistics, logistic regression models are usually used to calculate the probability of a certain event existing. This method is based on linear regression and used the sigmoid function to transform the continuous probabilities to binary classes. In general, for a linear regression model, if considering the input features as (x_1, x_2, x_3, ..., x_m), the probability p can be calculated as following:

$$p(x) = \theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_m x_m \quad (1.1)$$

where $\theta_0, \theta_1, \ldots, \theta_m$ are coefficients before each input element. In linear regression, probability p has continuous values in the range of $(-\infty, +\infty)$. However, in logistic regression, the predicted value is expected to be 0 or 1. Hence, in logistic regression, a sigmoid function is usually needed to transform the continuous probabilities into binary classes. Following is the example of a sigmoid function:

$$g(x) = \frac{1}{1+e^{-x}} \quad (1.2)$$

In logistic regression, the predictions can be calculated by using the following equation:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \quad (1.3)$$

The above equation is usually called the logistic equation or sigmoid equation. **Figure 1.1** shows the shape of the sigmoid equation. It is clear that when $\theta^T x$ is close to $+\infty$, $g(\theta^T x)$ is near 1; and when $\theta^T x$ is close to $-\infty$, $g(\theta^T x)$ is near 0. By using the sigmoid function, the range of predicted values can be changed from $(-\infty, +\infty)$ to $(0, 1)$.

The coefficients θ should be calculated to determine the functional form of $h_\theta(x)$. Usually, with a training set contains a set of inputs and corresponding classes, the parameters can be calculated by maximizing the likelihood between the predicted probabilities and actual classes.

If we assume that:

$$P(y = 1|x;\ \theta) =\ h_\theta(x) \tag{1.4}$$

$$P(y = 0|x;\ \theta) =\ 1 -\ h_\theta(x) \tag{1.5}$$

If considering a training set contains *m* independently generated examples, the likelihood of the parameters can be written as below:

$$L(\theta) =\ \prod_{i=1}^{m} p\left(y^{(i)}\middle|x^{(i)};\ \theta\right) = \prod_{i=1}^{m}(h_\theta\left(x^{(i)}\right))^{y^{(i)}}(1 -\ h_\theta\left(x^{(i)}\right))^{1-\ y^{(i)}} \tag{1.6}$$

Instead of maximizing the likelihood in equation (1.6), it will be easier to maximize the logarithm of the likelihood as below:

$$l(\theta) = logL(\theta) = \sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + (1 - y^{(i)})\log\left(1 - h_\theta\left(x^{(i)}\right)\right) \tag{1.7}$$

There are many methods (for example, gradient ascent, conjugate gradient, BFGS, L-BFGS) that can be used to maximize equation (1.7). Here, the gradient ascent method is discussed as an example. If we assume the learning rate is α, the update of θ will be $\theta =\ \theta +\ \alpha\frac{\partial l(\theta)}{\partial\theta}$. One training example (x, y) was used to show the calculation details as below:

$$l(\theta) =\ y\log h_\theta(x) + (1 - y)\log\left(1 - h_\theta(x)\right) \tag{1.8}$$

$$\frac{\partial l(\theta)}{\partial \theta} = \left(\frac{y}{h_\theta(x)} - \frac{1-y}{1-h_\theta(x)}\right)\frac{\partial h_\theta(x)}{\partial \theta} = \left(\frac{y}{h_\theta(x)} - \frac{1-y}{1-h_\theta(x)}\right) h_\theta(x)\big(1-h_\theta(x)\big)\frac{\partial \theta^T x}{\partial \theta} = (y\big(1 -$$

$$h_\theta(x) - (1-y)h_\theta(x)\big)x = \big(y - h_\theta(x)\big)x \tag{1.9}$$

Here $\frac{\partial h_\theta(x)}{\partial \theta} = h_\theta(x)\big(1-h_\theta(x)\big)\frac{\partial \theta^T x}{\partial \theta}$ has been used, the derivation is in equation (1.10) as below:

$$\frac{\partial h_\theta(x)}{\partial \theta} = \frac{\partial \frac{1}{1+e^{-\theta^T x}}}{\partial \theta} = (-1) \times \frac{1}{\left(1+e^{-\theta^T x}\right)^2} \times e^{-\theta^T x} \times (-1) \times \frac{\partial \theta^T x}{\partial \theta} = \frac{1}{1+e^{-\theta^T x}} \times \frac{e^{-\theta^T x}}{1+e^{-\theta^T x}} \times \frac{\partial \theta^T x}{\partial \theta} =$$

$$h_\theta(x)\big(1-h_\theta(x)\big)\frac{\partial \theta^T x}{\partial \theta} \tag{1.10}$$

Hence, the update of parameters θ should be:

$$\theta = \theta + \alpha\big(y - h_\theta(x)\big)x \tag{11}$$

For one iteration, the parameters θ can be updated once, after several iterations, the parameters will be converged. Then, a logistic regression model will be constructed.

1.5 Decision tree

The decision tree classification model is another important classifier other than logistic regression models. The decision tree algorithm can be used for both regression and classification problems. Based on the training data, decision tree models can learn a series of questions to infer the class labels of different examples. **Figure 1.2** shows an example of a simple decision tree model.

Considering the decision tree model in **Figure 1.2**, the most important challenge is to locate all questions, in other words, how to get the right order of questions (condition) is the most significant problem of constructing a decision tree model. Here, we will go through the decision tree working methodology from first principles to understand the details about how to locate every question in the model. For example, a decision tree model is needed to predict if it is good to go out for jogging or not. First of all, a collection contains weather and jogging information can be obtained in **Table 1.1**.

Considering **Table 1.1**, every feature (includes weather, temperature, humidity, and wind) can be used to decide jogging or not. The order of the features to be used is needed to construct a decision tree for predicting the jogging decision. For example, should humidity or weather to be considered first? Which feature is the most insignificant one to make the decision? Here, the feature humidity is used as an example of calculations to determine the order of features. **Table 1.2** contains the relationship between humidity and jogging decisions. Here, three values (Chi-square automatic interaction detection, information gain, Gini) can be calculated based on **Table 1.2**, and by using those three values, the order of features will be identified.

(1) Chi-square automatic interaction detection (CHAID)

Chi-square and degrees of freedom can be calculated using the following equations:

$$Chi - square = \sum \frac{(0-E)^2}{E} = \frac{(-0.5)^2}{3.5} + \frac{(0.5)^2}{3.5} + \frac{(1.5)^2}{3.5} + \frac{(-1.5)^2}{3.5} = 1.4 \quad (1.12)$$

$$degrees\ of\ freedom = (r-1) \times (c-1) = (2-1) \times (2-1) = 1 \quad (1.13)$$

In equation (1.13), r represents the number of row components in **Table 1.2**, c is the number of response variables.

With Chi-square and degrees of freedom, the p-value (the right-tailed probability of the chi-square distribution) can be calculated with the function called "CHIDIST" in EXCEL. Here, the calculated p-value is 0.237.

Similarly, p-values for each feature will be calculated, based on the calculated p-values, the order of features can be obtained, the best feature is the one with the lowest p-value.

(2) Information gain

Based on **Table 1.2**, entropy can be calculated using the following equation.

$$Entropy = -\sum p \times log_2 p \quad (1.14)$$

The concept entropy came from information theory, it represents the impurity in data. Entropy values are in the range of [0, 1]. As an example, the total entropy of humidity can be obtained with the equation as below:

$$Entropy_{Total} = -\frac{7}{14} \times log_2\left(\frac{7}{14}\right) - \frac{7}{14} \times log_2\left(\frac{7}{14}\right) = 1 \quad (1.15)$$

Entropy values for high humidity and normal humidity can be calculated with equation (1.16) and (1.17) as following:

$$Entropy_{High} = -\frac{3}{8} \times log_2\left(\frac{3}{8}\right) - \frac{5}{8} \times log_2\left(\frac{5}{8}\right) = 0.9544 \quad (1.16)$$

$$Entropy_{Normal} = -\frac{4}{6} \times log_2\left(\frac{4}{6}\right) - \frac{2}{6} \times log_2\left(\frac{2}{6}\right) = 0.9183 \quad (1.17)$$

With the three entropy values above, information gain can be generated as following:

$$Information\ gain = Entropy_{Total} - \frac{7}{14} \times Entropy_{High} - \frac{7}{14} \times Entropy_{Normal}$$

$$= 0.06365 \quad (1.18)$$

In equation (1.18), information gain represents the reduction in entropy if the feature humidity is split to “high humidity” and “normal humidity”. The goal of the decision tree is that, by splitting data, the resultant node only contains examples from a specific class. Hence, the feature with the largest information gain should be the most important feature.

Similarly, the values of information gain for each feature can be calculated. Based on this information gain values, the order of features will be determined, the feature with the largest information gain is the best feature among all features.

(3) Gini

The value of Gini represents the degree of misclassification, it works similar to entropy but can be calculated faster. Equation (1.19) is the general way to obtain the value of Gini.

$$Gini = 1 - \sum_i p_i^2 \tag{1.19}$$

Here, the humidity is taken as an example of calculating the corresponding Gini value.

$$Gini_{High} = 1 - \left(\frac{3}{8}\right)^2 - \left(\frac{5}{8}\right)^2 = 0.4688 \tag{1.20}$$

$$Gini_{Normal} = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = 0.4444 \tag{1.21}$$

Based on $Gini_{High}$ and $Gini_{Normal}$, the value of expected Gini can be calculated with the following equation:

$$Expected\ Gini = \frac{8}{14} \times 0.4688 + \frac{6}{14} \times 0.4444 = 0.4583 \tag{1.22}$$

With the calculation details discussed above, the values of expected Gini can be obtained for other features, based on those values, the order of features can be generated. The best feature is the one with the lowest expected Gini.

Based on the discussions above, one of CHAID, information gain, and expected Gini can be used to obtain the order of features. Then, the decision tree classifier will be generated with the order of features.

1.6 Support vector machine

Besides logistic regression and decision tree classifier, another important classification algorithm is the support vector machine (SVM). In general, the SVM method is trying to construct hyperplanes by maximizing the boundaries between different types of data points, and those hyperplanes can create sub regions with the most homogeneous points. Based on working algorithms, support vector machines can be classified into three major methods: maximum margin classifier, support vector classifier, and support vector machine.

(1) Maximum margin classifier

If considering a dataset which contains two categories of examples, and those examples can be separated by using a hyperplane. As we know, there will be an infinite number of hyperplanes can be used to separate the data set. The most challenge question is, which hyperplane is the best one to be used? The maximum margin classifier answers question that the hyperplane with the maximum margin of separation width is the best.

If considering a dataset contains n training examples, $x_1, x_2, x_3, \dots, x_n$ (each x is a column vector and contains p elements), with corresponding labels, $y_1, y_2, y_3, \dots, y_n \in \{-1, 1\}$. Then, the hyperplane defined by the maximum margin classifier is the solution to the following optimization problem:

$$\max_{\beta_0, \beta_1, \dots, \beta_p} M \tag{1.23}$$

$$constraint\ 1\text{: } \sum_{i=0}^{p} \beta_i^2 = 1 \tag{1.24}$$

$$constraint\ 2: y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \ \forall\ i = 1, 2, \ldots, n \quad (1.25)$$

$$hyperplane: \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0 \quad (1.26)$$

In equation (1.23), *M* is the width of the margin. Equation (1.25) represents the distance between an observation and the hyperplane. Equation (1.26) shows the form of the hyperplane. With the two constraints showed in equation (1.24) and (1.25), the hyperplane obtained will be the one with the maximum margin.

Figure 1.3 shows the comparison between general hyperplanes which can be used to separate a data set and the hyperplane determined by the maximum margin classifier. The three points with a vector showed in **Figure 1.3 (b)** are called "support vectors". The hyperplane determined with the maximum margin classifier can be obtained only with those three support vectors. If the support vector changes, the function of the hyperplane will change. On the other hand, if examples other than support vectors change, the hyperplane will stay the same.

(2) Support vector classifier

A maximum margin classifier can only be used if there is at least one hyperplane that can separate the whole data set. However, there might be cases cannot be separated by a hyperplane. For those cases, a support vector classifier can be used instead of a maximum margin classifier. A support vector classifier works similarly to a maximum margin classifier, but they allow some observations in the margin area or on the wrong side of the hyperplane. The support vector classifier sacrifices some observations to guarantee the majority of data points are on the right side of the hyperplane.

In general, the hyperplane determined by a support vector classifier is the solution of the optimization problem as below:

$$\max_{\beta_0,\beta_1,\ldots,\beta_p} M \tag{1.27}$$

$$constraint\ 1\colon \sum_{i=0}^{p} \beta_i^2 = 1 \tag{1.28}$$

$$constraint\ 2\colon y_i\left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\right) \geq M(1-\epsilon_i)\ \forall\ i = 1, \ldots, n \tag{1.29}$$

$$constraint\ 3\colon \epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \tag{1.30}$$

$$hyperplane\colon \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0 \tag{1.31}$$

Here, equation (1.27), (1.28), and (1.31) are the same as the maximum margin classifier. The difference between a support vector classifier and a maximum margin classifier is the second and the third constraints. In a maximum margin classifier, the second constraint showed in equation (1.25) requires every observation to be on the right side of the margin area. In support vector classifier, the second constraint (equation (1.29)) allows some observations in the margin area or on the wrong side of the hyperplane. Variable ϵ controls the positions of observations, and variable C controls the total number of observations that are on the wrong side of margin or hyperplane. For example, if $\epsilon_i = 0$, the corresponding point will be on the right side of the margin; if $\epsilon_i > 0\ and\ \epsilon_i \leq 1$, the corresponding point will be on the wrong side of the margin and the right side of the hyperplane, in other words, the observation has violated the margin; if $\epsilon_i > 1$, the corresponding example will be on the wrong side of the hyperplane. By tuning variable C in equation (1.30), we can control the number and severity of violations that the model tolerates. If $C = 0$, the support vector classifier is the same as a maximum margin classifier; if C increases, more observations will be allowed to violate the margin, and the model will become more flexible.

(3) Support vector machine

Support vector classifier and maximum margin classifier require that there is at least one hyperplane can be used to separate the majority of the data set. However, there are data sets that cannot be separated by using a hyperplane. In order to separate those data sets, a support vector machine is needed.

Figure 1.4 is an example of the support vector machine. In **Figure 1.4 (a)**, the data points in a dataset are linearly distributed, all of the points are one dimensional points. It is obvious that there is no hyperplane can be used to separate the two classes. Hence, a maximum margin classifier and support vector classifier cannot be used in this case. By using a support vector machine, a concept called "kernel function" is used. In the specific case shown in **Figure 1.4 (b)**, values of x_1^2 are calculated to transfer those one dimensional data points to two dimensional. With the two dimensional points, a support vector classifier can be used to build up a hyperplane to separate the two classes.

The general idea of support vector machine is to include more features to make the whole dataset in a higher dimension, then the support vector classifier can be used to separate those higher dimensional points. The challenge in a support vector machine is to find the best kernel function. There are four popular kernel functions, linear kernel, polynomial kernel, radial basis function (RBF) / Gaussian kernel, and sigmoid kernel functions. The polynomial and RBF kernel functions are popular choices.

1.7 The random forest algorithm

Random forest (RF) model is an ensemble learning method, the algorithm can be used for classification, regression, and other tasks. A large number of decision trees were constructed to build up the RF model. The prediction from the RF model can be calculated as the vote result from all individual decision trees (classification) or the mean value of the predictions from all decision trees (regression).

The first algorithm of RF was created by Tin Kam Ho,[122] then, Leo Breiman[116] and Adele Cutler[123] developed the extension of the algorithm and registered "Random Forests" as a trademark. RF model is built on single decision trees, to understand why the RF algorithm needs to be developed, it is necessary to understand the limitations of a single decision tree model. As discussed in section 1.5, the decision tree model can be constructed by using CHAID, information gain, and Gini calculations. Based on those values, the order (importance) of each feature can be calculated. However, the decision tree model has a high risk of overfitting. If considering a whole data set be randomly split as two subsets (80% training, 20% testing), and two decision tree models are trained on the two different training sets. The order of features can be totally different for those two models. This is because the examples in the training data can strongly affect the importance (values of CHAID, information gain, and Gini) of each feature. In order to make the decision tree model with a lower variance, a bagging strategy is first used. Here, the bagging classifier is introduced.

Bagging is referred to as bootstrap aggregation. It usually repeats training with the replacement of examples and performs aggregation of the result. It is a general methodology to reduce the variance

in a model. **Figure 1.5** is an example of bagging. For example, the whole data set contains six observations (in real cases, the number will be much larger). Instead of fitting a decision tree model based on all six examples, the bagging algorithm selected subsets of the whole data set. In **Figure 1.5** there are two subsets selected (orange and blue). Based on the selected examples, two decision trees can be constructed to reduce the variance of the decision tree model built upon all examples. Theoretically, this procedure should reduce the variance value. However, this algorithm cannot reduce the variance efficiently. For all subsets selected by the bagging algorithm, they contain all features (columns). This might make the decision trees constructed on subsets correlated with each other. And those correlated trees might not be good enough to solve the overfitting problem from the decision tree model.

RF model was developed to solve the problem that decision trees built on subsets might be correlated. **Figure 1.6** shows an example of the RF algorithm. In the bagging algorithm, decision trees built upon different subsets tend to be correlated because every subset contains all existed features. In order to reduce the correlation between decision trees, the features considered in each subset should be different. In the RF algorithm, the RF model randomly selected some features in each subset to construct decision trees, and the correlation between those trees will be reduced. Hence, with the two dimensional randomness of selecting examples and features, the RF algorithm can effectively reduce the risk of overfitting.

CHAPTER 2: METHOD

2.1 From potential functions to descriptors

The general protocols for calculating descriptors for protein-folding and protein-ligand system are similar. Here, protein-folding system is used as an example to describe how to calculate the descriptors based on potential functions.

If all independent pair wise probabilities with different magnitudes in an *n*-body system are known, the probability of the whole *n*-particle system can be obtained as:

$$p_n = \prod_{i,j=1,\, i\neq j}^{n} c_{ij} * p_{ij}, \quad (2.1)$$

where p_n is the probability of the *n*-particle system, c_{ij} is the scaling factor, which can be evaluated using the random forest model, of pair wise probability p_{ij}, *i* and *j* represent two different particles. Using a knowledge-based potential with pair wise independent interactions, the independent pair wise probabilities for the bond, angle, torsion, and non-bonding terms can be obtained. If the protein structure is treated as a *n*-particle system, the probability is:

$$p_{protein} = (\prod_{bond} c_{ij} * p_{ij})(\prod_{angle} c_{kl} * p_{kl})(\prod_{torsion} c_{mn} * p_{mn})(\prod_{nonbond} c_{pq} * p_{pq}) \quad (2.2)$$

$p_{protein}$ is the protein structure probability, $c_{\alpha\beta}$ and $p_{\alpha\beta}$ represent the scaling factor and the probability of atom pair α and β, the subscripts *ij, kl, mn,* and *pq* correspond to bond, angle, torsion, and non-bonded atom pairs, respectively. In this work, we make two further assumptions: (i) $\prod_{bond} p_{bond}$ and $\prod_{angle} p_{angle}$ are similar for native and all decoys, hence, the product of those

two values is treated as a constant *C*; (ii) the probabilities for the torsion and nonbond atom pairs are independent, since a reference state is used to remove contributions from the ideal-gas state. With these assumptions, the probability of a *n*-atom protein can be written as:

$$p_{protein} = C(\prod_{torsion} c_{mn} * p_{mn})(\prod_{nonbond} c_{pq} * p_{pq}) \quad (2.3)$$

Taking the logarithm on both sides of equation (2.3) we get:

$$log(p_{protein}) = log(C) + \sum_{torsion} x_{mn} * log(p_{mn}) + \sum_{nonbond} x_{pq} * log(p_{pq}) \quad (2.4)$$

where x_{mn} and x_{pq} are the logarithm of c_{mn} and c_{pq}, respectively. A detailed potential database, KECSA2, was utilized to obtain p_{mn} and p_{pq}. Below we use O-MET-CG-MET as an example for what is involved in calculating the pair wise probability of a given protein. From KECSA2, the probability *versus* distance function, shown as a red curve in **Figure 2.1**, can be found. If the distance between O-MET-CG-MET in the protein is 4.5 Å, we first obtain the corresponding probabilities for the distances from 4 Å to 5 Å with an interval of 0.005 Å. Next, we take the logarithm of the average of the 201 probabilities obtained in the previous step, and use it to represent the probability at distance 4.5 Å.

Equation (2.5) shows a general way to obtain the probability of atom pair A-B with distance r_1, where $KECSA2_{A\text{-}B}$ is the potential function of atom pair A-B obtained from the KECSA2 potential data base, r_{ABi} is a distance between r_1-0.5 and r_1+0.5 with an interval of 0.005 Å.

$$p_{A-B}(r_1) = \log\left[\sum_{r_1-0.5}^{r_1+0.5} KECSA2_{A-B}(r_{ABi})\right] - \log(201) \quad (2.5)$$

Using equation (2.5), the probability for each atom pair present in the protein can be calculated; for the same atom pairs, the probabilities were summed yielding the final probability. In this way, the probability list for each protein examined can be generated.

2.2 Random Forest model

A traditional native structure recognition problem is detecting the native or most native-like structure from a collection of decoys. Many different scoring functions have been described[2, 10-40, 47-53] that attempt to address this problem. At first glance, it is hard to use unbalanced decoy sets as the training data set directly. Yet a balanced data structure can be generated if the decoy set is replaced with a 'comparison' data set. Instead of training ML model that focuses on directly finding the native structure from hundreds of decoys, we can create a ML model that can accurately distinguish between native and decoy structures.

In a 'comparison' data set for decoy detection native structure should have the highest probability, which means:

$$log(p_{native}) - log(p_{decoy}) > 0$$
$$log(p_{decoy}) - log(p_{native}) < 0 \quad (2.6)$$

Figure 2.2 shows a detailed workflow for our protocol. If a decoy set consists of one native and *m* decoys, for each structure, an atom pair wise descriptor (probability) can be built as described above. For each descriptor set, there are 16029 elements in total (KECSA2 has 2001 torsion atom pairs and 14028 nonbonded atom pairs, yielding 16029 = 2001+14028.). The descriptor sets are defined as the 'Descriptor vector' in **Figure 2**.**2** Next, the descriptor vector of the native minus the vector of each decoy are classified as class '1', which means 'more stable than' since the native structure is always more stable than the decoys; the descriptor vector of each decoy minus the

vector of the native is defined as class '0', which represents 'less stable than'. The resultant descriptors are described as the 'final descriptor vector' in **Figure 2.2** In this way, equal members of class '0' and class '1' can be generated, which is an ideal situation for classification. Hence, a RF model can be obtained based on using those two classes. Through the use of this classification system a RF model can be generated where the relative probabilities of two proteins with the same sequence can be compared. A final descriptor vector can be generated using the descriptor vector of the first protein minus the second's. Then the RF model can be used to predict the class for that final descriptor vector. If the prediction from the RF model is '1', it means the first protein is 'more stable than' the second one, and if the prediction is '0', the first protein is 'less stable than' the second one.

Constructing a RF model that can accurately differentiate native and decoy structures is not enough. For a native recognition blind test, in order to identify the native structure, a ranking of all structures should be generated. Thus, the RF model needs to be used to obtain the ranking list for a decoy set.

Figure 2.3 gives the protocol used to obtain the ranking of a decoy set with *n* structures. First, the probability descriptor of each protein structure can be built using the KECSA2 database. Second, a table for each structure was obtained from the probability descriptor of the individual protein structure minus the probability vectors of all the other structures. Then, the RF model is used to predict the class of each column in all tables; in other words, the RF model is used to 'compare' two structures. Finally, a row with length *n-1* can be generated for each structure. The value of each column in the resultant row is either '0' or '1', which represents the comparison result of each

structure with all other structures. The sum of the resultant row is defined as a ‘score’, which indicates if the corresponding structure is more stable than the “score” amount of decoys. In this way, the score of each structure can be generated, thereby, creating a ranking list.

2.3 Decoy sets

The decoy sets we used for protein-folding systems include the multiple decoy sets from the Decoys 'R' Us collection (http://compbio.buffalo.edu/dd/download.shtml), which include the 4state_reduced, fisa, fisa_casp3, hg_structal, ig_structal, ig_structal_hires, lattice_ssfit, lmds, and lmds_v2 decoy sets. The MOULDER decoy set was downloaded from https://salilab.org/decoys/; the I-TASSER decoy set-II was obtained from https://zhanglab.ccmb.med.umich.edu/decoys/decoy2.html; and the ROSETTA all-atom decoy set from https://zenodo.org/record/48780#.WvtCA63MzLF.

Our RF model for protein-folding systems was compared to the following potentials designed for decoy detection: KECSA2, GOAP,[2] DFIRE,[40] dDFIRE,[37, 43] and RWplus.[39] The programs for these methods were downloaded from the corresponding author's website.

For protein-ligand systems, 191 systems were selected out of the 195 systems in CASF-2013[112] due to formatting issues with our program. CASF-2013[112] is known as the 'Comparative Assessment of Scoring Functions', it includes data sets for testing the scoring, docking, screening, and ranking powers of scoring functions. Here, we only used the data sets, which were designed to test the docking power of scoring functions. The decoy ligand binding poses were prepared with three popular molecular docking programs: GOLD(v5), Surflex-Dock implemented in SYBYL(v8.1), and the docking module built in MOE(v2011). These three programs have different algorithms for ligand pose sampling, therefore, the resultant decoy set is more complete and avoids

the bias inherent in using only one program. In total, we used 191 protein ligand systems, 15802 ligand decoy poses, and 31604 native-decoy comparisons.

2.4 Structure preparation

All protein structures (including both native and decoys) for protein-folding systems were converted into their biological oligomerization state and prepared with the Protein Preparation Wizard, REF which adds missing atoms, optimizes the H-bond network, and performs energy minimization to clean up the structures for subsequent calculations. The decoy sets can be found here https://github.com/JunPei000/protein_folding-decoy-set.

Ligand pose structures are directly obtained from CASF-2013[112] for protein-ligand systems.

2.5 Potential functions

KECSA2 potential function set was used for the first project about protein-folding pose selection. KECSA2 is developed based on KECSA potential function set. KECSA REF is a potential data base originally designed for protein-ligand systems, we applied the same methodology used to derive KECSA to protein structures to generate KECSA2, a potential data base only for protein systems. The detailed derivation and parameters for KECSA can be found in reference 91. PDBbind v2014[113] was used as the protein crystal structure source, two criteria were used to filter these structures, (1) protein structures with resolution better than 2.5 Å were selected; (2) Metal ions and residues within 4 Å around the metal ions were deleted. After filtering, 9606 protein crystal structure were selected as the protein structure source. A detailed atom type definition was used in the KECSA2 potential; in other words, every atom type represents a specific heavy atom in the twenty naturally occurring amino acids. For instance, 'CA_ALA' corresponds to the alpha carbon in alanine. In total, there are 167 atom types in KECSA2. The methodology of KECSA was used to construct the reference state and remove ideal gas contributions. Finally, 2001 torsion and 14028 nonbond atom pairwise interactions were generated. The follow function was used to describe the nonbond interactions between each atom pair:

$$E_{AB}(r_i) = \varepsilon_1 \left(\frac{\sigma}{r_i}\right)^{\alpha} - \varepsilon_2 \left(\frac{\sigma}{r_i}\right)^{\beta} \quad (2.7)$$

The five parameters in equation (1), ε_1, ε_2, σ, α and β for each atom pair in KECSA2 can be found at https://github.com/JunPei000/protein_folding-decoy-set.

For protein-ligand systems, we used the GARF[114] potential to calculate the pairwise probabilities for each protein ligand complex. GARF is a potential database developed by our group. It employed a graphical-model-based approach with Bayesian field theory to construct atom pairwise potential functions. There are 20 atom types for the protein atoms and 24 atom types for the ligands. All definitions of the atom types are listed in **Table 2.1**. Further details regarding GARF can be found in the original article.[114]

The force field parameter sets ff94[120] and ff14SB[121] from Amber were also used for protein-folding pose selection. For ff94, atom types with their corresponding charges were obtained from file "all_amino94.lib" in directory "$AMBERHOME/dat/leap/lib/". R_{min}, ε, and torsion related parameters (V_n , n , and γ) were from file "parm94.dat" in directory "$AMBERHOME/dat/leap/parm/". For ff14SB, atom types with corresponding charge values were obtained from file "amino12.lib" in directory "$AMBERHOME/dat/leap/lib/". R_{min} and ε values for each atom type were from file "parm10.dat", in directory "$AMBERHOME/dat/leap/parm/". Torsion related parameters (V_n, n, and γ) were gained from file "frcmod.ff14SB", in directory "$AMBERHOME/dat/leap/parm/".

2.6 Machine learning and validation

The sklearn.ensemble.RandomForestClassifier function from Scikit-learn was used to create the proposed classification model.[115] One training-testing iteration includes: (1) Randomly split the whole data set into two parts, 80% as the training data set and 20% as the test set. (2) A grid search with five-fold cross validation was performed on the training set in order to identify the best set of hyperparameters for the RF model. (3) The RF model with the best set of hyperparameters was then validated on the test set. Although the data set is randomly split as training and testing sets, there is still a bias buried in the splitting procedure. In this work, ten independent iterations were performed on the combined decoy set in order to avoid bias from data partitioning scheme.

Accuracy, a typical evaluation for ML classifiers, was used to evaluate the performance of RF models. An accuracy value can be calculated based on a "confusion matrix", which is usually used in the supervised ML field. The general format of a confusion matrix is presented as **Table 2.2**.

There are four values in a confusion matrix, which are True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). TPs refers to the cases whose predicted classes are class 1 - same as their actual classes. FPs are the cases predicted as class 1 whereas their actual class is class 0. FNs represent cases whose predicted class are class 0, however, their actual class is class 1. TNs represent the cases where the predicted class is class 0, which is the same as their actual class. Accuracy can be calculated based on these four numbers from the confusion matrix using:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.8}$$

In many cases, accuracy cannot be used to judge the performance of a ML classifier. For example, if considering a data set with 90 positive samples and 10 negative ones, a naïve classifier will predict all samples as positive. At the same time, the accuracy of that naïve classifier is 0.90. However, it is obvious that the naïve classifier is not able to provide reliable predictions. In this work, accuracy can be selected to represent the performance of RF models due to the fact that, the data base is evenly distributed (the numbers of positive and negative samples are the same). On the other hand, ten accuracy values can be obtained from ten independent RF models built with different data partitioning schemes, the highest, lowest, and averaged accuracy values were used to represent the general performance of RF models.

CHAPTER 3: RANDOM FOREST MODEL WITH KECSA2 FOR PROTEIN-FOLDING POSE SELECTION

3.1 Accuracy of individual decoy set training

The most important characteristic of the resultant scoring function is its ability to differentiate the native structure from decoys. **Table 3.1** shows the accuracy for both the RF models and traditional scoring functions. Since ten cycles of independent training and testing were performed for each decoy set, the highest, lowest, and averaged accuracy were used to represent the general performance of RF models on that specific decoy set. In this way, the performance of the RF model can be better interpreted. In general, the RF model shows higher accuracies than all traditional methods for all of the decoy sets. For some decoy sets, like fisa, ig_structal, lmds_v2, and rosetta, RF models significantly improved the averaged accuracy to nearly 1.000, and the lowest accuracy values are still higher than the best accuracies of the other scoring functions. For the other decoy sets, such as 4state_reudeced, fisa_casp3, hg_structal, ig_structal_hires, I-TASSER, lattice_ssfit, lmds, and MOULDER, the averaged accuracies from the RF models are similar to the best accuracies of traditional scoring functions, while the lowest accuracies of the RF models are similar to the accuracies of other methods. Overall, the RF models show better performance.

3.2 Native ranking of individual decoy set training

Although the accuracies of RF models are higher than other methods, we still wanted to further validate their performance. Other than accuracy, another important criteria for judging a model/scoring function is whether the model/scoring function identifies the native structure as having the lowest rank. Hence, native structure ranking from the different methods were also compared. **Table 3.2** shows the rankings of the native structures from several different models. The highest, lowest, and averaged rankings are shown to assess the performance of RF models. For the decoy sets fisa, ig_structal, lmds, lmds_v2, and ROSETTA, the RF models substantially improves native structure ranking over the other models. In the remain decoy sets, the averaged rankings of native structures are similar to the best performance of the other scoring functions. It can be concluded that, in general, the RF model shows a better performance in ranking the native structure over other methods we tested.

3.3 1st decoy RMSD and TM-score of individual decoy set training

Although the ability to recognize the native structure as the most stable structure is a crucial characteristic of a good model/potential. For a model/potential to be useful for guiding conformation sampling, it should have a good correlation with structural quality. The RMSD and TM-score were used as two criteria for assessing the quality of each decoy structure. RMSD is the root mean squared deviation of all Cα pairs of the decoy to the native structure. TM-score[47] gives a large distance a small weight and makes the magnitude of TM-score more sensitive to the topology. **Table 3.3** and **Table 3.4** summarize the results of best model selection of different methods.

Table 3.3 shows the 1st decoy's RMSD of RF models and against a range of available scoring functions. The RMSD values of available methods are generally within the range of lowest and highest RMSD values of the RF models for each decoy set. This means the performance of those traditional scoring functions are within the confident range of our RF models. **Table 3.4** shows the 1st decoy's TM-score for the RF model and against several models; these results are similar to what we observed for the RMSD analysis. In each decoy set, the 1st decoy's TM-score is within the range of the lowest and the highest TM-score from the RF models. Considering that the independent training and testing process was done ten times, the range of lowest to highest RMSD/TM-score values show the confidence range of RF models for each decoy set. In general, the RMSD/TM-score performance of available models are within the confidence range of the RF models, and the averaged values are similar in RF models and models we tested against. In other

words, the performance of the RF models against a range of models, when it comes to selecting the best decoy structure, were similar.

3.4 Feature importance analysis for the overall decoy set

It is important to understand whether the better performances of RF models are due to overfitting because a large number of descriptors were used. To this end, a feature importance analysis was performed and is shown in **Figure 3.1**. Based on the analysis, new RF models using only the top 500 features were constructed using the previous procedure. **Table 3.5** shows the comparison of the accuracies between using only the top 500 and all 16029 features. In general, the highest, lowest, and averaged accuracy values of the RF models using the top 500 features for each decoy set are similar to the corresponding values of the RF models using all features. Hence, we conclude that the better performance of RF models with all features is not simply due to overfitting and the current method is robust even in the face of potentially non-essential features.

3.5 Comparison of overall RF model with traditional scoring functions

Besides creating RF models for each individual decoy set, combined RF models using all decoy sets were also constructed. This examines the situation where in a study one might generate decoys using one method and then score them with another. There were 291 individual systems across the 12 decoy sets that were combined finally, yielding 235 different protein systems (several proteins overlapped amongst the decoy sets). In these studies, 80% of the combined data set was used as the training data to build the RF models instead of choosing several specific decoy sets (like 4state_redueced, fisa, etc.). This was done to insure that the training and testing data set covered the same feature space and had the same distribution – this is known as an independent and identical distribution (IID).[116] The feature space and distribution of decoys from different decoy sets are different because different models were used to generate those structures.

Table 3.6 shows the result of comparing the overall performance of RF models with a number of available potentials. Due to the large number of descriptors, it is impossible to obtain RF models using the entire 16029 feature set. Based on the importance analysis discussed previously, instead of using all features, top 100 and 500 features were used to build up the overall RF models on the combined decoy sets. First, all RF models with different importance features provide higher averaged accuracy values than other traditional scoring functions. Clearly, the accuracies of the RF models outperform the other conventional methods. Second, the highest rankings of the native structure from RF models are smaller than the rankings of other methods, and all of the averaged rankings of the RF models were ~10 or less, which means the RF models can identify the native structure within the top ten structures. Hence, the RF models outperform other methods on this

task. Finally, the RMSD and TM-score values of conventional potentials are within the corresponding confidence range of the RF models, and those values are similar to the averaged RMSDs and TM-scores of the RF models. Both the RMSD and TM-score results suggest that the performance of the RF models is similar to other conventional potentials.

3.6 Importance of potential

Based on the previous discussion, it is clear that the RF models with KECSA2 perform the best in accuracy and ranking both on individual and overall decoy sets. This directly leads to the interesting question: does the potential plays a significant role in RF models? Here, two analyses were done to address the importance of the KECSA2 potential in the performance of the RF model. First, the probability functions of top 100 and 500 features (atom pairs) were scrambled to test if the probability functions played a role in the RF model, for example, after scrambling, the probability function of atom pair O-PRO and N-ALA might be changed to the probability function of CA-GLY and C-THR. If the KECSA2 potential plays a role in RF models, the performance of the scrambled probability functions should be worse than KECSA2. The peak positions in the probability functions represent the most favorable distances between different atom pairs found in the experimental structure database. For example, the peak position of the atom pair O-PRO and N-ALA is 3.04 Å, which means those two atoms are most stable when they form a hydrogen bond at that distance. However, after scrambling, the peak position might change to 4.51 Å (peak position of CA-GLY and C-THR), which no longer represents a hydrogen bond. Thus, the scrambled probability function suggests that the atom pair O-PRO and N-ALA is most stable when they do not form a hydrogen bond. It is clear that the scrambled probability functions are unphysical. We expect that it would be unlikely that the RF model, as employed herein, could correct these deficiencies so, the performance of the scrambled probability function is expected to be worse than original KECSA2.

Second, the uniform probability functions (or potential functions) were built for the top 100 and 500 atom pairs to test if the KECSA2 probability peak heights (or well-depths) are important in RF models. The uniform probability functions have the same peak positions found in KECSA2, but with same heights. By doing so, the interaction strength 'bias' of different atom types from KECSA2 can be eliminated via use of uniform probability functions. If the KECSA2 probability peak heights (or interaction potentials) are significant, the performance of uniform potential should be worse than KECSA2.

The comparison of the result between the original KECSA2 potential, scrambled potential, and uniform potential are shown in **Table 7**. From the comparison between the original KECSA2 and the scrambled potentials, we find the accuracy of the models decreased by ~0.15, which gives a clear signal that the full KECSA2 potential (well depth and energy minimum) plays a role in RF models. The comparison between the uniform and original KECSA2 potential gives an evidence of how important the r_{max} component of the KECSA2 potential is in building an effective model. For the RF models with the top 100 features, the averaged, highest, and lowest accuracies based on the original KECSA2 potential are slightly higher than the corresponding accuracies from the RF models based on uniform potentials. However, if the number of features is increased to 500, the averaged, highest, and lowest accuracies from the RF models based on the original KECSA2 are similar to the uniform potentials. This provides strong evidence that only peak positions in the probability functions are critical in building up RF models for native protein structure detection. More importantly, the result also implies that RF models can be used to tune the height of peaks in probability functions (or the depth of potential functions) only with the information of peak positions in protein structures.

3.7 Conclusion

In this work, we utilized a 'comparison' concept to construct RF models on an unbalanced data set. With these RF models, the knowledge-based potential, KECSA2, was refined via assignment of different importance factors to different atom pairs present in the scoring function. The performance of the resultant RF models were assessed with individual and combined decoy sets and compared with the results from conventional models. We find that the RF models perform better in accuracy and native ranking and have similar performance in the RMSD and TM-score tests. In other words, the RF models improved the effectiveness of finding native structures from a set of decoys, without compromising their ability to find the best decoy structures. This RF model based refinement not only can be used to improve the performance of KECSA2, but it can also be applied to other atom/residue pair based potentials. More importantly, we find that only peak positions in probability functions play a significant role in constructing the RF models. This result implies that, with peak position information, RF models can be created to construct probability functions (or potential functions) by tuning the height of peaks in those functions based on native and decoy protein structures.

CHAPTER 4: RANDOM FOREST MODEL WITH GARF FOR PROTEIN-LIGAND POSE SELECTION

4.1 Accuracy

The most important goal of a scoring function is to accurately identify the native structure among a plethora of decoy structures. In order to evaluate the ability of a scoring function to identify the native structure, the concept of accuracy is used in this work. If a decoy set contains 100 decoy structures but only one native, the scoring function is expected to make 200 correct comparisons to identify the native pose. The higher the accuracy of the comparison, the better the performance of the scoring function. The third column in **Table 4.1** shows the comparison of accuracies from RF models and 29 other scoring functions. The averaged, highest, and lowest accuracy of the RF models are 0.953, 0.969, 0.942. The averaged accuracy value is higher than all of the other tested scoring function, and the lowest accuracy value is still higher than all of the other accuracies. It is clear that the RF models have a higher accuracy, which means that the RF models perform better than all other scoring functions in comparing the ligand native pose to decoy poses.

4.2 Native ranking

Other than accuracy, another criteria for evaluating a scoring function is the ranking of the ligand native pose. In other words, a scoring function is expected to give the ligand native pose the lowest rank. The fourth column in **Table 4.1** shows the result of ligand native pose ranking from each method. The averaged, highest, and lowest ligand native pose ranking from RF models are 4.49, 5.54, and 3.54, respectively. The confidence interval of the native pose's ranking from RF models is [3.54, 5.54]. It is clear that all 29 scoring functions have ligand native rankings higher than the averaged native ranking obtained from the RF models, and of these rankings they are larger than the highest native ranking from the RF models. Thus, it can be concluded that the RF models perform better in selecting the ligand native pose than existing models.

If the accuracy values are compared with the ligand native pose rankings, a correlation between those two sets of data can be found. The higher the accuracy, the lower the native pose ranking. The most important goal of a scoring function is to identify the most stable ligand pose (native pose), therefore, the minimum standard for a scoring function is to correctly compare native pose to decoy ones. Using our previous example of a decoy set containing 100 decoy structures and one native pose we have 200 comparisons between the native and decoy poses. Hence, minimally the scoring function should make 200 correct comparisons to obtain the native structure. With more correct comparisons, the native pose has a higher chance to be found at a lower rank. For example, if the scoring function makes ten mistakes, the accuracy is around 0.95, and the ligand pose ranking would be ≥ 5.

4.3 Random Forest model with decoy comparison information

Our RF model is focused on identifying the native binding pose of a ligand among all decoy poses. However, it is not effective in identifying the best decoy due to the lack of comparison information between the best decoy structure and the other decoy poses. In order to include the comparison information between decoy poses into the RF analysis we made the following assumption. The assumption is that the ligand decoy pose with the lowest RMSD is the most stable decoy structure (best decoy pose). **Figure 4.1** shows the protocol of adding comparisons between the best decoy pose and other decoy poses. For example, a decoy set contains m decoy structures and one native pose, two kinds of comparisons were considered when the model was trained: (1) the comparison between the native binding pose and all other decoy poses, in total there are $2m$ comparisons (m comparisons for each class); (2) without the native binding pose, the best decoy pose was compared with all other decoy poses for a total of $2(m - 1)$ comparisons. Then, RF models, which were trained on these comparisons, were used to select the best decoy through the protocol discussed in chapter 2.

4.4 1st decoy RMSD and TM-score

Besides accuracy and native pose ranking, there is another criteria, RMSD of the best decoy structure, which is used to judge the performance of a scoring function. The best ligand decoy pose refers to the decoy structure that is selected by a scoring function as the structure among all decoy poses most similar to the native pose. Scores generated by a scoring function are expected to be correlated with the quality or native-likeness of a structure. The RMSD value between the ligand native binding pose and a decoy binding pose is often used to represent the quality of that decoy pose. If the RMSD is below a predefined cutoff (RMSD < 2 Å), the decoy binding pose is believed to be "native-like". The last column in **Table 4.1** shows the RMSD values from each of the scoring functions. The averaged, highest, and lowest RMSD values from RF models are 3.87 Å, 4.47 Å, and 3.38 Å, respectively. The confidence interval for the RF models is [3.38, 4.47]. It is clear that there are 26 scoring functions that can identify ligand decoy poses with RMSDs lower than 3.38 Å, and two scoring functions provide RMSD values within the confidence range of the RF models. In general, 28 scoring functions perform better than our initial RF models in selecting the best decoy structure.

The RF models used in **Table 4.1** only contain comparisons between native and decoy poses, while comparison information between decoy poses was not considered when the models were trained. Hence, we conclude, that these RF models do not have enough information to find the "best" ligand decoy poses among a large number of decoy structures. In order to improve our RF models' ability to identify the best decoy structure, comparison information between decoy poses should be added when training the RF models. Here, we make an assumption that, among all decoy poses, the pose

with the lowest RMSD is perhaps the most stable of all the decoys because it is most "native-like". With this assumption, the comparison between the best decoy and other decoy poses could be generated. Instead of just using comparisons between native and all decoy poses, the new training set also included comparisons between the best decoy pose and all other decoy poses. **Table 4.2** shows the result when different number of decoy structures were identified as the most stable poses. Four sets of training data were used: (1) only including comparisons between the native and decoy poses; (2) including comparisons between the native and decoy poses, and between the decoy structure with the lowest RMSD with all other decoy poses; (3) including comparisons between the native and decoy poses, between the two lowest RMSD decoy poses and all other decoy poses; (4) including comparisons between the native and decoy poses and, between the three lowest RMSD decoy poses with all other decoy poses. **Table 4.2** gives the overall performance on accuracy, ligand native pose ranking, and the best decoy RMSD. With the inclusion of decoy structures in the training set, the accuracy of the RF models and the ligand native binding pose's ranking were slightly negatively affected. On the other hand, the best decoy pose's RMSD dropped dramatically. The averaged, highest, and lowest RMSD of the best decoy pose from RF models trained on data set only including comparisons between native and decoy binding poses are 3.87 Å, 4.47 Å, and 3.38 Å, respectively. Alternatively, the corresponding values from RF models including the three lowest RMSD decoy structures are 2.27 Å, 2.44 Å, and 1.73 Å, respectively (confidence interval is [1.73, 2.44]). By including low RMSD decoy structure comparisons we obtain RF models (see **Table 4.1**) that give better first decoy RMSDs than 13 scoring functions, a further 15 scoring functions have first decoy RMSDs with the confidence interval of the RF model and only one scoring function gave a RMSD smaller than 1.73 Å. Hence, we conclude that the

overall performance (*i.e.,* accuracy, native rank, and low RMSD first decoy) of RF models can be improved by including lowest RMSD decoy comparisons in the fitting of the model.

Based on previous discussion, it is clear that with a higher accuracy, a scoring function can give the native binding pose a lower rank. If the accuracy values are compared with the RMSD of the best decoy, it is obvious that those two sets of data do not appear to correlate. Some scoring functions are better at selecting the native pose but provide a relatively larger RMSD value, whereas other scoring functions do a better job selecting the best decoy structure but do not have the ability to identify the native binding pose. This leads to a basic philosophical question: which one is more important, accuracy or RMSD? Both of them should be important in the limit that all decoy poses can be obtained. However, it is almost impossible to generate all relevant decoy poses using contemporary approaches. In our opinion, the basic requirement for a scoring function is that the function can accurately identify the native pose. To some degree, RMSD might be useful in judging if a structure has a low free energy, but it is obvious that a decoy structure can have a high free energy while enjoying a low RMSD value. Hence, if two scoring functions were compared solely on identifying the best decoy and one gives a RMSD larger than 2 Å while another is less than 2 Å, it is unclear, at least to us, how to judge which one is better. On the other hand, accuracy, the factor that represents the performance of a scoring function when comparing native and decoy poses, is a clear standard. The explicit hypothesis we are making when docking and scoring is that the native structure always has a lower free energy than the decoys. When comparing two scoring functions, the better scoring function should be the one with a higher accuracy. Put another way, when creating, for example, ML models for a self-driving car what is more important – accurately

identifying an obstacle or being close to identifying an obstacle? Therefore, we believe that accuracy is the more important criteria.

4.5 Uniform probability function

The RF models perform better than all other scoring function on accuracy and native binding pose ranking. It is interesting to consider if the GARF potential is critical in these RF models. Two tests were set up in order to test the importance of the GARF potential database. First, a scrambled probability function set was constructed based on GARF followed, by a uniform probability function set to test whether GARF's peak position is more important or if the peak height is more critical.

The scrambled probability function set was generated by randomly mixing up the atom pairs in the GARF potential database. Taking the 480 atom pairwise potential functions in GARF we randomly scrambled the atom pair names. For example, before scrambling, one probability function represented the interaction between N and O.co2, while after scrambling, the same probability function might be used to describe the interaction between C and F. Hence, the scrambled probability function set is physically unrealistic. Based on the scrambled probability function set, ten independent RF models were constructed following the same procedure described in the methods section. Since the scrambled function set is physically unrealistic, it is expected that the performance of these RF models would be worse than models using the original GARF potential.

There are two kinds of information embedded in the GARF potential, peak positions (well position) and peak heights (well depth). Which is more important – or are both important? To address this a uniform probability function set was built up to probe this fundamental question.

Uniform probability functions share the same peak positions with the original GARF potential, but set all the peak heights at a constant value eliminating the impact of prior peak heights. If the obtained RF models based on a uniform probability function set performs similarly to models obtained with the original GARF, peak positions will be more important than peak height. Alternatively, if the obtained RF models perform more poorly than original the models peak height is significant.

Table 4.3 compares the accuracy result from RF models based on the original, scrambled, and uniform GARF potential database. If we compare the accuracy values between RF models based on original and scrambled GARF, it is clear that the averaged, highest, and lowest accuracies from RF models with a scrambled probability function perform poorer. The accuracy value did not drop as much as we have seen in the past[51] because the GARF potential only contains intermolecular interactions found in protein ligand systems. Moreover, the 480 peak positions found in GARF are all in the range of [2.5, 5.1] with 355 peak positions in the range of [3.4, 4.4] (see **Table 4.4**). Therefore, the scrambled peak positions in the scrambled probability function set might be similar to the original positions in GARF. It is reasonable to expect that the accuracy of RF model based on scrambled probability function set is lower than the corresponding values from original models. On the other hand, if we compare the accuracy values from the uniform probability function set to the values provided by the original RF models, it is obvious that the averaged accuracy values from those two sets of models are the same. This further supports the notion that peak position is more important than well depths in given a potential function used to build a RF model.[117]

4.6 Influence of training set size

Usually in the field of supervised machine learning, especially when the data set does not contain a large number of data points, it is common to split the data set into training (80% of total, 16% cross validation set, five-fold cross validation in training data) and test sets (20% of total). The 80:20 ratio works well in most cases, but we wanted to test whether the RF models can achieve a similar accuracy with a smaller training set. **Table 4.5** shows the accuracy result from RF models based on the original and uniform GARF data base trained on data sets of differing sizes. **Figure 4.2** is the corresponding plot obtained using the data of **Table 4.5**. The blue and orange lines in **Figure 4.2** represent the performance of RF models based on the original and uniform GARF database, respectively. Both lines show that by increasing the size of the training set, the accuracy of RF models generally increased. Accuracy values converge with training sets >60% and the RF models based on the original and uniform GARF potential have the same trend.

4.7 Conclusion

In this work, we constructed RF models on unbalanced data sets utilizing the 'comparison' concept to identify native protein-ligand poses. Using RF, the GARF potential database was refined by assigning different importance factors to each atom pair in that potential. The resultant RF models were tested on a well-known protein-ligand decoy set, CASF-2013,[5] which includes decoy structures generated from three docking packages using different docking algorithms. The results suggest that our RF models outperformed other scoring functions on accuracy and native binding pose selection. By including comparisons between the best decoy pose and the remaining decoy pose structures, the RMSD value of the best decoy was reduced. We also tested the importance of GARF in creating the corresponding RF models. The use of a scrambled GARF probability function to build a RF model provided evidence for the significance of the GARF potential, while the uniform GARF potential indicated that peak position (or the well position) is most relevant in building a RF model. Finally, we tested the influence of training set size, which showed that the accuracy converged when ~60% of the data set was used in building the RF model. Overall, we showed that potential function based RF models perform at a high level when identifying a native pose from a collection of decoys.

CHAPTER 5: COMBINE RANDOM FOREST WITH AMBER FORCE FIELD FOR PROTEIN-FOLDING POSE SELECTION

5.1 Definitions of atom types, torsion types, and nonbond types

For protein systems, Amber has its unique definition for atom types. It uses three parameters – atomic charge, potential well depth-ε, and van der Waals radii - to assign different atoms in various chemical environments. Atoms sharing the same values for ε and van der Waals radii were defined as one atom type; however, atoms belonging to the same atom type by this criteria might have very different charges due to different chemical environments. For example, the gamma carbon (defined as 'CG2' in pdb format and 'CT' in Amber atom type format) in isoleucine has a charge value of -0.3204, whereas the delta carbon (defined as 'CD' in pdb format and 'CT' in Amber atom type) in proline has a positive charge as 0.0192. Hence, atoms with different charge values but the same Amber atom type necessarily had to separated into different atom types. In this work, charge, ε, and van der Waals radii were used to define different atom types. **Table 5.1** and **Table 5.2** summarizes the names of our atom types, their corresponding values of charge, ε, and van der Waals radii in ff94 and ff14SB force field, respectively. In total, there are 191 detailed atom types for both the ff94 and ff14SB force fields.

Definitions of torsion types in Amber are based on Amber atom types, hence, with detailed atom types, representations of torsion interactions must also be redefined. In Amber, it primarily uses four atoms to define a torsion type, for example, X-CT-CT-X (X represents any atom), represents

torsion interactions between any two atoms connected by two CT atoms. This definition eliminates the total number of torsion interactions, however, it lumps different and unique torsion information together. Here, it is necessary to split the Amber torsion types based on our more detailed atom type definitions. For example, the Amber torsion type “H -N-C- O” (ff94) is split by us into 14 torsion types as “H-0 -N-C- O-0”, “H-0 -N-C- O-1”, “H-0 -N-C- O-3”, “H-0 -N-C- O-5”, “H-1 -N-C- O-0”, “H-1 -N-C- O-1”, “H-1 -N-C- O-3”, “H-1 -N-C- O-5”, “H-4 -N-C- O-2”, “H-5 -N-C- O-0”, “H-5 -N-C- O-1”, “H-5 -N-C- O-3”, “H-5 -N-C- O-5”, and “H-6 -N-C- O-4”. In this way, torsion interactions in different chemical environments can be classified as different torsion types. For instance, torsion type “H-4 -N-C- O-2” represents the torsion interaction between the HD and OD1 atoms in ASN’s sidechain.

Other than torsion angles, nonbond interactions are also redefined based on the available atom types. For example, “C-0_C-1” represents the nonbond interaction between C-0 and C-1 atoms, and so on. Briefly, in this work, for ff94, there are 191 detailed atom types, 1,143 torsion types, and 18,336 nonbond types; for ff14SB, there are 191 detailed atom types, 1,175 torsion types, and 18,336 nonbond types.

5.2 From Amber parameters to descriptors

In Amber, the total energy of a protein structure can be calculated as follows:[118]

$$E_{total} = \sum_{bonds} k_b(r - r_0)^2 + \sum_{angle} k_\theta(\theta - \theta_0)^2 + \sum_{dihedrals} V_n[1 + cos(n\phi - \gamma)] + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left[\frac{\varepsilon R_{min}^{12}}{R_{ij}^{12}} - \frac{2\varepsilon R_{min}^{6}}{R_{ij}^{6}} + \frac{q_i q_j}{\varepsilon R_{ij}}\right] \quad (5.1)$$

Where k_b and k_θ are force constants, r_0 and θ_0 are equilibrium bond lengths and bond angles, respectively. V_n, n, and γ are the torsion barrier, phase, and periodicity, respectively. R_{min} is the sum of van der Waals radii of atoms *i* and *j*. ε is the depth of the potential well for the interaction between atoms *i* and *j*. q_i and q_j are charges on atoms *i* and *j*. R_{ij} is the distance between atom *i* and *j*. Here, we make the first assumption that, in both native and decoy protein structures, all bond and angle interactions are identical. Hence, the total energy of a structure can be simplified to:

$$E_{total} = Cons + \sum_{dihedrals} V_n[1 + cos(n\phi - \gamma)] + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left[\frac{\varepsilon R_{min}^{12}}{R_{ij}^{12}} - \frac{2\varepsilon R_{min}^{6}}{R_{ij}^{6}} + \frac{q_i q_j}{\varepsilon R_{ij}}\right] \quad (5.2)$$

Where *Cons* is the total energy for the bond and angle interactions. With parameters extracted from Amber, the energy of one specific torsion interaction, which is the sum of the dihedral, 1_4 van der Waals, and 1_4 electrostatics energies, can be calculated using equation (3):

$$E_{A-B-C-D} = V_{A-B-C-D}[1 + cos(n_{A-B-C-D}\phi - \gamma_{A-B-C-D})] + \frac{1}{2.0} \times \left[\frac{\varepsilon_{AD} R^{12}_{\min\_AD}}{R^{12}_{AD}} - \frac{2\varepsilon_{AD} R^{6}_{\min\_AD}}{R^{6}_{AD}}\right] + \frac{1}{1.2} \times \frac{q_A q_D}{\varepsilon R_{AD}} \quad (5.3)$$

$$E_{A-B} = \left[\frac{\varepsilon_{AD} R^{12}_{\min\_AD}}{R^{12}_{AD}} - \frac{2\varepsilon_{AD} R^{6}_{\min\_AD}}{R^{6}_{AD}}\right] + \frac{q_A q_D}{\varepsilon R_{AD}} \quad (5.4)$$

$$E_{A-B-C-D} = V_{A-B-C-D}[1 + cos(n_{A-B-C-D}\phi - \gamma_{A-B-C-D})] \quad (5.5)$$

In the Amber software package, three parts of equation (3) were calculated separately, and belonged to different energy components, $E_{dihedral}$, $E_{1\text{-}4\text{-}vdW}$, and $E_{1\text{-}4\text{-}EEL}$, respectively. $E_{dihedral}$, $E_{1\text{-}4\text{-}vdW}$, and $E_{1\text{-}4\text{-}EEL}$ are energies of dihedral, van der Waals, and electrostatics between terminal atoms (for example, atom A and D in torsion A-B-C-D), respectively. Here, instead of calculating energies for torsions separately (using three energy components), only one value was generated to represent a torsion interaction based on equation (5.3). In equation (5.3), 2.0 and 1.2 are scale factors for the energies of the 1-4 van der Waals and electrostatics interactions.[118] Similarly, equation (5.4) is used to obtain the pair wise nonbond energies, which include energies for both van der Waals and electrostatics. Other than general torsion and nonbond interactions described in equation (5.3) and (5.4), there is another torsion interaction considered in Amber. Out-of-plane terms, also referred as improper torsions, represent "branched" four atoms systems. In these branched systems, there are three bonds between the central atom and other three atoms, and the central atom is forced into the plane of the other three. Equation (5.5) was used to calculate the torsion energies of these branched systems. In out-of-plane systems, the torsion energies do not contain energies from van der Waals and electrostatics, hence, the out-of-plane torsion definitions in this work are the same as in Amber. In this way, the total energy of a protein structure can be simplified as:

$$E_{total} = Cons + \sum_{torsion} e_{ij} + \sum_{nonbond} e_{kl} \quad (5.6)$$

where e_{ij} and e_{kl} are pair wise energies for the torsion and nonbond interactions, respectively.

It is known that by assigning different importance factors to emphasize more significant pair wise interactions can efficiently improve a scoring function's ability to identify the native protein structure,[117] therefore, equation (5.7) was used to represent the final score (S_{total}) of a given protein structure.

$$S_{total} = C + \sum_{torsion} p_{ij} \times e_{ij} + \sum_{nonbond} p_{kl} \times e_{kl} \quad (5.7)$$

where C is a constant, p_{ij} and p_{kl} are the pair wise importance factors, which are to-be-determined parameters, for torsion and nonbond interactions, respectively. In this way, a three dimensional protein structure can be represented using a series of atom pair wise energies. Specifically, with the known amount of torsion and nonbond interactions in this work, the total score of a protein structure can be obtained using:

$$S_{total} = C + \sum_{tor=1}^{1143\ or\ 1175} p_{ij} \times e_{ij} + \sum_{nonb=1}^{18336} p_{kl} \times e_{kl} \quad (5.8)$$

In one protein structure, a specific torsion / nonbond atom pair might exist several times. For each torsion / nonbond atom pair, e_{ij} / e_{kl} in equation (5.8) is the sum of all the energies of the specific atom pairs that exist in the protein structure. For example, if torsion "H-5 -N-C- O-5" is present three times in a protein structure, there are three corresponding energies that can be obtained using equation (5.3). The sum of these three energies is assigned as the energy of torsion "H-5 -N-C- O-5" in equation (5.8). In this way, the total score S_{total} for any protein structure can be represented as the sum of 19479 (19479 = 1143 + 18336, ff94) pair wise energies or 19511 (19511 =

1175+18336, ff14SB) pair wise energies. With equation (5.8), a three dimensional structure can be represented as a one dimensional vector.

5.3 Structures for encoding validation

The first and most important thing is to make sure that the calculations from both FFENCODER and Amber are consistent. In FFENCODER, there are five assumptions: (1) only 20 common amino acids were considered; (2) terminal amino acids were not treated differently; (3) amino acids HIP, HIE, and HID were treated as HID; (4) there is no energy cutoff when the repulsion is too strong; (5) no metal ions were considered in the calculation. Assumptions (1) ~ (3) were made to control the total number of atom pairs used in RF models.

In order to test if FFENCODER was consistent with the Amber package, two sets of structures were constructed. The first test set contains 20 different amino acid structures in order to test if torsion / nonbond pairs, which exist in the same amino acid structure, were encoded correctly. The second test set, which consists of 210 dipeptide structures (all possible connections between those 20 amino acids), was used to test if torsion / nonbond atom pairs exists in different amino acids were correctly encoded.

The LEaP module, which is from AMBERTools 18,[118] was used to generate the topologies of the single amino acids and dipeptides. Systems were first minimized in implicit solvent described by the generalized Born model.[119] 25000 cycles of steepest descent minimization followed by 25000 cycles of conjugate gradient minimization were performed. Without considering solvent, by setting the minimization step to zero, energies of the minimized single amino acids / dipeptides were calculated. All simulations were performed with pmemd.cuda from the Amber18 package.[118] No periodicity was applied and the cut off was set as 9999 Å to include all long range interactions.

5.4 Encoding validation

In order to confirm the force field parameters were correctly encoded, calculated results from FFENCODER and Amber were compared. Based on the RF algorithm described in the method section, it is necessary to guarantee all atom pair wise energies were encoded correctly. Because of the complexity of protein structures, it is hard to compare all pairwise energies in a protein structure. Here, two sets of relatively simple structures were used instead of using whole proteins as test structures. The first test set contains 20 common amino acids, and the second one contains 210 (21×20/2) dipeptides. Although the structures in these two test sets are relatively simple, they cover most of the parameter space of the Amber force fields. As a compromise, five energy components were compared instead of comparing all pairwise energies. Here, the total energies of dihedral, 1_4_van der Waals (1_4_vdW), 1_4 electrostatics (1_4_EEL), van der Waals (vdW), and electrostatics (EEL) interactions were compared. Those five energy values calculated using FFENCODER and the Amber software package are listed in **Table 5.3 -Table 5.14**. **Figure 5.1** shows the comparisons of different energy components between the two programs. The *x-axis* is the energy calculated with Amber, and the *y-axis* represents the corresponding energies provided by FFENCODER. Panels (a)-(e) represent the comparisons of the dihedral, 1_4_vdW, 1_4_EEL, vdW, and EEL between the two programs. Columns (1) and (3) represent the comparison for ff94 for the single amino acid and the dipeptide test sets, respectively. Similarly, columns (2) and (4) are the comparisons for ff14SB. The trend line, equation of the trend line, and R-squared value are presented in each plot to evaluate the overall performance of FFENCODER. Theoretically, the trend line equation should be $y = x$. Thus, if the trend line equations showed in the plot are closer to $y = x$, that means the results calculated from FFENCODER are closer to the original Amber

output. In Figure 2, all trend line equations are close to $y = x$, the range of slopes from those trend lines is [0.9985, 1.0003], and the range of absolute intercept value is [0.000002, 0.0046]. It is clear that the ranges of both slope and intercept are small. Furthermore, R-squared values for all plots are equal to 1.

Table 5.15 gives a brief summary of the absolute energy difference results given in **Table 5.5**, **Table 5.8**, **Table 5.11**, and **Table 5.14**. For the single amino acid test set, the ranges of energy differences for ff94 and ff14SB are [-0.0177, 0.0244 kcal/mol] and [-0.0191, 0.0247 kcal/mol], respectively. For the dipeptide test set, the range of energy differences for ff94 is [-0.0519, 0.0807 kcal/mol], and the corresponding range for ff14SB is [-0.0434, 0.0584 kcal/mol]. It is clear that for both the ff94 and ff14SB parameter set, our results are consistent with the Amber force fields. Based on **Figure 5.1** and **Table 5.15**, we conclude that we are modeling canonical Amber force fields.

5.5 Feature importance analysis

Based on the definitions of atom types, torsion types, and nonbond types given in the method section, there are 191 atom types, 1143 pair wise torsion types, and 18336 pair wise nonbond types in ff94, and 191 atom types, 1175 pair wise torsion types, and 18336 pair wise nonbond types in ff14SB. Because of the limited number of data points (around 154,000), it is computationally more intensive and largely unnecessary to include all pair wise interactions as features for our RF models. Here, before building up the final RF model, a feature importance analysis was performed to filter the pair wise interactions. **Figure 5.2** shows the importance analyses for features in the ff94 and ff14SB parameter sets. In **Figure 5.2**, the red points in each plot are the 500^{th} most important feature in the corresponding parameter set. After the 500^{th} feature, contributions from atom pairs are trivial. Hence, in this work, top 100, 200, 300, 400, and 500 features were used to construct RF models. At the same time, the risk of overfitting and the computational cost can also be diminished with these smaller feature sets.

5.6 Accuracy

The most important criteria to judge a scoring function is its ability to identify the native structure. In this work, accuracy values are considered to represent the ability of scoring functions to locate native structures. The accuracy values used here is defined in the method section, it evaluates the capability of a scoring function to compare two structures. A scoring function with a high accuracy is expected to perform better in identifying the native structure. For example, if a decoy set contains one native and 100 decoy structures, in order to locate the native structure, the scoring function is required to make 200 correct comparisons between native and each decoy structure. In other words, the native structure cannot be detected if the scoring function makes one wrong comparison. In general, the higher the accuracy value is, the better the scoring function performs.

In **Table 5.16**, from column 3 to column 5 are comparisons of accuracy between RF models and other scoring functions. The results can be analyzed from three perspectives: (i) If RF models with force field parameters are compared with traditional scoring functions (RWplus, DFIRE, dDFIRE, and GOAP), it is obvious that RF models achieved higher averaged accuracy values, and the lowest accuracies are still higher than accuracies from conventional scoring functions. Therefore, RF models have a better performance when differentiating native and decoy structures relative to traditional scoring functions. (ii) If RF models with force field parameters are compared with RF models based on knowledge-based potentials (KECSA2), for models based on different numbers of input features (top 100 to 500), RF models with force field parameters always provide a higher average accuracy than models based on a knowledge-based potential (KECSA2). (iii) If RF models with ff94 are compared with models based on ff14SB, with the same number of input features (100

to 500), models based on each force field parameter sets generated similar averaged accuracies, and the confidence ranges ([lowest accuracy, highest accuracy]) are similar as well. With increasing number of input features, averaged accuracy values from RF models based on ff94 and ff14SB remain similar.

Based on this we conclude that RF models with force field potentials perform better than all other scoring functions considered in this work, and the RF models depend on ff94 and ff14SB perform similar with the same number of input features.

5.7 Native ranking

Other than accuracy, another important criteria to evaluate the performance of a scoring function is native ranking. A scoring function is always expected to give the native structure the lowest rank. In **Table 5.16**, from column 6 to column 8 are the comparisons of native ranking between different scoring functions. The comparison results can be analyzed again in three ways: (i) If the RF models with force field parameters are compared with traditional scoring functions (RWplus, DFIRE, dDFIRE, and GOAP), it is clear that the averaged native rankings from RF models are smaller than the corresponding values from conventional scoring functions. Furthermore, the native rankings provided by conventional scoring functions are higher than all of the highest native ranking generated by RF models with force field parameters. Hence, RF models with force field parameters outperformed the traditional scoring functions considered in this work. (ii) If RF models based on force field parameters are compared with RF models based on knowledge-based potential (KECSA2), with the same number of input features, RF models with force field potentials can always achieve lower native rankings than models with KECSA2. (iii) If RF models based on ff94 are compared with RF models based on ff14SB, with the same number of input features, the averaged native rankings generated by RF models with each potential set are similar.

In summary, we can conclude that, in the native ranking test, RF models with force field parameters outperformed all other scoring functions considered in this work; RF models with ff94 and ff14SB perform similar; with increasing number of input features, RF models with force field potentials provide native rankings of around 4.

When comparing native rankings with the corresponding accuracies, a correlation can be found. In general, the higher the accuracy is, the lower the native ranking will be. Here, a decoy set with one native and 100 decoys can be used as an example. In order to locate the native structure, the scoring function is required to make 200 correct comparisons between the native structure and each decoy. If a scoring function has an accuracy value of 0.95, that means it made 10 incorrect comparisons. The ranking of native structure provided by that scoring function will be larger or equal to 5. If a scoring function made more incorrect comparisons, it will have a lower accuracy value, and a higher native ranking. Hence, in general the higher the accuracy, the lower the native ranking.

5.8 1st decoy RMSD and TM-score

Besides accuracy and native ranking, there are other two values, 1st decoy RMSD and TM-score,[79] usually used to evaluate the capability of a scoring function to identify the best decoy structure. The best decoy is the most stable decoy structure selected by a scoring function among a set of candidates. In the protein design and protein structure prediction fields, the scoring function is expected to identify the most stable decoy structure among a large number of structure candidates. RMSD and TM-score are often used to represent the quality of a decoy structure. RMSD refers to the root mean squared deviation of all Cα pairs of the decoy to the native structure. TM-score gives a large distance a small weight, and makes the magnitude of TM-score more sensitive to topology. The best decoy structure selected by a scoring function is always expected to have low RMSD and a high TM-score value.

Table 5.17 shows the 1st decoy RMSD and TM-score comparisons between different scoring functions. In general, all scoring functions in Table 4 provide 1st decoy RMSD values around 4.5 Å, and a 1st decoy TM-score around 0.62. Some of them generate a RMSD or TM-score slightly better than others. The RMSD difference between the highest and lowest RMSD values is smaller than 1 Å, and the TM-sore difference is within 0.1. When comparing RF based scoring functions, no matter which potential data set was used (force field potential like ff94 and ff14SB, or knowledge-based potentials like KECSA2), the performances in selecting the best decoy are similar. This common performance from RF models is due to the fact that the RF models were trained on native-decoy comparisons, and the decoy-decoy comparisons are missing in the training data set. However, the decoy-decoy comparisons are necessary to locate the best decoy structure

from a large number of candidates. In our previous work,[124] we proved that with more decoy-decoy comparisons included in the training set, the ability of RF models to identify the best decoy can be improved. In this work, it is hard to include more decoy-decoy comparisons due to the sparseness of the data. **Table 5.18** shows the distribution of the decoys lowest RMSD values in all 234 protein systems, there are only 75 systems that provide one decoy structure with a RMSD smaller than 1 Å. Compared to the total size of native-decoy comparisons in the training data, the decoy-decoy comparison information is insufficient. Therefore, it is hard to improve the ability of RF models with force field potentials to identify the best decoy structure in this work. Taken altogether, it can be concluded that, RF models with force field potentials perform similar to other scoring functions considered in this work for the best decoy selection test.

5.9 Impact of the RF algorithm

RF models with force field potentials can achieve a higher accuracy value and a lower native ranking than other scoring functions considered in this work. At the same time, all scoring functions used here perform similar in selecting the best decoy structure. It is interesting to test whether the better performance from RF models is the result of the RF algorithm or not. In order to test the importance of RF refinement, accuracy values from scoring functions with and without RF models should be compared. RF models can emphasize more important pair wise interactions and ignore insignificant ones, on the contrary, a scoring function without RF refinement should treat every pair wise energy as the same. In other words, without RF refinement, a score value can be directly calculated as the sum (the sum of each descriptor) of all pair wise energies obtained from FFENCODER with the ff94 or ff14SB parameter sets. Then, an accuracy can be obtained based on these calculated scores. **Table 5.19** shows the comparison between scoring functions with and without RF refinement. It is clear that with RF refinement, the accuracy values can be improved from ~0.65 to ~0.99. With both the ff94 and ff14SB force field parameters, the trend was the same. Therefore, it can be concluded that the RF refinement protocol is important, and it helped the scoring functions to achieve higher accuracy values.

5.10 Potential analysis

The performance of scoring functions with force field descriptors can be improved by RF refinement. On the other hand, it is also necessary to test whether the force field parameters are important or not. In order to test the importance of force field potentials in RF models, the performance of RF models with and without force field parameters need to be compared. A set of RF models constructed based on counts of interactions were used as a reference. In potential functions, different distances between atoms will provide different pair wise energies, using counts of interactions eliminates the impact from potential functions on RF models. In detail, there are no energy difference between different atom pairs, and the same atom pair with different distances are treated as the same. For example, if an atom pair 'H-0_C-0' exists five times in a protein, the count of interactions of that atom pair is five, and five will be used instead of total energy. In this way, the torsion and nonbond potential function in force fields were replaced by horizontal lines with intercepts of 1. **Table 5.20** shows the comparison between scoring functions with and without force field parameters. Without force field potential functions, the RF models can only generate accuracy values of 0.679 and 0.712, with 100 and 500 features, respectively. On the other hand, with force field parameters, the accuracy values increased to ~0.99 using either 100 or 500 features. Hence, the force field parameters are also important in the model. Furthermore, it can be concluded that both force field potential functions and the RF refinement algorithm are important in the RF model, and none of them can generate high accuracies by itself, combining them together is the only way to achieve the best performance.

5.11 Conclusion

In this work, Amber force field pairwise potentials from ff94 and ff14SB were successfully encoded based on five assumptions: (1) only 20 common amino acids are considered; (2) terminal amino acids are not treated differently; (3) amino acids HIP, HIE, and HID are all treated as HID; (4) there is no energy cutoff if the repulsion is too strong; (5) no metal ions were considered in the calculation. Detailed pair wise energies obtained from FFENCODER were used as input features to construct RF models. 12 popular protein folding decoy sets were combined based on protein systems, and used to train and test RF models. The comparisons between RF models and other scoring functions suggest that RF models with force field parameters outperformed other scoring functions in accuracy and in native ranking tests, and perform similar to other scoring functions in selecting the best decoy. The importance of the RF algorithm was tested by comparing scoring functions with and without RF refinement and the results clearly showed that the RF algorithm is an important reason for the observed high accuracies. On the other hand, counts of interactions were used to replace all force field potential functions in order to test the importance of the force field potentials. The comparisons between RF models with and without force field potentials suggest that force fields also play a key role in the observed high accuracy values. A model cannot achieve high accuracy without both RF refinement and appropriate force field parameters. Moreover, in this work we only showed one example where we built ML models using force field potential functions. FFENCODER makes it possible to combine other novel ML algorithms with pair wise energies as encoded by Amber force field potentials.

APPENDICES

APPENDIX A: TABLES

Table 1.1. An example of training data to construct a decision tree.

Day	weather	Temperature	Humidity	Wind	Jogging
1	Sunny	Hot	Normal	Weak	No
2	Sunny	Mild	High	Weak	Yes
3	Overcast	Mild	High	Strong	Yes
4	Overcast	Hot	Normal	Weak	No
5	Rain	Cool	High	Strong	No
6	Rain	Cool	High	Strong	No
7	Sunny	Mild	Normal	Weak	Yes
8	Sunny	Mild	Normal	Weak	Yes
9	Sunny	Hot	High	Strong	No
10	Overcast	Hot	High	Strong	Yes
11	Overcast	Hot	Normal	Weak	Yes
12	Rain	Cool	High	Weak	No
13	Rain	Cool	High	Strong	No
14	Overcast	Mild	Normal	Weak	Yes

Table 1.2. Relationship between humidity and jogging decisions.

Humidity	Jogging		Expected		Difference	
	Yes	No	Yes	No	Yes	No
High	3	5	3.5	3.5	-0.5	1.5
Normal	4	2	3.5	3.5	0.5	-1.5
total	7	7	7	7	-	-

Table 2.1. Atom types in the GARF potential database.[a]

Atom type	definition
Protein atom types	
C	sp^2 carbonyl carbon and aromatic carbon with hydroxyl substituent in tyrosine
C*	sp^2 aromatic carbon in 5-membered ring with one substituent
CA	sp^2 aromatic carbon in 6-membered ring with one substituent
CB	sp^2 aromatic carbon at junction between 5- and 6-membered rings
CC	sp^2 aromatic carbon in 5-membered ring with one substituent and next to a nitrogen atom
CN	sp^2 aromatic junction carbon in between 5- and 6-membered rings
CR	sp^2 aromatic carbon in 5-membered ring between two nitrogen atoms and bonded to one hydrogen atom (in HIS)
CT	sp^3 carbon with four explicit substituents
CV	sp^2 aromatic carbon in 5-membered ring bonded to one nitrogen atom and bonded to an explicit hydrogen
CW	sp^2 aromatic carbon in 5-membered ring bonded to one N−H group and an explicit hydrogen
N	sp^2 nitrogen in amide group
N2	sp^2 nitrogen in base NH_2 group or arginine NH_2
N3	sp^2 nitrogen with four substituents
NA	sp^2 nitrogen in 5-membered ring with hydrogen attached
NB	sp^2 nitrogen in 5-membered ring with lone pairs
O	carbonyl oxygen
O2	carboxyl oxygen
OH	alcohol oxygen
S	sulfur in disulfide linkage or methionine
SH	sulfur in cysteine
Ligand atom types	
C.3	sp^3 carbon without polar group substituent
C.2	sp^2 carbonyl carbon without polar group substituent
C.1	sp carbon
C.ar	sp^2 aromatic carbon without polar group substituent
O.3	alcohol oxygen
O.3P	ether oxygen
O.2	carbonyl oxygen
O.co2	carboxylate oxygen
O.2v	sulfate/phosphate oxygen
N.2	sp/sp^2/aromatic nitrogen
N.1h	sp^3 nitrogen with one hydrogen atom attached
N.2h	sp^3 nitrogen with two hydrogen atoms attached
N.3h	sp^3 nitrogen with three hydrogen atoms attached
P	Phosphorus
F	Fluorine
Cl	Chlorine
Br	Bromine
I	Iodine
C.cat	carbon cation

Table 2.1. (cont'd)

S.3	thiol/thioether sulfur
S.o	sulfoxide sulfur
C.3X	sp^3 carbon with polar group substituent
C.2X	sp^2 carbonyl carbon with polar group substituent
C.arX	sp^2 aromatic carbon with polar group substituent

a. This table is as same as Table 2 in GARF paper, reference 114.

Table 2.2. General form of a confusion matrix.

	Predicted (class 1)	Predicted (class 0)
Actual (class 1)	TP	FN
Actual (class 0)	FP	TN

Table 3.1. Accuracy values for different models.[a]

Decoy sets	RF model			KECSA2	RWplus	DFIRE	dDFIRE	GOAP
	Averaged accuracy	Highest accuracy	Lowest accuracy					
4state_reduced	1.000	1.000	0.998	0.997	1.000	1.000	1.000	1.000
fisa	1.000	1.000	0.997	0.751	0.775	0.810	0.761	0.816
fisa_casp3	1.000	1.000	0.999	0.842	1.000	1.000	0.989	1.000
hg_structal	0.971	0.994	0.939	0.828	0.902	0.882	0.881	0.934
ig_structal	1.000	1.000	0.999	0.887	0.540	0.536	0.895	0.955
ig_structal_hires	1.000	1.000	1.000	0.953	0.580	0.567	0.942	1.000
I-TASSER	0.982	0.998	0.966	0.971	0.914	0.856	0.919	0.857
lattice_ssift	0.999	1.000	0.998	1.000	1.000	1.000	1.000	1.000
lmds	0.999	1.000	0.997	0.963	0.722	0.727	0.735	0.798
lmds_v2	0.999	1.000	0.990	0.762	0.861	0.899	0.871	0.906
MOULDER	0.988	1.000	0.969	0.829	0.982	0.985	0.982	0.991
ROSETTA	1.000	1.000	1.000	0.776	0.939	0.770	0.537	0.798

a) RF models were trained on different decoy sets.

Table 3.2. Native structure's ranking of different models.[a]

Decoy sets	RF model			KECSA2	RWplus	DFIRE	dDFIRE	GOAP
	Averaged native ranking	Highest native ranking	Lowest native ranking					
4state_reduced	1.70	7.00	1.00	3.29	1.00	1.00	1.00	1.00
fisa	1.50	3.00	1.00	125.50	113.5	95.75	120.25	92.75
fisa_casp3	1.00	1.00	1.00	228.60	1.60	1.60	17.20	1.00
hg_structal	2.43	5.67	1.33	5.93	3.79	4.38	4.41	2.90
ig_structal	1.01	1.08	1.00	8.03	29.7	29.98	7.57	3.79
ig_structal_hires	1.00	1.00	1.00	1.90	8.95	9.20	2.10	1.00
I-TASSER	13.26	39.17	2.83	13.71	38.13	63.25	36.16	62.89
lattice_ssift	1.05	1.50	1.00	1.38	1.00	1.00	1.00	1.00
lmds	1.05	1.50	1.00	138.91	138.90	136.50	132.2	101.10
lmds_v2	1.40	5.00	1.00	29.50	17.70	13.10	16.5	12.30
MOULDER	5.65	11.50	1.00	55.15	6.65	5.75	6.65	3.80
ROSETTA	1.00	1.00	1.00	23.33	7.07	23.84	47.16	21.07

a) RF models were trained on different decoy sets.

Table 3.3. 1st decoy's RMSD for different models.[a]

Decoy sets	RF model			KECSA2	RWplus	DFIRE	dDFIRE	GOAP
	Averaged 1st decoy RMSD	Highest 1st decoy RMSD	Lowest 1st decoy RMSD					
4state_reduced	3.28	6.06	1.34	3.17	2.69	2.61	2.25	1.83
fisa	6.13	9.60	4.68	6.51	5.26	5.77	6.05	4.48
fisa_casp3	11.67	15.76	6.35	12.30	11.80	11.10	9.88	10.40
hg_structal	2.62	4.88	1.39	2.59	2.31	2.45	2.66	2.43
ig_structal	2.21	2.62	1.73	2.02	2.00	2.06	1.86	1.88
ig_structal_hires	2.63	4.10	1.48	2.06	2.14	2.13	2.10	2.08
I-TASSER	1.71	2.21	1.27	1.73	1.73	1.70	1.70	1.65
lattice_ssift	10.37	11.44	9.17	9.55	9.26	9.17	9.21	10.01
lmds	7.91	10.75	4.13	7.72	8.08	8.23	6.69	8.55
lmds_v2	7.60	9.38	4.46	8.01	7.74	7.82	7.67	7.36
MOULDER	9.18	12.83	6.67	10.77	9.74	9.98	10.08	9.96
ROSETTA	7.27	8.75	5.88	8.54	7.65	7.36	7.53	7.53

a) RF models were trained on different decoy sets.

Table 3.4. 1st decoy's TM-score of different models.[a]

Decoy sets	RF model			KECSA2	RWplus	DFIRE	dDFIRE	GOAP
	Averaged 1st decoy TM-score	Highest 1st decoy TM-score	Lowest 1st decoy TM-score					
4state_reduced	0.620	0.864	0.278	0.617	0.700	0.714	0.725	0.791
fisa	0.398	0.468	0.315	0.411	0.467	0.432	0.389	0.472
fisa_casp3	0.263	0.318	0.233	0.296	0.285	0.286	0.313	0.313
hg_structal	0.871	0.924	0.790	0.888	0.894	0.892	0.869	0.891
ig_structal	0.931	0.941	0.923	0.943	0.945	0.943	0.951	0.950
ig_structal_hires	0.936	0.950	0.914	0.939	0.948	0.947	0.949	0.951
I-TASSER	0.431	0.529	0.377	0.451	0.442	0.451	0.445	0.444
lattice_ssift	0.224	0.291	0.179	0.240	0.270	0.258	0.277	0.249
lmds	0.347	0.430	0.283	0.333	0.344	0.336	0.376	0.342
lmds_v2	0.367	0.484	0.296	0.363	0.442	0.451	0.445	0.444
MOULDER	0.429	0.555	0.211	0.394	0.426	0.418	0.416	0.422
ROSETTA	0.487	0.573	0.410	0.438	0.460	0.466	0.477	0.471

a) RF models were trained on different decoy sets.

Table 3.5. Comparison of accuracies of RF models with different numbers of features.[a]

Decoy sets	RF model_with_IMP_500			RF_model_with_all_features		
	Averaged accuracy	Highest accuracy	Lowest accuracy	Averaged accuracy	Highest accuracy	Lowest accuracy
4state_reduced	1.000	1.000	0.999	1.000	1.000	0.998
fisa	1.000	1.000	1.000	1.000	1.000	0.997
fisa_casp3	0.992	0.999	0.987	1.000	1.000	0.999
hg_structal	0.955	0.977	0.909	0.971	0.994	0.939
ig_structal	0.999	1.000	0.998	1.000	1.000	0.999
ig_structal_hires	1.000	1.000	1.000	1.000	1.000	1.000
I-TASSER	0.978	0.997	0.955	0.982	0.998	0.966
lattice_ssift	1.000	1.000	0.997	0.999	1.000	0.998
lmds	0.994	1.000	0.948	0.999	1.000	0.997
lmds_v2	1.000	1.000	1.000	0.999	1.000	0.990
MOULDER	0.989	1.000	0.974	0.988	1.000	0.969
ROSETTA	0.999	1.000	0.996	1.000	1.000	1.000

a) RF models were trained on different decoy sets.

Table 3.6. Comparison of the overall performance of RF models (with different number of features) with traditional potentials on overall data set.

		nIMP_100	nIMP_500	KECSA2	RWplus	DFIRE	dDFIRE	GOAP
Accuracy	Averaged	0.963	0.981	0.908	0.916	0.886	0.904	0.917
	Lowest	0.931	0.965	-	-	-	-	-
	Highest	0.987	0.994	-	-	-	-	-
Ranking of native structure	Averaged	10.62	7.95	25.67	23.43	31.35	26.49	23.09
	Lowest	4.86	2.77	-	-	-	-	-
	Highest	21.64	17.59	-	-	-	-	-
RMSD of 1st selected decoy	Averaged	4.62	4.57	4.84	4.53	4.51	4.44	4.45
	Lowest	3.77	3.52	-	-	-	-	-
	Highest	5.49	5.72	-	-	-	-	-
TM-score of 1st selected decoy	Averaged	0.634	0.614	0.610	0.622	0.623	0.625	0.674
	Lowest	0.574	0.536	-	-	-	-	-
	Highest	0.685	0.695	-	-	-	-	-

Table 3.7. Comparison of RF models based on different potentials.

	RF model_nIMP_100			RF model_nIMP_500		
	Averaged accuracy	Highest accuracy	Lowest accuracy	Averaged accuracy	Highest accuracy	Lowest accuracy
Original	0.963	0.987	0.931	0.981	0.994	0.965
Scrambled	0.822	0.854	0.805	0.827	0.864	0.799
Uniform	0.940	0.968	0.872	0.977	0.990	0.959

Table 4.1. Comparisons between RF models and 29 other scoring functions.

		accuracy	Native's ranking	1st decoy RMSD
RF models	Averaged	0.953	4.49	3.87
	Highest	0.969	5.54	4.47
	Lowest	0.942	3.54	3.38
Conventional SFs	GOLD-ASP	0.924	6.13	1.74
	GOLD-ChemPLP	0.923	6.25	1.51
	DS-PLP1	0.917	6.68	1.80
	DS-PLP2	0.914	7.07	1.87
	MOE-Affinity_dG	0.900	8.23	2.42
	Xscore-HMScore	0.891	8.89	2.45
	Xscore-Average	0.886	9.33	2.38
	GOLD-ChemScore	0.882	9.58	1.72
	DS-PMF04	0.874	10.58	3.38
	SYBYL-PMF	0.874	10.53	3.42
	Xscore-HPScore	0.871	10.63	2.75
	MOE-Alpha	0.870	10.38	1.85
	Xscore-HSScore	0.869	10.80	2.64
	DS-LigScore2	0.867	10.67	1.83
	MOE-London_dG	0.863	11.38	2.52
	DS-PMF	0.857	11.89	3.48
	MOE-ASE	0.856	11.88	2.91
	GlideScore-SP	0.832	13.20	1.72
	DS-LigScore1	0.823	14.28	2.31
	GlideScore-XP	0.823	14.07	1.86
	GOLD-GoldScore	0.819	14.70	1.88
	DS-LUDI2	0.807	15.62	2.23
	DS-LUDI1	0.799	16.35	2.34
	DS-LUDI3	0.783	17.48	2.87
	SYBYL-ChemScore	0.782	17.70	2.40
	SYBYL-Gscore	0.725	22.64	3.13
	dSAS	0.692	25.48	3.96
	DS-Jain	0.685	25.63	2.90
	SYBYL-Dscore	0.674	26.70	4.03

Table 4.2. Comparison between RF models with considering different number of decoy pose in training set.

		Accuracy	Native's ranking	1st decoy RMSD
With no decoy structure	Averaged	0.953	4.49	3.87
	Highest	0.969	5.54	4.47
	Lowest	0.942	3.54	3.38
With one lowest RMSD decoy structure	Averaged	0.958	4.28	2.41
	Highest	0.974	7.49	2.72
	Lowest	0.921	3.03	2.13
With two lowest RMSD decoy structure	Averaged	0.950	5.03	2.50
	Highest	0.957	6.08	2.99
	Lowest	0.937	4.39	1.95
With three lowest RMSD decoy structure	Averaged	0.947	5.21	2.27
	Highest	0.963	6.56	2.44
	Lowest	0.930	3.97	1.73

Table 4.3. Comparison between RF models with different probability function sets.

	RF models		
	Averaged accuracy	Highest accuracy	Lowest accuracy
Original GARF	0.953	0.969	0.942
Scrambled GARF	0.933	0.951	0.911
Uniform GARF	0.953	0.980	0.918

Table 4.4. Summary of peak positions and number of probability functions at each peak positions in GARF.

Peak position / Å	number of Probability functions
2.5	2
2.6	2
2.7	5
2.8	14
2.9	9
3.0	6
3.1	5
3.2	2
3.3	14
3.4	23
3.5	23
3.6	23
3.7	53
3.8	43
3.9	34
4.0	36
4.1	24
4.2	28
4.3	38
4.4	30
4.5	8
4.6	12
4.7	13
4.8	17
4.9	8
5.0	4
5.1	4

Table 4.5. Accuracy values for different training set sizes from RF models with original and uniform GARF.

Training size / %	Original GARF			Uniform GARF		
	Averaged accuracy	Highest accuracy	Lowest accuracy	Averaged accuracy	Highest accuracy	Lowest accuracy
5	0.883	0.926	0.833	0.900	0.929	0.867
10	0.923	0.939	0.912	0.916	0.936	0.879
15	0.919	0.941	0.884	0.933	0.948	0.913
20	0.926	0.947	0.887	0.936	0.946	0.902
25	0.938	0.953	0.918	0.939	0.960	0.919
30	0.943	0.953	0.935	0.945	0.965	0.933
35	0.947	0.962	0.928	0.942	0.960	0.914
40	0.945	0.959	0.924	0.946	0.951	0.933
45	0.941	0.951	0.925	0.949	0.969	0.922
50	0.953	0.966	0.938	0.944	0.957	0.935
55	0.952	0.973	0.916	0.945	0.968	0.923
60	0.954	0.975	0.935	0.950	0.968	0.937
65	0.945	0.964	0.937	0.949	0.965	0.921
70	0.954	0.980	0.922	0.955	0.972	0.924
75	0.953	0.980	0.922	0.952	0.977	0.924
80	0.953	0.969	0.942	0.953	0.980	0.918

Table 5.1. Summary of charge, ε, and van der Waals radii for each atom type in ff94 force field.

Atom type	Charge	Van de Waals radii	ε	Atom type	charge	Van de Waals radii	ε
C*-0	-0.1415	1.9080	0.0860	H1-1	0.1560	1.3870	0.0157
C-0	0.5973	1.9080	0.0860	H1-10	0.0881	1.3870	0.0157
C-1	0.7341	1.9080	0.0860	H1-11	0.0869	1.3870	0.0157
C-2	0.713	1.9080	0.0860	H1-12	0.0922	1.3870	0.0157
C-3	0.7994	1.9080	0.0860	H1-13	0.1426	1.3870	0.0157
C-4	0.5366	1.9080	0.0860	H1-14	0.0440	1.3870	0.0157
C-5	0.6951	1.9080	0.0860	H1-15	0.0684	1.3870	0.0157
C-6	0.8054	1.9080	0.0860	H1-16	0.0978	1.3870	0.0157
C-7	0.5896	1.9080	0.0860	H1-17	0.0391	1.3870	0.0157
C-8	0.3226	1.9080	0.0860	H1-18	0.0641	1.3870	0.0157
CA-0	0.8076	1.9080	0.0860	H1-19	0.0843	1.3870	0.0157
CA-1	0.0118	1.9080	0.0860	H1-2	0.0687	1.3870	0.0157
CA-10	-0.1906	1.9080	0.0860	H1-20	0.0352	1.3870	0.0157
CA-11	-0.2341	1.9080	0.0860	H1-21	0.1007	1.3870	0.0157
CA-2	-0.1256	1.9080	0.0860	H1-22	0.0043	1.3870	0.0157
CA-3	-0.1704	1.9080	0.0860	H1-23	0.1123	1.3870	0.0157
CA-4	-0.1072	1.9080	0.0860	H1-24	0.0876	1.3870	0.0157
CA-5	-0.2601	1.9080	0.0860	H1-25	0.0969	1.3870	0.0157
CA-6	-0.1134	1.9080	0.0860	H1-3	0.1048	1.3870	0.0157
CA-7	-0.1972	1.9080	0.0860	H1-4	0.0880	1.3870	0.0157
CA-8	-0.2387	1.9080	0.0860	H1-5	0.1124	1.3870	0.0157
CA-9	-0.0011	1.9080	0.0860	H1-6	0.1112	1.3870	0.0157
CB-0	0.1243	1.9080	0.0860	H1-7	0.0850	1.3870	0.0157
CC-0	-0.0266	1.9080	0.0860	H1-8	0.1105	1.3870	0.0157
CN-0	0.138	1.9080	0.0860	H1-9	0.0698	1.3870	0.0157
CR-0	0.2057	1.9080	0.0860	H4-0	0.1147	1.4090	0.0150
CT-0	0.0337	1.9080	0.1094	H4-1	0.2062	1.4090	0.0150
CT-1	-0.1825	1.9080	0.1094	H5-0	0.1392	1.3590	0.0150
CT-10	0.0213	1.9080	0.1094	HA-0	0.1330	1.4590	0.0150
CT-11	-0.1231	1.9080	0.1094	HA-1	0.1430	1.4590	0.0150
CT-12	-0.0031	1.9080	0.1094	HA-2	0.1297	1.4590	0.0150
CT-13	-0.0036	1.9080	0.1094	HA-3	0.1572	1.4590	0.0150
CT-14	-0.0645	1.9080	0.1094	HA-4	0.1417	1.4590	0.0150
CT-15	0.0397	1.9080	0.1094	HA-5	0.1447	1.4590	0.0150
CT-16	0.056	1.9080	0.1094	HA-6	0.1700	1.4590	0.0150
CT-17	0.0136	1.9080	0.1094	HA-7	0.1699	1.4590	0.0150
CT-18	-0.0252	1.9080	0.1094	HA-8	0.1656	1.4590	0.0150
CT-19	0.0188	1.9080	0.1094	HC-0	0.0603	1.4870	0.0157
CT-2	-0.2637	1.9080	0.1094	HC-1	0.0327	1.4870	0.0157
CT-20	-0.0462	1.9080	0.1094	HC-10	0.0187	1.4870	0.0157
CT-21	-0.0597	1.9080	0.1094	HC-11	0.0882	1.4870	0.0157
CT-22	0.1303	1.9080	0.1094	HC-12	0.0236	1.4870	0.0157
CT-23	-0.3204	1.9080	0.1094	HC-13	0.0186	1.4870	0.0157
CT-24	-0.0430	1.9080	0.1094	HC-14	0.0457	1.4870	0.0157
CT-25	-0.066	1.9080	0.1094	HC-15	-0.0361	1.4870	0.0157
CT-26	-0.0518	1.9080	0.1094	HC-16	0.1000	1.4870	0.0157
CT-27	-0.1102	1.9080	0.1094	HC-17	0.0362	1.4870	0.0157
CT-28	0.3531	1.9080	0.1094	HC-18	0.0103	1.4870	0.0157
CT-29	-0.4121	1.9080	0.1094	HC-19	0.0621	1.4870	0.0157
CT-3	-0.0007	1.9080	0.1094	HC-2	0.0285	1.4870	0.0157

Table 5.1. (cont'd)

CT-30	-0.2400	1.9080	0.1094	HC-20	0.0241	1.4870	0.0157
CT-31	-0.0094	1.9080	0.1094	HC-21	0.0295	1.4870	0.0157
CT-32	0.0187	1.9080	0.1094	HC-22	0.0213	1.4870	0.0157
CT-33	-0.0479	1.9080	0.1094	HC-23	0.0253	1.4870	0.0157
CT-34	-0.0143	1.9080	0.1094	HC-24	0.0642	1.4870	0.0157
CT-35	-0.0237	1.9080	0.1094	HC-25	0.0339	1.4870	0.0157
CT-36	0.0342	1.9080	0.1094	HC-26	-0.0297	1.4870	0.0157
CT-37	0.0018	1.9080	0.1094	HC-27	0.0791	1.4870	0.0157
CT-38	-0.0536	1.9080	0.1094	HC-3	0.0797	1.4870	0.0157
CT-39	-0.0024	1.9080	0.1094	HC-4	-0.0122	1.4870	0.0157
CT-4	0.039	1.9080	0.1094	HC-5	0.0171	1.4870	0.0157
CT-40	-0.0343	1.9080	0.1094	HC-6	0.0352	1.4870	0.0157
CT-41	0.0192	1.9080	0.1094	HC-7	-0.0173	1.4870	0.0157
CT-42	0.0189	1.9080	0.1094	HC-8	-0.0425	1.4870	0.0157
CT-43	-0.007	1.9080	0.1094	HC-9	0.0402	1.4870	0.0157
CT-44	-0.0266	1.9080	0.1094	HO-0	0.4275	0.0000	0.0000
CT-45	-0.0249	1.9080	0.1094	HO-1	0.4102	0.0000	0.0000
CT-46	0.2117	1.9080	0.1094	HO-2	0.3992	0.0000	0.0000
CT-47	-0.0389	1.9080	0.1094	HP-0	0.1135	1.1000	0.0157
CT-48	0.3654	1.9080	0.1094	HS-0	0.1933	0.6000	0.0157
CT-49	-0.2438	1.9080	0.1094	N-0	-0.4157	1.8240	0.1700
CT-5	0.0486	1.9080	0.1094	N-1	-0.3479	1.8240	0.1700
CT-50	-0.0275	1.9080	0.1094	N-2	-0.9191	1.8240	0.1700
CT-51	-0.005	1.9080	0.1094	N-3	-0.5163	1.8240	0.1700
CT-52	-0.0014	1.9080	0.1094	N-4	-0.9407	1.8240	0.1700
CT-53	-0.0152	1.9080	0.1094	N-5	-0.2548	1.8240	0.1700
CT-54	-0.0875	1.9080	0.1094	N2-0	-0.5295	1.8240	0.1700
CT-55	0.2985	1.9080	0.1094	N2-1	-0.8627	1.8240	0.1700
CT-56	-0.3192	1.9080	0.1094	N3-0	-0.3854	1.8240	0.1700
CT-6	0.0143	1.9080	0.1094	NA-0	-0.3811	1.8240	0.1700
CT-7	-0.2041	1.9080	0.1094	NA-1	-0.3418	1.8240	0.1700
CT-8	0.0381	1.9080	0.1094	NB-0	-0.5727	1.8240	0.1700
CT-9	-0.0303	1.9080	0.1094	O-0	-0.5679	1.6612	0.2100
CV-0	0.1292	1.9080	0.0860	O-1	-0.5894	1.6612	0.2100
CW-0	-0.1638	1.9080	0.0860	O-2	-0.5931	1.6612	0.2100
H-0	0.2719	0.6000	0.0157	O-3	-0.5819	1.6612	0.2100
H-1	0.2747	0.6000	0.0157	O-4	-0.6086	1.6612	0.2100
H-2	0.3456	0.6000	0.0157	O-5	-0.5748	1.6612	0.2100
H-3	0.4478	0.6000	0.0157	O2-0	-0.8014	1.6612	0.2100
H-4	0.4196	0.6000	0.0157	O2-1	-0.8188	1.6612	0.2100
H-5	0.2936	0.6000	0.0157	OH-0	-0.6546	1.7210	0.2104
H-6	0.4251	0.6000	0.0157	OH-1	-0.6761	1.7210	0.2104
H-7	0.3649	0.6000	0.0157	OH-2	-0.5579	1.7210	0.2104
H-8	0.3400	0.6000	0.0157	S-0	-0.2737	2.0000	0.2500
H-9	0.3412	0.6000	0.0157	SH-0	-0.3119	2.0000	0.2500
H1-0	0.0823	1.3870	0.0157				

Table 5.2. Summary of charge, ε, and van der Waals radii for each atom type in ff14SB force field.

Atom type	Charge	Van de Waals radii	ε	Atom type	Charge	Van de Waals radii	ε
2C-0	-0.2041	1.9080	0.1094	H1-1	0.1560	1.3870	0.0157
2C-1	-0.0303	1.9080	0.1094	H1-10	0.0881	1.3870	0.0157
2C-10	0.0018	1.9080	0.1094	H1-11	0.0869	1.3870	0.0157
2C-11	0.2117	1.9080	0.1094	H1-12	0.0922	1.3870	0.0157
2C-2	-0.1231	1.9080	0.1094	H1-13	0.1426	1.3870	0.0157
2C-3	-0.0036	1.9080	0.1094	H1-14	0.0440	1.3870	0.0157
2C-4	-0.0645	1.9080	0.1094	H1-15	0.0684	1.3870	0.0157
2C-5	0.0560	1.9080	0.1094	H1-16	0.0978	1.3870	0.0157
2C-6	0.0136	1.9080	0.1094	H1-17	0.0391	1.3870	0.0157
2C-7	-0.0430	1.9080	0.1094	H1-18	0.0641	1.3870	0.0157
2C-8	-0.1102	1.9080	0.1094	H1-19	0.0843	1.3870	0.0157
2C-9	0.0342	1.9080	0.1094	H1-2	0.0687	1.3870	0.0157
3C-0	0.1303	1.9080	0.1094	H1-20	0.0352	1.3870	0.0157
3C-1	0.3531	1.9080	0.1094	H1-21	0.1007	1.3870	0.0157
3C-2	0.3654	1.9080	0.1094	H1-22	0.0043	1.3870	0.0157
3C-3	0.2985	1.9080	0.1094	H1-23	0.1123	1.3870	0.0157
C-0	0.5973	1.9080	0.0860	H1-24	0.0876	1.3870	0.0157
C-1	0.7341	1.9080	0.0860	H1-25	0.0969	1.3870	0.0157
C-2	0.7130	1.9080	0.0860	H1-3	0.1048	1.3870	0.0157
C-3	0.5366	1.9080	0.0860	H1-4	0.0880	1.3870	0.0157
C-4	0.6951	1.9080	0.0860	H1-5	0.1124	1.3870	0.0157
C-5	0.5896	1.9080	0.0860	H1-6	0.1112	1.3870	0.0157
C-6	0.3226	1.9080	0.0860	H1-7	0.0850	1.3870	0.0157
C*-0	-0.1415	1.9080	0.0860	H1-8	0.1105	1.3870	0.0157
C8-0	-0.0007	1.9080	0.1094	H1-9	0.0698	1.3870	0.0157
C8-1	0.0390	1.9080	0.1094	H4-0	0.1147	1.4090	0.0150
C8-2	0.0486	1.9080	0.1094	H4-1	0.2062	1.4090	0.0150
C8-3	-0.0094	1.9080	0.1094	H5-0	0.1392	1.3590	0.0150
C8-4	0.0187	1.9080	0.1094	HA-0	0.1330	1.4590	0.0150
C8-5	-0.0479	1.9080	0.1094	HA-1	0.1430	1.4590	0.0150
C8-6	-0.0143	1.9080	0.1094	HA-2	0.1297	1.4590	0.0150
CA-0	0.8076	1.9080	0.0860	HA-3	0.1572	1.4590	0.0150
CA-1	0.0118	1.9080	0.0860	HA-4	0.1417	1.4590	0.0150
CA-10	-0.1906	1.9080	0.0860	HA-5	0.1447	1.4590	0.0150
CA-11	-0.2341	1.9080	0.0860	HA-6	0.1700	1.4590	0.0150
CA-2	-0.1256	1.9080	0.0860	HA-7	0.1699	1.4590	0.0150
CA-3	-0.1704	1.9080	0.0860	HA-8	0.1656	1.4590	0.0150
CA-4	-0.1072	1.9080	0.0860	HC-0	0.0603	1.4870	0.0157
CA-5	-0.2601	1.9080	0.0860	HC-1	0.0327	1.4870	0.0157
CA-6	-0.1134	1.9080	0.0860	HC-10	0.0187	1.4870	0.0157
CA-7	-0.1972	1.9080	0.0860	HC-11	0.0882	1.4870	0.0157
CA-8	-0.2387	1.9080	0.0860	HC-12	0.0236	1.4870	0.0157
CA-9	-0.0011	1.9080	0.0860	HC-13	0.0186	1.4870	0.0157
CB-0	0.1243	1.9080	0.0860	HC-14	0.0457	1.4870	0.0157
CC-0	-0.0266	1.9080	0.0860	HC-15	-0.0361	1.4870	0.0157
CN-0	0.138	1.9080	0.0860	HC-16	0.1000	1.4870	0.0157
CO-0	0.7994	1.9080	0.0860	HC-17	0.0362	1.4870	0.0157
CO-1	0.8054	1.9080	0.0860	HC-18	0.0103	1.4870	0.0157
CR-0	0.2057	1.9080	0.0860	HC-19	0.0621	1.4870	0.0157

Table 5.2. (cont'd)

CT-0	-0.1825	1.9080	0.1094	HC-2	0.0285	1.4870	0.0157
CT-1	-0.0462	1.9080	0.1094	HC-20	0.0241	1.4870	0.0157
CT-10	-0.2438	1.9080	0.1094	HC-21	0.0295	1.4870	0.0157
CT-11	-0.005	1.9080	0.1094	HC-22	0.0213	1.4870	0.0157
CT-12	-0.0152	1.9080	0.1094	HC-23	0.0253	1.4870	0.0157
CT-13	-0.3192	1.9080	0.1094	HC-24	0.0642	1.4870	0.0157
CT-2	-0.3204	1.9080	0.1094	HC-25	0.0339	1.4870	0.0157
CT-3	-0.0660	1.9080	0.1094	HC-26	-0.0297	1.4870	0.0157
CT-4	-0.4121	1.9080	0.1094	HC-27	0.0791	1.4870	0.0157
CT-5	-0.0536	1.9080	0.1094	HC-3	0.0797	1.4870	0.0157
CT-6	-0.0343	1.9080	0.1094	HC-4	-0.0122	1.4870	0.0157
CT-7	0.0192	1.9080	0.1094	HC-5	0.0171	1.4870	0.0157
CT-8	0.0189	1.9080	0.1094	HC-6	0.0352	1.4870	0.0157
CT-9	-0.0070	1.9080	0.1094	HC-7	-0.0173	1.4870	0.0157
CV-0	0.1292	1.9080	0.0860	HC-8	-0.0425	1.4870	0.0157
CW-0	-0.1638	1.9080	0.0860	HC-9	0.0402	1.4870	0.0157
CX-0	0.0337	1.9080	0.1094	HO-0	0.4275	0.0000	0.0000
CX-1	-0.2637	1.9080	0.1094	HO-1	0.4102	0.0000	0.0000
CX-10	-0.0518	1.9080	0.1094	HO-2	0.3992	0.0000	0.0000
CX-11	-0.2400	1.9080	0.1094	HP-0	0.1135	1.1000	0.0157
CX-12	-0.0237	1.9080	0.1094	HS-0	0.1933	0.600	0.0157
CX-13	-0.0024	1.9080	0.1094	N-0	-0.4157	1.8240	0.1700
CX-14	-0.0266	1.9080	0.1094	N-1	-0.3479	1.8240	0.1700
CX-15	-0.0249	1.9080	0.1094	N-2	-0.9191	1.8240	0.1700
CX-16	-0.0389	1.9080	0.1094	N-3	-0.5163	1.8240	0.1700
CX-17	-0.0275	1.9080	0.1094	N-4	-0.9407	1.8240	0.1700
CX-18	-0.0014	1.9080	0.1094	N-5	-0.2548	1.8240	0.1700
CX-19	-0.0875	1.9080	0.1094	N2-0	-0.5295	1.8240	0.1700
CX-2	0.0143	1.9080	0.1094	N2-1	-0.8627	1.8240	0.1700
CX-3	0.0381	1.9080	0.1094	N3-0	-0.3854	1.8240	0.1700
CX-4	0.0213	1.9080	0.1094	NA-0	-0.3811	1.8240	0.1700
CX-5	-0.0031	1.9080	0.1094	NA-1	-0.3418	1.8240	0.1700
CX-6	0.0397	1.9080	0.1094	NB-0	-0.5727	1.8240	0.1700
CX-7	-0.0252	1.9080	0.1094	O-0	-0.5679	1.6612	0.2100
CX-8	0.0188	1.9080	0.1094	O-1	-0.5894	1.6612	0.2100
CX-9	-0.0597	1.9080	0.1094	O-2	-0.5931	1.6612	0.2100
H-0	0.2719	0.6000	0.0157	O-3	-0.5819	1.6612	0.2100
H-1	0.2747	0.6000	0.0157	O-4	-0.6086	1.6612	0.2100
H-2	0.3456	0.6000	0.0157	O-5	-0.5748	1.6612	0.2100
H-3	0.4478	0.6000	0.0157	O2-0	-0.8014	1.6612	0.2100
H-4	0.4196	0.6000	0.0157	O2-1	-0.8188	1.6612	0.2100
H-5	0.2936	0.6000	0.0157	OH-0	-0.6546	1.7210	0.2104
H-6	0.4251	0.6000	0.0157	OH-1	-0.6761	1.7210	0.2104
H-7	0.3649	0.6000	0.0157	OH-2	-0.5579	1.7210	0.2104
H-8	0.3400	0.6000	0.0157	S-0	-0.2737	2.0000	0.2500
H-9	0.3412	0.6000	0.0157	SH-0	-0.3119	2.0000	0.2500
H1-0	0.0823	1.3870	0.0157				

Table 5.3. Torsion and nonbond energies calculated by an encoded program with ff94 parameters for single amino acid test set.

	Energies / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA	0.0088	0.5494	45.9693	-0.1578	-24.2543
ARG	0.0706	1.8904	-260.3572	-1.2202	148.0743
ASN	4.1250	1.3423	-23.2339	-0.8248	-39.3539
ASP	0.0194	0.8207	56.3004	-0.4736	-36.3626
CYS	0.0447	0.7164	39.1112	-0.3116	-16.1384
GLN	4.2578	1.3251	-30.2072	-0.8694	-14.0307
GLU	0.0742	1.3344	46.5920	-0.7514	-15.0577
GLY	0.0000	0.4985	37.1013	-0.0511	-21.1957
HID	0.0158	0.5368	23.8454	-1.2102	-10.6580
ILE	0.3316	2.1826	21.2849	-0.0174	-7.2089
LEU	0.0679	2.3109	16.3250	-0.4998	-21.3487
LYS	0.0743	1.6171	58.9714	-0.8348	-1.7579
MET	0.0668	0.8724	36.1038	-0.7017	-17.0426
PHE	0.0120	4.3120	36.2859	-1.3411	-17.6305
PRO	3.1304	0.4955	14.0621	-0.3670	-3.2757
SER	0.0115	0.6387	12.8409	-0.1832	1.0681
THR	0.0783	1.5075	-21.1780	-0.2961	12.0923
TRP	0.0115	3.5231	52.6121	-2.2212	-26.2422
TYR	0.0131	4.2971	24.5879	-1.4562	-23.3515
VAL	0.1042	1.8272	-10.7949	-0.0816	9.0526

Table 5.4. Torsion and nonbond energies calculated by Amber software with ff94 parameters for single amino acid test set.

	Energies / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA	0.0088	0.5492	45.9746	-0.1578	-24.2532
ARG	0.0704	1.8940	-260.3507	-1.2195	148.0715
ASN	4.1252	1.3436	-23.2274	-0.8252	-39.3537
ASP	0.0184	0.8247	56.3181	-0.4730	-36.3870
CYS	0.0446	0.7168	39.1207	-0.3116	-16.1453
GLN	4.2583	1.3287	-30.1907	-0.8689	-14.0280
GLU	0.0740	1.3323	46.5987	-0.7506	-15.0496
GLY	0.0000	0.4957	37.0937	-0.0510	-21.1894
HID	0.0155	0.5387	23.8397	-1.2100	-10.6602
ILE	0.3327	2.1786	21.2916	-0.0195	-7.2120
LEU	0.0676	2.3076	16.3301	-0.5007	-21.3532
LYS	0.0750	1.6158	58.9735	-0.8338	-1.7559
MET	0.0663	0.8775	36.1123	-0.7021	-17.0471
PHE	0.0120	4.3059	36.2908	-1.3421	-17.6348
PRO	3.1301	0.4933	14.0603	-0.3672	-3.2746
SER	0.0115	0.6394	12.8384	-0.1832	1.0766
THR	0.0789	1.5039	-21.1695	-0.2965	12.0833
TRP	0.0123	3.5237	52.6181	-2.2210	-26.2490
TYR	0.0128	4.2906	24.5879	-1.4566	-23.3583
VAL	0.1039	1.8261	-10.7968	-0.0813	9.0583

Table 5.5. Comparisons of torsion and nonbond energies calculated by encoded programs and Amber software with ff94 parameters for single amino acid test set.

	Energy difference / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA	0.0000	0.0002	-0.0053	0.0000	-0.0011
ARG	0.0002	-0.0036	-0.0065	-0.0007	0.0028
ASN	-0.0002	-0.0013	-0.0065	0.0004	-0.0002
ASP	0.0010	-0.0040	-0.0177	-0.0006	0.0244
CYS	0.0001	-0.0004	-0.0095	0.0000	0.0069
GLN	-0.0005	-0.0036	-0.0165	-0.0005	-0.0027
GLU	0.0002	0.0021	-0.0067	-0.0008	-0.0081
GLY	0.0000	0.0028	0.0076	-0.0001	-0.0063
HID	0.0003	-0.0019	0.0057	-0.0002	0.0022
ILE	-0.0011	0.0040	-0.0067	0.0021	0.0031
LEU	0.0003	0.0033	-0.0051	0.0009	0.0045
LYS	-0.0007	0.0013	-0.0021	-0.0010	-0.0020
MET	0.0005	-0.0051	-0.0085	0.0004	0.0045
PHE	0.0000	0.0061	-0.0049	0.0010	0.0043
PRO	0.0003	0.0022	0.0018	0.0002	-0.0011
SER	0.0000	-0.0007	0.0025	0.0000	-0.0085
THR	-0.0006	0.0036	-0.0085	0.0004	0.0090
TRP	-0.0008	-0.0006	-0.0060	-0.0002	0.0068
TYR	0.0003	0.0065	0.0000	0.0004	0.0068
VAL	0.0003	0.0011	0.0019	-0.0003	-0.0057
Maximum	0.0010	0.0065	0.0076	0.0021	0.0244
Minimum	-0.0011	-0.0051	-0.0177	-0.0010	-0.0085

Table 5.6. Torsion and nonbond energies calculated by an encoded program with ff14SB parameters for single amino acid test set.

	Energies / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA	0.1711	0.5419	45.9076	-0.1566	-24.1773
ARG	0.6312	1.9243	-260.1021	-1.2163	147.9333
ASN	10.7889	1.3478	-23.4786	-0.5022	-39.5225
ASP	6.0681	1.3423	58.0559	-0.1236	-38.6990
CYS	2.1838	0.7572	38.8337	-0.3214	-15.8136
GLN	9.2666	1.5997	-30.421	-0.6938	-12.3584
GLU	6.0002	1.5892	46.1863	-0.6777	-14.6146
GLY	0.7558	0.4869	37.0364	-0.0505	-21.1259
HID	4.0438	0.8284	26.7941	-1.0869	-13.7885
ILE	4.7027	2.5636	26.2700	-0.1579	-12.1176
LEU	3.4216	2.3113	16.4199	-0.5225	-21.4754
LYS	1.0358	1.6150	59.0155	-0.8341	-1.7977
MET	3.4048	0.8709	35.9177	-0.7018	-16.8777
PHE	1.2477	4.5331	39.0332	-1.5347	-20.8393
PRO	3.6242	0.4790	14.0622	-0.3663	-3.2798
SER	2.4248	1.0014	14.0655	-0.2804	-0.8490
THR	7.6226	1.5090	-21.5826	-0.2696	12.7759
TRP	1.5419	3.7928	55.3833	-2.3750	-29.7700
TYR	1.5085	4.5045	27.0749	-1.6484	-26.6298
VAL	2.6052	2.2623	-7.2995	-0.2499	5.5357

Table 5.7. Torsion and nonbond energies calculated by Amber software with ff14SB parameters for single amino acid test set.

	Energies / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA	0.1712	0.5424	45.9121	-0.1567	-24.1781
ARG	0.6313	1.9285	-260.0866	-1.2161	147.9086
ASN	10.7881	1.3465	-23.4731	-0.5034	-39.5127
ASP	6.0686	1.3472	58.0653	-0.1221	-38.7120
CYS	2.1834	0.7575	38.8367	-0.3212	-15.8130
GLN	9.2677	1.6053	-30.4211	-0.6948	-12.3676
GLU	5.9995	1.5870	46.1810	-0.6764	-14.6191
GLY	0.7553	0.4867	37.0303	-0.0505	-21.1246
HID	4.0424	0.8261	26.7944	-1.0859	-13.7808
ILE	4.7030	2.5634	26.2721	-0.1615	-12.1235
LEU	3.4221	2.3102	16.4172	-0.5229	-21.4743
LYS	1.0363	1.6139	59.0196	-0.8337	-1.7965
MET	3.4055	0.8690	35.9189	-0.7020	-16.8745
PHE	1.2475	4.5322	39.0335	-1.5350	-20.8375
PRO	3.6209	0.4811	14.0628	-0.3666	-3.2788
SER	2.4255	1.0041	14.0670	-0.2803	-0.8570
THR	7.6225	1.5065	-21.5635	-0.2691	12.7765
TRP	1.5421	3.7865	55.3799	-2.3752	-29.7711
TYR	1.5077	4.5149	27.0864	-1.6483	-26.6331
VAL	2.6041	2.2635	-7.2958	-0.2495	5.5362

Table 5.8. Comparisons of torsion and nonbond energies calculated by encoded programs and Amber software with ff14SB parameters for single amino acid test set.

	Energy difference / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA	-0.0001	-0.0005	-0.0045	0.0001	0.0008
ARG	-0.0001	-0.0042	-0.0155	-0.0002	0.0247
ASN	0.0008	0.0013	-0.0055	0.0012	-0.0098
ASP	-0.0005	-0.0049	-0.0094	-0.0015	0.0130
CYS	0.0004	-0.0003	-0.0030	-0.0002	-0.0006
GLN	-0.0011	-0.0056	0.0001	0.0010	0.0092
GLU	0.0007	0.0022	0.0053	-0.0013	0.0045
GLY	0.0005	0.0002	0.0061	0.0000	-0.0013
HID	0.0014	0.0023	-0.0003	-0.0010	-0.0077
ILE	-0.0003	0.0002	-0.0021	0.0036	0.0059
LEU	-0.0005	0.0011	0.0027	0.0004	-0.0011
LYS	-0.0005	0.0011	-0.0041	-0.0004	-0.0012
MET	-0.0007	0.0019	-0.0012	0.0002	-0.0032
PHE	0.0002	0.0009	-0.0003	0.0003	-0.0018
PRO	0.0033	-0.0021	-0.0006	0.0003	-0.0010
SER	-0.0007	-0.0027	-0.0015	-0.0001	0.0080
THR	0.0001	0.0025	-0.0191	-0.0005	-0.0006
TRP	-0.0002	0.0063	0.0034	0.0002	0.0011
TYR	0.0008	-0.0104	-0.0115	-0.0001	0.0033
VAL	0.0011	-0.0012	-0.0037	-0.0004	-0.0005
Maximum	0.0033	0.0063	0.0061	0.0036	0.0247
Minimum	-0.0011	-0.0104	-0.0191	-0.0015	-0.0098

Table 5.9. Torsion and nonbond energies calculated by an encoded program with ff94 parameters for double amino acid test set.

	Energies / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA_ALA	2.4541	2.4645	118.0755	-1.3252	-96.9829
ALA_ARG	2.5390	3.7265	-155.1786	-2.6131	57.5582
ALA_ASN	6.6826	3.0835	50.9307	-2.1937	-111.2274
ALA_ASP	2.4786	3.0685	139.1431	-1.9733	-122.5850
ALA_CYS	2.4849	2.6440	116.2346	-1.6243	-93.4271
ALA_GLN	6.7149	3.2556	53.4273	-2.2422	-96.5684
ALA_GLU	2.5180	3.2527	132.1015	-2.1393	-105.8521
ALA_GLY	1.6734	2.1374	123.1075	-0.9507	-105.6187
ALA_HID	2.4609	2.6784	106.8670	-2.7616	-95.8874
ALA_ILE	2.7927	4.5115	118.8190	-1.8707	-102.5848
ALA_LEU	2.5165	4.1994	97.7516	-1.9502	-98.8091
ALA_LYS	2.5381	3.4659	161.4802	-2.2229	-90.8739
ALA_MET	2.5177	2.8025	122.6702	-2.0665	-102.5814
ALA_PHE	2.4561	6.4551	121.5178	-2.9508	-103.4251
ALA_PRO	6.6337	2.9762	112.4849	-1.3731	-86.7326
ALA_SER	2.4803	2.8831	109.0765	-1.6101	-95.8092
ALA_THR	2.7028	3.2771	85.3268	-2.0097	-93.7234
ALA_TRP	2.4672	5.4390	139.2381	-3.8377	-109.2930
ALA_TYR	2.4550	6.4463	109.7341	-3.0755	-109.3333
ALA_VAL	2.5765	4.1802	95.8870	-1.9082	-95.3666
ARG_ARG	3.5989	5.0196	-464.1369	-4.2882	250.8879
ARG_ASN	7.6781	4.4892	-271.0586	-3.6803	63.7242
ARG_ASP	3.7294	4.0194	-187.4092	-3.5980	16.2763
ARG_CYS	3.5285	3.9415	-204.7689	-3.2842	80.1727
ARG_GLN	7.7672	4.5536	-265.7064	-3.8918	77.1582
ARG_GLU	3.5508	4.5561	-191.2274	-3.7826	35.4200
ARG_GLY	2.7520	3.3233	-195.2478	-2.4176	66.3483
ARG_HID	3.5023	3.9778	-213.3264	-4.4010	78.3238
ARG_ILE	4.0226	5.4434	-198.7507	-3.5590	67.2681
ARG_LEU	3.5625	5.4637	-221.5235	-3.5774	74.1001
ARG_LYS	3.5979	4.7507	-148.3798	-3.8962	103.1412
ARG_MET	3.5586	4.0921	-195.4928	-3.7156	68.8743
ARG_PHE	3.4939	7.7531	-197.9532	-4.5825	68.5464
ARG_PRO	8.1324	4.0358	-191.2188	-3.5445	73.4977
ARG_SER	3.6510	4.0405	-207.3283	-3.3388	71.8203
ARG_THR	3.9047	4.4901	-227.9708	-3.8193	69.9015

Table 5.9. (cont'd)

ARG_TRP	3.5026	7.0099	-176.9734	-5.6027	55.4917
ARG_TYR	3.4931	7.7412	-209.6550	-4.7163	62.5798
ARG_VAL	3.7927	5.0970	-218.4410	-3.5805	71.2631
ASN_ASN	11.0339	3.8413	-20.3290	-3.0611	-127.1591
ASN_ASP	6.8400	3.6823	68.0288	-3.1432	-135.1406
ASN_CYS	6.8886	3.2317	45.1966	-2.6885	-108.1918
ASN_GLN	11.1264	3.8336	-17.6676	-3.2636	-111.5403
ASN_GLU	6.8894	3.8464	60.8697	-3.2337	-117.7684
ASN_GLY	6.0943	2.5801	51.4823	-1.7582	-119.7158
ASN_HID	6.8701	3.2598	35.7919	-3.8110	-110.8099
ASN_ILE	7.1218	5.1757	47.8864	-3.0779	-117.5689
ASN_LEU	6.9373	4.7856	26.7859	-2.9730	-113.4098
ASN_LYS	6.9685	4.0641	90.7499	-3.1775	-107.9412
ASN_MET	6.9262	3.3775	51.5919	-3.1027	-116.9573
ASN_PHE	6.8602	7.0480	50.4911	-4.0217	-117.9304
ASN_PRO	12.7024	3.3361	42.5972	-3.2959	-105.3569
ASN_SER	6.8498	3.4902	38.0614	-2.7114	-110.4575
ASN_THR	7.1053	3.8595	14.5698	-2.9901	-110.8819
ASN_TRP	6.9157	6.0046	68.1078	-4.7112	-127.2461
ASN_TYR	6.9122	6.8035	36.8430	-3.7995	-122.2596
ASN_VAL	6.9041	4.8457	25.0682	-3.0967	-110.4704
ASP_ASP	3.1607	3.2354	144.2870	-3.4380	-78.8422
ASP_CYS	3.1835	2.8171	122.5938	-2.9579	-101.4878
ASP_GLN	7.3641	3.4838	58.9727	-3.5224	-106.3848
ASP_GLU	3.1665	3.4410	136.4192	-3.4401	-69.0539
ASP_GLY	2.0169	2.2660	126.9188	-1.1338	-116.5652
ASP_HID	3.1247	2.8672	112.6631	-4.0776	-105.7256
ASP_ILE	3.5392	4.6492	123.9353	-3.2977	-105.8915
ASP_LEU	2.9222	4.5808	103.5540	-2.6487	-108.6484
ASP_LYS	2.9092	3.8309	166.3842	-2.7405	-137.4003
ASP_MET	3.1749	3.0066	128.0064	-3.3565	-110.0100
ASP_PHE	3.1360	6.6599	127.2950	-4.3028	-110.1849
ASP_PRO	8.0789	3.2402	115.2603	-2.5744	-79.6765
ASP_SER	3.1873	3.0302	113.4443	-2.9438	-99.9177
ASP_THR	2.9886	3.6904	87.1147	-2.4968	-100.6664
ASP_TRP	3.1386	5.8814	146.8720	-5.3003	-120.3027
ASP_TYR	3.1347	6.6239	115.4293	-4.4321	-115.9908

Table 5.9. (cont'd)

ASP_VAL	3.3172	4.3338	99.9173	-3.3141	-97.6671
CYS_CYS	2.9858	2.5885	106.2667	-1.6170	-82.7387
CYS_GLN	7.1923	3.2100	43.3146	-2.2132	-86.1207
CYS_GLU	3.0456	3.2067	121.0599	-2.1566	-92.1949
CYS_GLY	2.1382	2.0243	112.8776	-0.8067	-94.5699
CYS_HID	2.9452	2.6232	96.8071	-2.7539	-85.4161
CYS_ILE	3.3271	4.4902	108.9223	-1.9375	-91.9022
CYS_LEU	3.0051	4.1647	87.7933	-1.9280	-88.1488
CYS_LYS	3.0124	3.4130	152.2366	-2.1891	-82.3759
CYS_MET	3.0035	2.7517	112.5730	-2.0482	-91.7585
CYS_PHE	2.9505	6.4087	111.4804	-2.9572	-92.6200
CYS_PRO	6.8531	3.1440	111.6524	-1.7290	-82.1681
CYS_SER	2.9942	2.8136	98.9303	-1.6165	-84.8319
CYS_THR	3.1399	3.2513	75.0342	-1.9495	-83.5785
CYS_TRP	2.9437	5.3702	128.8096	-3.7325	-99.0577
CYS_TYR	2.9509	6.3918	99.6718	-3.0864	-98.5395
CYS_VAL	3.1088	4.1469	85.9137	-1.9688	-84.6144
GLN_GLN	12.0125	3.8958	-29.4666	-3.4373	-79.1768
GLN_GLU	7.7871	3.8965	47.4288	-3.3190	-88.8434
GLN_GLY	6.9803	2.7353	40.1925	-2.0050	-88.5248
GLN_HID	7.8438	3.1434	22.3377	-3.9581	-74.7281
GLN_ILE	8.1410	4.7107	34.3822	-2.9965	-82.7077
GLN_LEU	7.8086	4.8222	14.9122	-3.1315	-81.6610
GLN_LYS	7.8463	4.1136	80.2541	-3.4401	-73.8852
GLN_MET	7.8095	3.4372	39.8187	-3.2595	-85.3447
GLN_PHE	7.8348	6.8875	36.9260	-4.1157	-83.3885
GLN_PRO	12.2978	3.4309	32.6702	-3.0442	-70.9782
GLN_SER	7.8505	3.3561	25.9878	-2.8537	-77.7492
GLN_THR	8.0483	3.8781	3.1903	-3.3084	-78.8237
GLN_TRP	7.8317	6.0764	56.6957	-5.1470	-94.4435
GLN_TYR	7.8354	6.8930	25.4572	-4.2521	-89.4582
GLN_VAL	7.9239	4.3761	12.1612	-3.0337	-76.3121
GLU_GLU	3.6441	3.6768	120.8392	-3.1749	-50.3009
GLU_GLY	2.7831	2.5925	113.6088	-1.8125	-85.6407
GLU_HID	3.5662	3.1174	97.7211	-3.7862	-76.3207
GLU_ILE	3.9177	4.9371	108.8633	-2.9739	-79.8023
GLU_LEU	3.6198	4.6441	88.9800	-2.9597	-78.8607

Table 5.9. (cont'd)

GLU_LYS	3.6384	3.9212	152.6513	-3.2500	-100.1574
GLU_MET	3.6275	3.2404	113.1587	-3.0892	-82.0002
GLU_PHE	3.5659	6.8793	112.2930	-3.9829	-82.5193
GLU_PRO	8.0638	3.2750	104.4112	-2.7713	-63.5008
GLU_SER	3.6084	3.3089	98.6877	-2.6488	-72.9980
GLU_THR	3.8444	3.7322	73.5535	-3.0288	-68.0000
GLU_TRP	3.6527	5.8363	129.3251	-4.9669	-84.5115
GLU_TYR	3.5642	6.8681	100.4585	-4.1154	-88.2992
GLU_VAL	3.7021	4.5986	85.0392	-2.9978	-71.8262
GLY_GLY	1.5000	1.6585	102.8303	-0.6707	-92.2244
GLY_HID	2.2982	1.9844	85.8240	-2.3846	-81.1907
GLY_ILE	2.6346	3.8167	98.1373	-1.4559	-88.3262
GLY_LEU	2.3558	3.5112	77.0045	-1.5708	-84.4465
GLY_LYS	2.3749	2.7742	143.0126	-1.8421	-78.7054
GLY_MET	2.3556	2.1040	101.7901	-1.6939	-88.0633
GLY_PHE	2.2971	5.7707	100.5821	-2.5727	-88.8199
GLY_PRO	5.9381	2.5873	96.9794	-1.0577	-77.4042
GLY_SER	2.3166	2.1843	88.2247	-1.2364	-81.3288
GLY_THR	2.5377	2.5896	64.4826	-1.6198	-79.3192
GLY_TRP	2.2977	4.7539	118.2816	-3.4389	-94.5748
GLY_TYR	2.2959	5.7412	88.7830	-2.6974	-94.7276
GLY_VAL	2.4186	3.4758	75.3282	-1.4954	-81.2503
HID_HID	2.7411	2.7769	82.2093	-5.3569	-80.8651
HID_ILE	3.3192	4.5185	94.0478	-3.7986	-91.7020
HID_LEU	2.7948	4.3076	73.2120	-4.5351	-83.5880
HID_LYS	2.8364	3.5823	138.0382	-4.7869	-77.3242
HID_MET	2.7975	2.9080	98.0181	-4.6417	-87.0351
HID_PHE	2.7464	6.5656	96.8807	-5.5492	-88.2746
HID_PRO	8.2597	2.8998	90.6641	-4.2738	-81.9202
HID_SER	2.7973	2.9601	84.2997	-4.1701	-80.5401
HID_THR	3.0268	3.5179	62.7066	-4.5594	-85.3626
HID_TRP	3.0908	5.8468	116.3558	-5.7944	-103.4417
HID_TYR	2.7466	6.5465	85.0769	-5.6854	-94.2045
HID_VAL	2.9019	3.9920	70.9571	-4.1012	-83.0132
ILE_ILE	3.9864	6.1954	84.5272	-2.7266	-77.4666
ILE_LEU	3.7072	5.8961	63.4315	-2.7114	-73.6186
ILE_LYS	3.7414	5.1632	130.3178	-3.0124	-68.8156

Table 5.9. (cont'd)

ILE_MET	3.7185	4.4992	88.1964	-2.8384	-77.2333
ILE_PHE	3.6532	8.1476	86.8547	-3.7223	-77.8879
ILE_PRO	8.4893	4.3784	84.7914	-2.4266	-67.7552
ILE_SER	3.7002	4.5634	74.5719	-2.3869	-70.5831
ILE_THR	3.9212	4.9769	51.2479	-2.8167	-68.8712
ILE_TRP	3.6978	7.1315	104.8627	-4.6951	-84.3497
ILE_TYR	3.6557	8.1320	75.0594	-3.8572	-83.7759
ILE_VAL	3.7748	5.8484	61.7970	-2.7504	-70.4904
LEU_LEU	3.5751	5.7919	63.1293	-3.1584	-93.7915
LEU_LYS	3.6360	5.0654	128.5000	-3.4701	-87.1845
LEU_MET	3.5843	4.3928	87.9066	-3.2875	-97.4388
LEU_PHE	3.5154	8.0419	86.5862	-4.1762	-98.1084
LEU_PRO	8.5188	4.4980	80.8828	-2.9783	-83.8714
LEU_SER	3.5617	4.4836	74.3449	-2.8578	-90.8153
LEU_THR	3.8547	4.8451	51.0380	-3.2704	-89.1047
LEU_TRP	4.6167	6.7250	102.9661	-4.9713	-102.9845
LEU_TYR	3.5145	8.0406	74.8016	-4.3098	-103.9483
LEU_VAL	3.5938	5.7632	61.4935	-3.2261	-90.6281
LYS_LYS	3.5837	4.4076	171.7248	-3.4583	-46.3914
LYS_MET	3.5439	3.7477	125.0376	-3.2698	-80.5089
LYS_PHE	3.6336	7.2059	121.1888	-4.1676	-79.4678
LYS_PRO	8.1071	3.7199	128.3801	-3.1028	-76.3896
LYS_SER	3.5235	3.8554	113.4248	-2.8400	-77.7344
LYS_THR	3.8315	4.1794	92.4664	-3.3389	-79.9144
LYS_TRP	3.6313	6.3921	141.9639	-5.1972	-91.5786
LYS_TYR	3.4805	7.3876	110.9817	-4.2773	-86.8944
LYS_VAL	3.6041	5.1130	102.6601	-3.1837	-79.3747
MET_MET	3.5960	2.9323	103.7658	-3.1716	-87.6575
MET_PHE	3.6690	6.4674	101.1667	-4.1196	-85.1823
MET_PRO	8.0672	2.9715	97.8373	-3.0239	-75.7004
MET_SER	3.5540	3.0426	90.2650	-2.7477	-81.1390
MET_THR	3.8860	3.3612	66.9389	-3.2329	-79.1961
MET_TRP	3.6729	5.6313	120.8851	-5.1474	-95.9502
MET_TYR	3.5259	6.5691	90.6895	-4.1833	-94.2637
MET_VAL	3.6244	4.3085	77.2636	-3.0982	-80.8930
PHE_PHE	2.9438	10.1367	107.2588	-6.0510	-92.2948
PHE_PRO	7.5244	6.5883	101.7166	-4.9261	-80.4459

Table 5.9. (cont'd)

PHE_SER	3.0410	6.4996	94.6164	-4.6578	-84.3437
PHE_THR	2.9420	7.1106	70.5430	-4.8268	-84.2807
PHE_TRP	2.6765	9.1694	123.8937	-6.5484	-98.3989
PHE_TYR	2.6793	9.9424	92.6308	-5.6331	-93.3411
PHE_VAL	3.1880	7.7654	81.6957	-5.0487	-84.0391
PRO_PRO	9.4988	2.9215	73.0478	-2.0934	-57.4202
PRO_SER	5.5616	2.7334	63.5137	-2.2047	-60.4104
PRO_THR	5.7983	3.1318	39.6713	-2.6288	-58.0718
PRO_TRP	5.5485	5.2737	93.7747	-4.4444	-73.7855
PRO_TYR	5.5344	6.2878	64.1343	-3.6779	-73.8010
PRO_VAL	5.6571	4.0296	50.5453	-2.5153	-60.2468
SER_SER	2.7780	2.8641	62.7424	-1.3467	-58.8199
SER_THR	2.8729	3.3333	38.5876	-1.7058	-57.2738
SER_TRP	2.7289	5.4444	92.3778	-3.4552	-72.8540
SER_TYR	2.7434	6.4582	63.4548	-2.8300	-72.4031
SER_VAL	2.8921	4.2053	49.9412	-1.6923	-58.8025
THR_THR	2.9842	4.3523	-0.0002	-2.5669	-32.3897
THR_TRP	2.7521	6.5861	54.2923	-4.4176	-48.8353
THR_TYR	2.6783	7.5370	24.0810	-3.6018	-48.1746
THR_VAL	2.8136	5.2687	10.7400	-2.4526	-34.8625
TRP_TRP	2.6206	8.4303	138.7384	-7.5334	-108.2350
TRP_TYR	2.6155	9.2108	107.4267	-6.6273	-103.1310
TRP_VAL	2.6460	6.7610	94.2838	-5.5834	-90.5809
TYR_TYR	2.6631	9.8961	80.3178	-5.8225	-99.2140
TYR_VAL	3.1168	7.7432	69.4624	-5.3466	-89.2782
VAL_VAL	3.5552	5.4546	21.9492	-2.6268	-46.6209

Table 5.10. Torsion and nonbond energies calculated by Amber software with ff94 parameters for double amino acid test set.

	Energy difference / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA_ALA	2.4544	2.4627	118.0785	-1.3248	-96.9775
ALA_ARG	2.5391	3.7338	-155.1922	-2.6129	57.5772
ALA_ASN	6.6843	3.0813	50.9371	-2.1954	-111.2287
ALA_ASP	2.4792	3.0652	139.1533	-1.9718	-122.5947
ALA_CYS	2.4851	2.6409	116.2341	-1.6264	-93.4288
ALA_GLN	6.7146	3.2590	53.4295	-2.2435	-96.5693
ALA_GLU	2.5170	3.2510	132.1055	-2.1362	-105.8683
ALA_GLY	1.6730	2.1359	123.1055	-0.9506	-105.6109
ALA_HID	2.4599	2.6777	106.8644	-2.7612	-95.8808
ALA_ILE	2.7934	4.5113	118.8296	-1.8709	-102.5990
ALA_LEU	2.5162	4.2023	97.7498	-1.9472	-98.8072
ALA_LYS	2.5390	3.4660	161.4899	-2.2236	-90.8720
ALA_MET	2.5178	2.7988	122.6631	-2.0685	-102.5792
ALA_PHE	2.4571	6.4543	121.5168	-2.9501	-103.4290
ALA_PRO	6.6365	2.9748	112.4853	-1.3720	-86.7384
ALA_SER	2.4817	2.8791	109.0744	-1.6111	-95.8011
ALA_THR	2.7032	3.2769	85.3299	-2.0119	-93.7124
ALA_TRP	2.4685	5.4374	139.2497	-3.8380	-109.2967
ALA_TYR	2.4567	6.4398	109.7249	-3.0767	-109.3268
ALA_VAL	2.5777	4.1744	95.8936	-1.9052	-95.3793
ARG_ARG	3.5997	5.0160	-464.2176	-4.2875	250.9398
ARG_ASN	7.6774	4.4885	-271.0645	-3.6819	63.7411
ARG_ASP	3.7284	4.0150	-187.3786	-3.5989	16.2760
ARG_CYS	3.5283	3.9402	-204.7583	-3.2870	80.1704
ARG_GLN	7.7688	4.5507	-265.7263	-3.8927	77.1420
ARG_GLU	3.5495	4.5547	-191.1995	-3.7789	35.4041
ARG_GLY	2.7524	3.3212	-195.2521	-2.4169	66.3516
ARG_HID	3.5017	3.9840	-213.3195	-4.4024	78.3416
ARG_ILE	4.0238	5.4497	-198.7196	-3.5621	67.2394
ARG_LEU	3.5602	5.4658	-221.4991	-3.5768	74.0706
ARG_LYS	3.5993	4.7538	-148.3877	-3.9016	103.1335
ARG_MET	3.5566	4.0948	-195.5267	-3.7152	68.8839
ARG_PHE	3.4919	7.7516	-197.9175	-4.5855	68.4979
ARG_PRO	8.1292	4.0352	-191.2415	-3.5461	73.5166
ARG_SER	3.6525	4.0429	-207.3750	-3.3379	71.8319
ARG_THR	3.9021	4.4818	-227.9744	-3.8195	69.8984
ARG_TRP	3.5026	7.0102	-177.0093	-5.6061	55.4896
ARG_TYR	3.4934	7.7420	-209.6527	-4.7161	62.5689
ARG_VAL	3.7929	5.1030	-218.4348	-3.5826	71.2634
ASN_ASN	11.0334	3.8400	-20.3347	-3.0666	-127.1316
ASN_ASP	6.8370	3.6817	68.0145	-3.1457	-135.1461
ASN_CYS	6.8863	3.2332	45.2105	-2.6877	-108.2019
ASN_GLN	11.1274	3.8422	-17.6776	-3.2635	-111.5313
ASN_GLU	6.8896	3.8416	60.8696	-3.2354	-117.7836
ASN_GLY	6.0926	2.5825	51.4769	-1.7563	-119.7288
ASN_HID	6.8698	3.2619	35.7755	-3.8100	-110.8445
ASN_ILE	7.1217	5.1775	47.8983	-3.0761	-117.5813
ASN_LEU	6.9374	4.7896	26.7930	-2.9731	-113.4332
ASN_LYS	6.9672	4.0655	90.7454	-3.1776	-107.9458

Table 5.10. (cont'd)

ASN_MET	6.9241	3.3836	51.5909	-3.1037	-116.9608
ASN_PHE	6.8606	7.0463	50.4930	-4.0226	-117.9426
ASN_PRO	12.7031	3.3373	42.6040	-3.3021	-105.3404
ASN_SER	6.8512	3.4881	38.0507	-2.7182	-110.4764
ASN_THR	7.1072	3.8617	14.5607	-2.9917	-110.8854
ASN_TRP	6.9151	6.0066	68.1072	-4.7071	-127.2454
ASN_TYR	6.9142	6.7951	36.8455	-3.8035	-122.2804
ASN_VAL	6.9034	4.8491	25.0748	-3.0930	-110.4792
ASP_ASP	3.1600	3.2306	144.2811	-3.4409	-78.8267
ASP_CYS	3.1859	2.8250	122.5939	-2.9569	-101.5003
ASP_GLN	7.3615	3.4733	58.9701	-3.5203	-106.3775
ASP_GLU	3.1683	3.4438	136.4099	-3.4423	-69.0742
ASP_GLY	2.0171	2.2736	126.9328	-1.1311	-116.5589
ASP_HID	3.1246	2.8712	112.6762	-4.0788	-105.7152
ASP_ILE	3.5394	4.6457	123.9222	-3.2958	-105.8612
ASP_LEU	2.9237	4.5787	103.5521	-2.6510	-108.6535
ASP_LYS	2.9100	3.8307	166.3765	-2.7342	-137.4193
ASP_MET	3.1772	3.0039	127.9881	-3.3566	-109.9989
ASP_PHE	3.1335	6.6542	127.2905	-4.3022	-110.1798
ASP_PRO	8.0744	3.2427	115.2702	-2.5702	-79.6948
ASP_SER	3.1877	3.0286	113.4609	-2.9425	-99.9249
ASP_THR	2.9870	3.6916	87.0965	-2.4957	-100.6566
ASP_TRP	3.1385	5.8781	146.8618	-5.2980	-120.2735
ASP_TYR	3.1348	6.6333	115.4404	-4.4336	-115.9782
ASP_VAL	3.3180	4.3218	99.9057	-3.3178	-97.6521
CYS_CYS	2.9818	2.5847	106.2724	-1.6156	-82.7409
CYS_GLN	7.1915	3.2086	43.3375	-2.2185	-86.1317
CYS_GLU	3.0453	3.2110	121.0603	-2.1576	-92.1871
CYS_GLY	2.1393	2.0234	112.8800	-0.8104	-94.5822
CYS_HID	2.9454	2.6218	96.7936	-2.7541	-85.4031
CYS_ILE	3.3274	4.4852	108.9150	-1.9379	-91.8907
CYS_LEU	3.0044	4.1645	87.7902	-1.9265	-88.1597
CYS_LYS	3.0131	3.4162	152.2474	-2.1883	-82.3812
CYS_MET	3.0019	2.7471	112.5761	-2.0484	-91.7541
CYS_PHE	2.9505	6.4050	111.4871	-2.9550	-92.6174
CYS_PRO	6.8521	3.1381	111.6451	-1.7317	-82.1610
CYS_SER	2.9935	2.8133	98.9338	-1.6176	-84.8356
CYS_THR	3.1392	3.2624	75.0432	-1.9480	-83.5826
CYS_TRP	2.9426	5.3762	128.8149	-3.7309	-99.0455
CYS_TYR	2.9525	6.3886	99.6839	-3.0857	-98.5499
CYS_VAL	3.1089	4.1492	85.9173	-1.9668	-84.6098
GLN_GLN	12.0110	3.8942	-29.4624	-3.4372	-79.1611
GLN_GLU	7.7898	3.8917	47.4497	-3.3223	-88.8583
GLN_GLY	6.9762	2.7326	40.1859	-2.0034	-88.5262
GLN_HID	7.8450	3.1454	22.3377	-3.9603	-74.7360
GLN_ILE	8.1417	4.7129	34.3919	-2.9997	-82.7214
GLN_LEU	7.8088	4.8238	14.9147	-3.1325	-81.6566
GLN_LYS	7.8437	4.1130	80.2583	-3.4370	-73.8935
GLN_MET	7.8068	3.4379	39.8478	-3.2607	-85.3444
GLN_PHE	7.8342	6.8882	36.9509	-4.1175	-83.3861
GLN_PRO	12.3028	3.4305	32.6720	-3.0451	-70.9599

Table 5.10. (cont'd)

GLN_SER	7.8519	3.3569	25.9844	-2.8528	-77.7516
GLN_THR	8.0497	3.8803	3.1854	-3.3087	-78.8275
GLN_TRP	7.8325	6.0791	56.7260	-5.1509	-94.4490
GLN_TYR	7.8369	6.8984	25.4729	-4.2536	-89.4721
GLN_VAL	7.9232	4.3765	12.1859	-3.0318	-76.3078
GLU_GLU	3.6440	3.6797	120.8308	-3.1756	-50.2902
GLU_GLY	2.7825	2.5935	113.5969	-1.8134	-85.6298
GLU_HID	3.5660	3.1174	97.7237	-3.7887	-76.3215
GLU_ILE	3.9172	4.9340	108.8781	-2.9737	-79.8164
GLU_LEU	3.6213	4.6447	88.9770	-2.9630	-78.8664
GLU_LYS	3.6432	3.9187	152.6442	-3.2493	-100.1717
GLU_MET	3.6271	3.2411	113.1626	-3.0886	-82.0180
GLU_PHE	3.5658	6.8935	112.2770	-3.9848	-82.5250
GLU_PRO	8.0626	3.2689	104.3902	-2.7755	-63.4770
GLU_SER	3.6104	3.3156	98.6891	-2.6492	-73.0100
GLU_THR	3.8443	3.7273	73.5461	-3.0266	-67.9961
GLU_TRP	3.6489	5.8355	129.3415	-4.9667	-84.4984
GLU_TYR	3.5664	6.8780	100.4523	-4.1151	-88.2858
GLU_VAL	3.7046	4.6008	85.0406	-2.9988	-71.8135
GLY_GLY	1.5000	1.6571	102.8327	-0.6700	-92.2307
GLY_HID	2.2986	1.9853	85.8342	-2.3853	-81.1880
GLY_ILE	2.6343	3.8187	98.1352	-1.4533	-88.3262
GLY_LEU	2.3559	3.5132	77.0051	-1.5720	-84.4497
GLY_LYS	2.3755	2.7771	143.0221	-1.8434	-78.7119
GLY_MET	2.3558	2.1078	101.7965	-1.6936	-88.0565
GLY_PHE	2.2967	5.7624	100.5767	-2.5726	-88.8228
GLY_PRO	5.9399	2.5833	96.9601	-1.0596	-77.3908
GLY_SER	2.3163	2.1858	88.2293	-1.2359	-81.3223
GLY_THR	2.5366	2.5890	64.4655	-1.6185	-79.3202
GLY_TRP	2.2977	4.7396	118.2748	-3.4395	-94.5778
GLY_TYR	2.2958	5.7470	88.7782	-2.6979	-94.7217
GLY_VAL	2.4178	3.4846	75.3170	-1.4933	-81.2464
HID_HID	2.7414	2.7761	82.1977	-5.3610	-80.8640
HID_ILE	3.3216	4.5215	94.0431	-3.8031	-91.7022
HID_LEU	2.7939	4.3138	73.2260	-4.5333	-83.5980
HID_LYS	2.8377	3.5796	138.0366	-4.7862	-77.3246
HID_MET	2.7976	2.8991	97.9979	-4.6444	-87.0275
HID_PHE	2.7464	6.5613	96.8854	-5.5503	-88.2734
HID_PRO	8.2695	2.9031	90.6678	-4.2754	-81.9213
HID_SER	2.7989	2.9605	84.2969	-4.1717	-80.5334
HID_THR	3.0262	3.5205	62.7036	-4.5587	-85.3689
HID_TRP	3.0918	5.8445	116.3516	-5.7936	-103.4272
HID_TYR	2.7477	6.5433	85.0745	-5.6842	-94.2101
HID_VAL	2.9015	3.9975	70.9666	-4.0980	-83.0187
ILE_ILE	3.9893	6.1898	84.5022	-2.7263	-77.4548
ILE_LEU	3.7068	5.8945	63.4259	-2.7123	-73.6242
ILE_LYS	3.7416	5.1646	130.3269	-3.0102	-68.8154
ILE_MET	3.7173	4.4936	88.1951	-2.8379	-77.2311
ILE_PHE	3.6519	8.1484	86.8631	-3.7280	-77.8917
ILE_PRO	8.4908	4.3771	84.7955	-2.4262	-67.7631
ILE_SER	3.6965	4.5701	74.5959	-2.3879	-70.5793

Table 5.10. (cont'd)

ILE_THR	3.9209	4.9757	51.2420	-2.8164	-68.8646
ILE_TRP	3.7009	7.1316	104.8576	-4.6936	-84.3266
ILE_TYR	3.6532	8.1342	75.0667	-3.8582	-83.7737
ILE_VAL	3.7771	5.8522	61.7753	-2.7475	-70.4835
LEU_LEU	3.5758	5.7878	63.1173	-3.1598	-93.7925
LEU_LYS	3.6350	5.0664	128.4960	-3.4687	-87.1873
LEU_MET	3.5843	4.3960	87.9149	-3.2904	-97.4306
LEU_PHE	3.5127	8.0506	86.5882	-4.1764	-98.1057
LEU_PRO	8.5212	4.4980	80.8836	-2.9817	-83.8724
LEU_SER	3.5623	4.4851	74.3365	-2.8558	-90.8023
LEU_THR	3.8552	4.8466	51.0338	-3.2705	-89.1112
LEU_TRP	4.6172	6.7301	102.9716	-4.9758	-102.9842
LEU_TYR	3.5157	8.0369	74.7967	-4.3086	-103.9834
LEU_VAL	3.5903	5.7649	61.4958	-3.2258	-90.6297
LYS_LYS	3.5815	4.4049	171.7302	-3.4559	-46.4067
LYS_MET	3.5450	3.7468	125.0406	-3.2716	-80.5012
LYS_PHE	3.6350	7.2086	121.1929	-4.1695	-79.4669
LYS_PRO	8.1075	3.7240	128.3883	-3.1022	-76.4028
LYS_SER	3.5211	3.8529	113.4396	-2.8401	-77.7377
LYS_THR	3.8299	4.1699	92.4868	-3.3378	-79.8854
LYS_TRP	3.6354	6.3872	141.9460	-5.1986	-91.5772
LYS_TYR	3.4809	7.3927	110.9881	-4.2775	-86.8880
LYS_VAL	3.6019	5.1101	102.6591	-3.1806	-79.3695
MET_MET	3.5965	2.9354	103.7696	-3.1733	-87.6672
MET_PHE	3.6705	6.4660	101.1788	-4.1244	-85.1871
MET_PRO	8.0664	2.9726	97.8380	-3.0201	-75.6937
MET_SER	3.5577	3.0452	90.2541	-2.7482	-81.1389
MET_THR	3.8853	3.3582	66.9378	-3.2337	-79.1920
MET_TRP	3.6734	5.6287	120.8928	-5.1500	-95.9500
MET_TYR	3.5277	6.5796	90.6851	-4.1868	-94.2762
MET_VAL	3.6268	4.3164	77.2806	-3.0987	-80.9085
PHE_PHE	2.9417	10.1371	107.2569	-6.0531	-92.3045
PHE_PRO	7.5262	6.5884	101.7216	-4.9263	-80.4541
PHE_SER	3.0406	6.4960	94.6190	-4.6570	-84.3487
PHE_THR	2.9373	7.1091	70.5480	-4.8241	-84.2736
PHE_TRP	2.6793	9.1659	123.9031	-6.5492	-98.3957
PHE_TYR	2.6789	9.9335	92.6355	-5.6323	-93.3557
PHE_VAL	3.1874	7.7650	81.6879	-5.0488	-84.0534
PRO_PRO	9.4985	2.9231	73.0407	-2.0938	-57.4253
PRO_SER	5.5604	2.7348	63.5045	-2.2033	-60.4158
PRO_THR	5.7972	3.1312	39.6629	-2.6273	-58.0726
PRO_TRP	5.5493	5.2889	93.7669	-4.4449	-73.7840
PRO_TYR	5.5369	6.2911	64.1512	-3.6784	-73.8046
PRO_VAL	5.6553	4.0326	50.5590	-2.5144	-60.2538
SER_SER	2.7792	2.8633	62.7370	-1.3553	-58.8400
SER_THR	2.8733	3.3342	38.5763	-1.7091	-57.2605
SER_TRP	2.7291	5.4406	92.3879	-3.4544	-72.8563
SER_TYR	2.7413	6.4489	63.4493	-2.8300	-72.4056
SER_VAL	2.8957	4.2003	49.9344	-1.6929	-58.7845
THR_THR	2.9849	4.3608	0.0107	-2.5675	-32.4047
THR_TRP	2.7528	6.5830	54.2738	-4.4163	-48.8367

Table 5.10. (cont'd)

THR_TYR	2.6785	7.5363	24.0798	-3.6039	-48.1608
THR_VAL	2.8117	5.2668	10.7514	-2.4545	-34.8742
TRP_TRP	2.6185	8.4391	138.7433	-7.5365	-108.2237
TRP_TYR	2.6174	9.2155	107.4061	-6.6271	-103.1196
TRP_VAL	2.6448	6.7645	94.2846	-5.5814	-90.5867
TYR_TYR	2.6629	9.8953	80.3259	-5.8205	-99.2277
TYR_VAL	3.1148	7.7502	69.4697	-5.3431	-89.2894
VAL_VAL	3.5552	5.4500	21.9507	-2.6293	-46.6316

Table 5.11. Comparisons of torsion and nonbond energies calculated by the encoded program and Amber software with ff94 parameters for double amino acid test set.

	Energy difference / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA_ALA	-0.0003	0.0018	-0.0030	-0.0004	-0.0054
ALA_ARG	-0.0001	-0.0073	0.0136	-0.0002	-0.0190
ALA_ASN	-0.0017	0.0022	-0.0064	0.0017	0.0013
ALA_ASP	-0.0006	0.0033	-0.0102	-0.0015	0.0097
ALA_CYS	-0.0002	0.0031	0.0005	0.0021	0.0017
ALA_GLN	0.0003	-0.0034	-0.0022	0.0013	0.0009
ALA_GLU	0.0010	0.0017	-0.0040	-0.0031	0.0162
ALA_GLY	0.0004	0.0015	0.0020	-0.0001	-0.0078
ALA_HID	0.0010	0.0007	0.0026	-0.0004	-0.0066
ALA_ILE	-0.0007	0.0002	-0.0106	0.0002	0.0142
ALA_LEU	0.0003	-0.0029	0.0018	-0.0030	-0.0019
ALA_LYS	-0.0009	-0.0001	-0.0097	0.0007	-0.0019
ALA_MET	-0.0001	0.0037	0.0071	0.0020	-0.0022
ALA_PHE	-0.0010	0.0008	0.0010	-0.0007	0.0039
ALA_PRO	-0.0028	0.0014	-0.0004	-0.0011	0.0058
ALA_SER	-0.0014	0.0040	0.0021	0.0010	-0.0081
ALA_THR	-0.0004	0.0002	-0.0031	0.0022	-0.0110
ALA_TRP	-0.0013	0.0016	-0.0116	0.0003	0.0037
ALA_TYR	-0.0017	0.0065	0.0092	0.0012	-0.0065
ALA_VAL	-0.0012	0.0058	-0.0066	-0.0030	0.0127
ARG_ARG	-0.0008	0.0036	0.0807	-0.0007	-0.0519
ARG_ASN	0.0007	0.0007	0.0059	0.0016	-0.0169
ARG_ASP	0.0010	0.0044	-0.0306	0.0009	0.0003
ARG_CYS	0.0002	0.0013	-0.0106	0.0028	0.0023
ARG_GLN	-0.0016	0.0029	0.0199	0.0009	0.0162
ARG_GLU	0.0013	0.0014	-0.0279	-0.0037	0.0159
ARG_GLY	-0.0004	0.0021	0.0043	-0.0007	-0.0033
ARG_HID	0.0006	-0.0062	-0.0069	0.0014	-0.0178
ARG_ILE	-0.0012	-0.0063	-0.0311	0.0031	0.0287
ARG_LEU	0.0023	-0.0021	-0.0244	-0.0006	0.0295
ARG_LYS	-0.0014	-0.0031	0.0079	0.0054	0.0077
ARG_MET	0.0020	-0.0027	0.0339	-0.0004	-0.0096
ARG_PHE	0.0020	0.0015	-0.0357	0.0030	0.0485
ARG_PRO	0.0032	0.0006	0.0227	0.0016	-0.0189
ARG_SER	-0.0015	-0.0024	0.0467	-0.0009	-0.0116
ARG_THR	0.0026	0.0083	0.0036	0.0002	0.0031
ARG_TRP	0.0000	-0.0003	0.0359	0.0034	0.0021
ARG_TYR	-0.0003	-0.0008	-0.0023	-0.0002	0.0109
ARG_VAL	-0.0002	-0.0060	-0.0062	0.0021	-0.0003
ASN_ASN	0.0005	0.0013	0.0057	0.0055	-0.0275
ASN_ASP	0.0030	0.0006	0.0143	0.0025	0.0055
ASN_CYS	0.0023	-0.0015	-0.0139	-0.0008	0.0101
ASN_GLN	-0.0010	-0.0086	0.0100	-0.0001	-0.0090
ASN_GLU	-0.0002	0.0048	0.0001	0.0017	0.0152
ASN_GLY	0.0017	-0.0024	0.0054	-0.0019	0.0130
ASN_HID	0.0003	-0.0021	0.0164	-0.0010	0.0346
ASN_ILE	0.0001	-0.0018	-0.0119	-0.0018	0.0124
ASN_LEU	-0.0001	-0.0040	-0.0071	0.0001	0.0234
ASN_LYS	0.0013	-0.0014	0.0045	0.0001	0.0046

Table 5.11. (cont'd)

ASN_MET	0.0021	-0.0061	0.0010	0.0010	0.0035
ASN_PHE	-0.0004	0.0017	-0.0019	0.0009	0.0122
ASN_PRO	-0.0007	-0.0012	-0.0068	0.0062	-0.0165
ASN_SER	-0.0014	0.0021	0.0107	0.0068	0.0189
ASN_THR	-0.0019	-0.0022	0.0091	0.0016	0.0035
ASN_TRP	0.0006	-0.0020	0.0006	-0.0041	-0.0007
ASN_TYR	-0.0020	0.0084	-0.0025	0.0040	0.0208
ASN_VAL	0.0007	-0.0034	-0.0066	-0.0037	0.0088
ASP_ASP	0.0007	0.0048	0.0059	0.0029	-0.0155
ASP_CYS	-0.0024	-0.0079	-0.0001	-0.0010	0.0125
ASP_GLN	0.0026	0.0105	0.0026	-0.0021	-0.0073
ASP_GLU	-0.0018	-0.0028	0.0093	0.0022	0.0203
ASP_GLY	-0.0002	-0.0076	-0.0140	-0.0027	-0.0063
ASP_HID	0.0001	-0.0040	-0.0131	0.0012	-0.0104
ASP_ILE	-0.0002	0.0035	0.0131	-0.0019	-0.0303
ASP_LEU	-0.0015	0.0021	0.0019	0.0023	0.0051
ASP_LYS	-0.0008	0.0002	0.0077	-0.0063	0.0190
ASP_MET	-0.0023	0.0027	0.0183	0.0001	-0.0111
ASP_PHE	0.0025	0.0057	0.0045	-0.0006	-0.0051
ASP_PRO	0.0045	-0.0025	-0.0099	-0.0042	0.0183
ASP_SER	-0.0004	0.0016	-0.0166	-0.0013	0.0072
ASP_THR	0.0016	-0.0012	0.0182	-0.0011	-0.0098
ASP_TRP	0.0001	0.0033	0.0102	-0.0023	-0.0292
ASP_TYR	-0.0001	-0.0094	-0.0111	0.0015	-0.0126
ASP_VAL	-0.0008	0.0120	0.0116	0.0037	-0.0150
CYS_CYS	0.0040	0.0038	-0.0057	-0.0014	0.0022
CYS_GLN	0.0008	0.0014	-0.0229	0.0053	0.0110
CYS_GLU	0.0003	-0.0043	-0.0004	0.0010	-0.0078
CYS_GLY	-0.0011	0.0009	-0.0024	0.0037	0.0123
CYS_HID	-0.0002	0.0014	0.0135	0.0002	-0.0130
CYS_ILE	-0.0003	0.0050	0.0073	0.0004	-0.0115
CYS_LEU	0.0007	0.0002	0.0031	-0.0015	0.0109
CYS_LYS	-0.0007	-0.0032	-0.0108	-0.0008	0.0053
CYS_MET	0.0016	0.0046	-0.0031	0.0002	-0.0044
CYS_PHE	0.0000	0.0037	-0.0067	-0.0022	-0.0026
CYS_PRO	0.0010	0.0059	0.0073	0.0027	-0.0071
CYS_SER	0.0007	0.0003	-0.0035	0.0011	0.0037
CYS_THR	0.0007	-0.0111	-0.0090	-0.0015	0.0041
CYS_TRP	0.0011	-0.0060	-0.0053	-0.0016	-0.0122
CYS_TYR	-0.0016	0.0032	-0.0121	-0.0007	0.0104
CYS_VAL	-0.0001	-0.0023	-0.0036	-0.0020	-0.0046
GLN_GLN	0.0015	0.0016	-0.0042	-0.0001	-0.0157
GLN_GLU	-0.0027	0.0048	-0.0209	0.0033	0.0149
GLN_GLY	0.0041	0.0027	0.0066	-0.0016	0.0014
GLN_HID	-0.0012	-0.0020	0.0000	0.0022	0.0079
GLN_ILE	-0.0007	-0.0022	-0.0097	0.0032	0.0137
GLN_LEU	-0.0002	-0.0016	-0.0025	0.0010	-0.0044
GLN_LYS	0.0026	0.0006	-0.0042	-0.0031	0.0083
GLN_MET	0.0027	-0.0007	-0.0291	0.0012	-0.0003
GLN_PHE	0.0006	-0.0007	-0.0249	0.0018	-0.0024
GLN_PRO	-0.0050	0.0004	-0.0018	0.0009	-0.0183

Table 5.11. (cont'd)

GLN_SER	-0.0014	-0.0008	0.0034	-0.0009	0.0024
GLN_THR	-0.0014	-0.0022	0.0049	0.0003	0.0038
GLN_TRP	-0.0008	-0.0027	-0.0303	0.0039	0.0055
GLN_TYR	-0.0015	-0.0054	-0.0157	0.0015	0.0139
GLN_VAL	0.0007	-0.0004	-0.0247	-0.0019	-0.0043
GLU_GLU	0.0001	-0.0029	0.0084	0.0007	-0.0107
GLU_GLY	0.0006	-0.0010	0.0119	0.0009	-0.0109
GLU_HID	0.0002	0.0000	-0.0026	0.0025	0.0008
GLU_ILE	0.0005	0.0031	-0.0148	-0.0002	0.0141
GLU_LEU	-0.0015	-0.0006	0.0030	0.0033	0.0057
GLU_LYS	-0.0048	0.0025	0.0071	-0.0007	0.0143
GLU_MET	0.0004	-0.0007	-0.0039	-0.0006	0.0178
GLU_PHE	0.0001	-0.0142	0.0160	0.0019	0.0057
GLU_PRO	0.0012	0.0061	0.0210	0.0042	-0.0238
GLU_SER	-0.0020	-0.0067	-0.0014	0.0004	0.0120
GLU_THR	0.0001	0.0049	0.0074	-0.0022	-0.0039
GLU_TRP	0.0038	0.0008	-0.0164	-0.0002	-0.0131
GLU_TYR	-0.0022	-0.0099	0.0062	-0.0003	-0.0134
GLU_VAL	-0.0025	-0.0022	-0.0014	0.0010	-0.0127
GLY_GLY	0.0000	0.0014	-0.0024	-0.0007	0.0063
GLY_HID	-0.0004	-0.0009	-0.0102	0.0007	-0.0027
GLY_ILE	0.0003	-0.0020	0.0021	-0.0026	0.0000
GLY_LEU	-0.0001	-0.0020	-0.0006	0.0012	0.0032
GLY_LYS	-0.0006	-0.0029	-0.0095	0.0013	0.0065
GLY_MET	-0.0002	-0.0038	-0.0064	-0.0003	-0.0068
GLY_PHE	0.0004	0.0083	0.0054	-0.0001	0.0029
GLY_PRO	-0.0018	0.0040	0.0193	0.0019	-0.0134
GLY_SER	0.0003	-0.0015	-0.0046	-0.0005	-0.0065
GLY_THR	0.0011	0.0006	0.0171	-0.0013	0.0010
GLY_TRP	0.0000	0.0143	0.0068	0.0006	0.0030
GLY_TYR	0.0001	-0.0058	0.0048	0.0005	-0.0059
GLY_VAL	0.0008	-0.0088	0.0112	-0.0021	-0.0039
HID_HID	-0.0003	0.0008	0.0116	0.0041	-0.0011
HID_ILE	-0.0024	-0.0030	0.0047	0.0045	0.0002
HID_LEU	0.0009	-0.0062	-0.0140	-0.0018	0.0100
HID_LYS	-0.0013	0.0027	0.0016	-0.0007	0.0004
HID_MET	-0.0001	0.0089	0.0202	0.0027	-0.0076
HID_PHE	0.0000	0.0043	-0.0047	0.0011	-0.0012
HID_PRO	-0.0098	-0.0033	-0.0037	0.0016	0.0011
HID_SER	-0.0016	-0.0004	0.0028	0.0016	-0.0067
HID_THR	0.0006	-0.0026	0.0030	-0.0007	0.0063
HID_TRP	-0.0010	0.0023	0.0042	-0.0008	-0.0145
HID_TYR	-0.0011	0.0032	0.0024	-0.0012	0.0056
HID_VAL	0.0004	-0.0055	-0.0095	-0.0032	0.0055
ILE_ILE	-0.0029	0.0056	0.0250	-0.0003	-0.0118
ILE_LEU	0.0004	0.0016	0.0056	0.0009	0.0056
ILE_LYS	-0.0002	-0.0014	-0.0091	-0.0022	-0.0002
ILE_MET	0.0012	0.0056	0.0013	-0.0005	-0.0022
ILE_PHE	0.0013	-0.0008	-0.0084	0.0057	0.0038
ILE_PRO	-0.0015	0.0013	-0.0041	-0.0004	0.0079
ILE_SER	0.0037	-0.0067	-0.0240	0.0010	-0.0038

Table 5.11. (cont'd)

ILE_THR	0.0003	0.0012	0.0059	-0.0003	-0.0066
ILE_TRP	-0.0031	-0.0001	0.0051	-0.0015	-0.0231
ILE_TYR	0.0025	-0.0022	-0.0073	0.0010	-0.0022
ILE_VAL	-0.0023	-0.0038	0.0217	-0.0029	-0.0069
LEU_LEU	-0.0007	0.0041	0.0120	0.0014	0.0010
LEU_LYS	0.0010	-0.0010	0.0040	-0.0014	0.0028
LEU_MET	0.0000	-0.0032	-0.0083	0.0029	-0.0082
LEU_PHE	0.0027	-0.0087	-0.0020	0.0002	-0.0027
LEU_PRO	-0.0024	0.0000	-0.0008	0.0034	0.0010
LEU_SER	-0.0006	-0.0015	0.0084	-0.0020	-0.0130
LEU_THR	-0.0005	-0.0015	0.0042	0.0001	0.0065
LEU_TRP	-0.0005	-0.0051	-0.0055	0.0045	-0.0003
LEU_TYR	-0.0012	0.0037	0.0049	-0.0012	0.0351
LEU_VAL	0.0035	-0.0017	-0.0023	-0.0003	0.0016
LYS_LYS	0.0022	0.0027	-0.0054	-0.0024	0.0153
LYS_MET	-0.0011	0.0009	-0.0030	0.0018	-0.0077
LYS_PHE	-0.0014	-0.0027	-0.0041	0.0019	-0.0009
LYS_PRO	-0.0004	-0.0041	-0.0082	-0.0006	0.0132
LYS_SER	0.0024	0.0025	-0.0148	0.0001	0.0033
LYS_THR	0.0016	0.0095	-0.0204	-0.0011	-0.0290
LYS_TRP	-0.0041	0.0049	0.0179	0.0014	-0.0014
LYS_TYR	-0.0004	-0.0051	-0.0064	0.0002	-0.0064
LYS_VAL	0.0022	0.0029	0.0010	-0.0031	-0.0052
MET_MET	-0.0005	-0.0031	-0.0038	0.0017	0.0097
MET_PHE	-0.0015	0.0014	-0.0121	0.0048	0.0048
MET_PRO	0.0008	-0.0011	-0.0007	-0.0038	-0.0067
MET_SER	-0.0037	-0.0026	0.0109	0.0005	-0.0001
MET_THR	0.0007	0.0030	0.0011	0.0008	-0.0041
MET_TRP	-0.0005	0.0026	-0.0077	0.0026	-0.0002
MET_TYR	-0.0018	-0.0105	0.0044	0.0035	0.0125
MET_VAL	-0.0024	-0.0079	-0.0170	0.0005	0.0155
PHE_PHE	0.0021	-0.0004	0.0019	0.0021	0.0097
PHE_PRO	-0.0018	-0.0001	-0.0050	0.0002	0.0082
PHE_SER	0.0004	0.0036	-0.0026	-0.0008	0.0050
PHE_THR	0.0047	0.0015	-0.0050	-0.0027	-0.0071
PHE_TRP	-0.0028	0.0035	-0.0094	0.0008	-0.0032
PHE_TYR	0.0004	0.0089	-0.0047	-0.0008	0.0146
PHE_VAL	0.0006	0.0004	0.0078	0.0001	0.0143
PRO_PRO	0.0003	-0.0016	0.0071	0.0004	0.0051
PRO_SER	0.0012	-0.0014	0.0092	-0.0014	0.0054
PRO_THR	0.0011	0.0006	0.0084	-0.0015	0.0008
PRO_TRP	-0.0008	-0.0152	0.0078	0.0005	-0.0015
PRO_TYR	-0.0025	-0.0033	-0.0169	0.0005	0.0036
PRO_VAL	0.0018	-0.0030	-0.0137	-0.0009	0.0070
SER_SER	-0.0012	0.0008	0.0054	0.0086	0.0201
SER_THR	-0.0004	-0.0009	0.0113	0.0033	-0.0133
SER_TRP	-0.0002	0.0038	-0.0101	-0.0008	0.0023
SER_TYR	0.0021	0.0093	0.0055	0.0000	0.0025
SER_VAL	-0.0036	0.0050	0.0068	0.0006	-0.0180
THR_THR	-0.0007	-0.0085	-0.0109	0.0006	0.0150
THR_TRP	-0.0007	0.0031	0.0185	-0.0013	0.0014

Table 5.11. (cont'd)

THR_TYR	-0.0002	0.0007	0.0012	0.0021	-0.0138
THR_VAL	0.0019	0.0019	-0.0114	0.0019	0.0117
TRP_TRP	0.0021	-0.0088	-0.0049	0.0031	-0.0113
TRP_TYR	-0.0019	-0.0047	0.0206	-0.0002	-0.0114
TRP_VAL	0.0012	-0.0035	-0.0008	-0.0020	0.0058
TYR_TYR	0.0002	0.0008	-0.0081	-0.0020	0.0137
TYR_VAL	0.0020	-0.0070	-0.0073	-0.0035	0.0112
VAL_VAL	0.0000	0.0046	-0.0015	0.0025	0.0107
Maximum	0.0047	0.0143	0.0807	0.0086	0.0485
Minimum	-0.0098	-0.0152	-0.0357	-0.0063	-0.0519

Table 5.12. Torsion and nonbond energies calculated by an encoded program with ff14SB parameters for double amino acid test set.

	Energy difference / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA_ALA	6.5280	2.3126	117.6865	-1.4025	-96.1481
ALA_ARG	7.0166	3.6239	-155.1195	-2.6920	57.7425
ALA_ASN	17.1385	3.0982	49.9245	-1.8712	-111.5289
ALA_ASP	12.4042	3.1055	137.8428	-1.5128	-122.4079
ALA_CYS	8.5105	2.5530	115.5498	-1.7194	-92.5868
ALA_GLN	15.6410	3.3854	52.9007	-2.1378	-95.2512
ALA_GLU	12.3539	3.3747	131.5935	-2.1234	-104.8375
ALA_GLY	4.0157	2.1425	123.3087	-1.0269	-106.0016
ALA_HID	9.9732	2.4055	106.3553	-2.7117	-95.0531
ALA_ILE	11.0335	4.3097	118.5912	-1.9128	-102.1124
ALA_LEU	9.7861	4.0513	97.5048	-2.0554	-98.1436
ALA_LYS	7.4148	3.3108	161.3028	-2.3049	-90.6006
ALA_MET	9.7597	2.6423	122.2876	-2.1289	-101.7359
ALA_PHE	7.5967	6.3179	121.1512	-3.0358	-102.6852
ALA_PRO	14.0934	2.9264	112.2445	-1.5247	-86.3372
ALA_SER	8.7284	2.7859	108.3262	-1.6823	-95.0701
ALA_THR	13.1230	3.5712	87.2247	-2.1949	-95.7348
ALA_TRP	7.8911	5.5500	140.9540	-4.0232	-114.0306
ALA_TYR	7.8507	6.2878	109.4003	-3.1586	-108.6624
ALA_VAL	8.9258	4.0006	95.4875	-1.9366	-94.7429
ARG_ARG	7.9093	4.9418	-464.2798	-4.3308	251.9799
ARG_ASN	18.0246	4.4111	-271.9103	-3.4613	64.5626
ARG_ASP	13.2676	4.4496	-186.8101	-3.0997	19.1102
ARG_CYS	9.3903	3.8928	-205.3387	-3.3387	82.6308
ARG_GLN	16.5225	4.7283	-266.3321	-3.7592	78.5311
ARG_GLU	13.2131	4.7245	-191.7080	-3.7306	39.4793
ARG_GLY	4.8914	3.4126	-194.9461	-2.5079	66.8828
ARG_HID	10.8838	3.7354	-213.8339	-4.3277	81.2921
ARG_ILE	11.8959	5.6543	-197.8616	-3.5909	67.7083
ARG_LEU	10.6662	5.3722	-221.7754	-3.6701	76.2546
ARG_LYS	8.3084	4.6439	-148.5852	-3.9435	104.1729
ARG_MET	10.6426	3.9693	-196.0188	-3.7440	71.5125
ARG_PHE	8.4738	7.6443	-198.3676	-4.6517	70.9449
ARG_PRO	15.2728	4.1005	-191.1913	-3.6667	73.5006
ARG_SER	9.6089	4.1231	-207.8877	-3.2955	74.3369
ARG_THR	14.1231	4.8720	-226.8575	-4.0157	66.6243
ARG_TRP	8.7656	6.8922	-177.4498	-5.6548	57.8402
ARG_TYR	8.7314	7.6511	-210.0522	-4.7774	64.8574
ARG_VAL	9.7880	5.3323	-218.1662	-3.6080	72.3257
ASN_ASN	27.9845	3.7466	-20.8185	-2.8517	-126.6835
ASN_ASP	23.2611	3.7717	66.9269	-2.5927	-133.8336
ASN_CYS	19.3692	3.1944	44.8261	-2.7487	-107.1895
ASN_GLN	26.4816	4.0282	-17.9992	-3.1324	-109.5975
ASN_GLU	23.1964	4.0255	60.5317	-3.2025	-116.2429
ASN_GLY	14.9039	2.8194	52.7182	-1.9765	-120.5007
ASN_HID	20.7802	3.0410	35.3693	-3.6568	-110.5193
ASN_ILE	21.8810	4.9814	47.9360	-3.0682	-116.3531
ASN_LEU	20.6334	4.7090	26.7883	-3.0481	-112.9382
ASN_LYS	18.2466	3.9508	90.7423	-3.2222	-107.8708

Table 5.12. (cont'd)

ASN_MET	20.6045	3.2704	51.3683	-3.1206	-116.2559
ASN_PHE	18.4480	6.9523	50.3266	-4.0792	-117.1698
ASN_PRO	26.9232	3.4028	42.5466	-3.5477	-104.6211
ASN_SER	19.5877	3.4210	37.4947	-2.7528	-109.3346
ASN_THR	24.6294	4.0473	15.6958	-3.2017	-111.0090
ASN_TRP	18.7492	6.1818	70.2015	-5.0988	-128.4517
ASN_TYR	18.7043	6.9387	38.6138	-4.2082	-123.2203
ASN_VAL	19.7815	4.6713	24.8437	-3.0867	-108.9964
ASP_ASP	18.9973	3.6512	141.7389	-1.7719	-83.9246
ASP_CYS	15.0767	3.0595	120.7193	-1.9233	-103.0556
ASP_GLN	22.1663	3.9078	57.1436	-2.2825	-104.4918
ASP_GLU	18.9232	3.8978	134.6703	-2.3619	-70.8014
ASP_GLY	10.6740	2.7197	127.8023	-1.2125	-113.8296
ASP_HID	16.4189	2.9559	110.8170	-2.7748	-108.9906
ASP_ILE	17.6283	4.8410	122.4760	-2.2940	-107.8805
ASP_LEU	16.3337	4.6208	102.5449	-2.2155	-108.7822
ASP_LYS	13.9114	3.8506	165.4792	-2.3452	-136.9553
ASP_MET	16.2893	3.1785	126.2913	-2.2814	-111.5839
ASP_PHE	14.1524	6.8256	125.6892	-3.2719	-111.8069
ASP_PRO	22.4531	3.3107	115.1372	-2.5028	-81.7720
ASP_SER	15.2961	3.2847	111.4700	-1.9350	-101.6278
ASP_THR	20.6377	3.7235	87.0424	-2.1337	-99.0827
ASP_TRP	14.4596	6.0729	145.2762	-4.2969	-121.8024
ASP_TYR	14.4070	6.8155	113.8923	-3.4045	-117.7040
ASP_VAL	15.5271	4.5476	98.3364	-2.3175	-99.4558
CYS_CYS	10.9090	2.3820	104.4951	-1.9397	-79.8821
CYS_GLN	18.0289	3.2132	41.7873	-2.3490	-82.6290
CYS_GLU	14.7703	3.1908	119.4827	-2.3647	-89.3800
CYS_GLY	6.4282	2.0082	112.2790	-1.1685	-93.3558
CYS_HID	12.3470	2.2323	95.2076	-2.9147	-82.7586
CYS_ILE	13.4405	4.1231	107.5202	-2.1709	-89.1540
CYS_LEU	12.1801	3.8890	86.4928	-2.2614	-85.5853
CYS_LYS	9.7930	3.1453	151.0383	-2.5100	-80.0173
CYS_MET	12.1512	2.4645	111.1543	-2.3366	-89.0926
CYS_PHE	9.9940	6.1437	110.0443	-3.2649	-89.9671
CYS_PRO	16.2765	2.6795	102.4443	-2.4433	-75.3615
CYS_SER	11.1365	2.6122	97.1852	-1.9034	-82.0838
CYS_THR	15.4396	3.3550	75.9955	-2.5582	-81.5455
CYS_TRP	10.2893	5.3699	129.8747	-4.2597	-101.2032
CYS_TYR	10.2463	6.1186	98.2846	-3.3865	-95.9285
CYS_VAL	11.3392	3.8139	84.3884	-2.1964	-81.7533
GLN_GLN	25.1965	4.3181	-30.5644	-3.1758	-76.8539
GLN_GLU	21.8911	4.3081	46.4162	-3.1491	-86.8118
GLN_GLY	13.5618	3.0293	39.8007	-1.9474	-87.5874
GLN_HID	19.5461	3.3314	22.7876	-3.7452	-76.2270
GLN_ILE	20.5590	5.2572	35.2401	-2.9972	-84.4041
GLN_LEU	19.3402	4.9763	14.1147	-3.0926	-79.9226
GLN_LYS	16.9780	4.2589	79.4603	-3.3551	-72.6103
GLN_MET	19.3138	3.5751	38.9063	-3.1609	-83.4619
GLN_PHE	17.1457	7.2553	37.6332	-4.0693	-84.4708
GLN_PRO	23.9525	3.7069	32.2013	-3.0117	-72.0316

Table 5.12. (cont'd)

GLN_SER	18.2711	3.7387	24.9622	-2.7113	-77.2972
GLN_THR	22.7446	4.4935	3.8166	-3.3947	-79.2694
GLN_TRP	17.4348	6.5009	57.5191	-5.0727	-95.9935
GLN_TYR	17.3970	7.2267	25.8928	-4.1949	-90.4446
GLN_VAL	18.4576	4.9438	12.3060	-3.0158	-77.1930
GLU_GLU	18.6469	4.1410	119.8831	-3.1349	-49.2654
GLU_GLY	10.2981	2.9417	113.2641	-1.9266	-84.1324
GLU_HID	16.2416	3.1823	96.7034	-3.7083	-75.6152
GLU_ILE	17.3143	5.0948	108.1624	-2.9681	-79.1952
GLU_LEU	16.1185	4.7117	88.2765	-2.8534	-77.9850
GLU_LYS	13.6872	4.1099	151.8997	-3.3098	-99.1985
GLU_MET	16.0387	3.4182	112.2653	-3.1265	-80.8048
GLU_PHE	13.8727	7.0954	111.4195	-4.0475	-81.4035
GLU_PRO	20.5202	3.4694	104.0916	-2.6343	-62.9287
GLU_SER	15.0082	3.5810	97.4747	-2.6851	-72.0477
GLU_THR	19.4009	4.3377	75.4383	-3.3655	-67.5472
GLU_TRP	14.1706	6.3260	131.0136	-5.0503	-91.7576
GLU_TYR	14.1272	7.0818	99.6316	-4.1777	-87.1942
GLU_VAL	15.2073	4.7776	84.1698	-2.9840	-71.0269
GLY_GLY	4.0568	1.6628	102.7650	-0.6670	-92.1571
GLY_HID	10.0112	1.9220	85.5779	-2.2987	-80.9198
GLY_ILE	11.0758	3.8174	98.1305	-1.4620	-88.3679
GLY_LEU	9.8318	3.5648	77.0117	-1.6355	-84.3610
GLY_LYS	7.4595	2.8325	143.0816	-1.8874	-78.8953
GLY_MET	9.8050	2.1521	101.6555	-1.7154	-87.7890
GLY_PHE	7.6425	5.8266	100.4681	-2.6193	-88.6530
GLY_PRO	14.1084	2.5868	96.6969	-1.2354	-77.0594
GLY_SER	8.7689	2.3038	87.7694	-1.2733	-81.1589
GLY_THR	13.1815	3.0914	66.6990	-1.7132	-81.9155
GLY_TRP	7.9347	5.0726	120.3785	-3.5978	-100.1006
GLY_TYR	7.8962	5.8082	88.7199	-2.7418	-94.6132
GLY_VAL	8.9699	3.5072	75.1814	-1.4938	-81.1571
HID_HID	13.8723	2.3360	81.3216	-5.2920	-78.6067
HID_ILE	14.8938	4.2318	93.7811	-4.6085	-85.2229
HID_LEU	13.6837	3.9912	72.6270	-4.6440	-81.4766
HID_LYS	11.3069	3.2492	137.4771	-4.9035	-74.4730
HID_MET	13.6595	2.5665	97.3235	-4.7074	-84.7784
HID_PHE	11.4934	6.2439	96.2257	-5.6568	-85.7839
HID_PRO	18.4480	2.8698	90.4206	-4.2054	-81.3746
HID_SER	12.6064	2.7177	83.4012	-4.2505	-78.3310
HID_THR	16.8116	3.7174	64.8478	-4.3805	-89.1514
HID_TRP	11.7783	5.4720	116.0573	-6.6665	-97.2551
HID_TYR	11.7490	6.2120	84.4592	-5.7844	-91.8738
HID_VAL	12.7866	3.9258	70.6870	-4.6053	-77.8978
ILE_ILE	15.8893	5.9655	84.2998	-2.6350	-77.1191
ILE_LEU	14.6457	5.7084	63.2005	-2.7063	-73.1160
ILE_LYS	12.2767	4.9821	130.0936	-2.9701	-68.6753
ILE_MET	14.6201	4.3029	87.8457	-2.7763	-76.5255
ILE_PHE	12.4555	7.9848	86.5235	-3.6973	-77.3056
ILE_PRO	19.2547	4.4081	84.5870	-2.5322	-67.5111
ILE_SER	13.5915	4.4535	73.8938	-2.3300	-70.0161

Table 5.12. (cont'd)

ILE_THR	17.9678	5.2311	52.7670	-3.0057	-70.5951
ILE_TRP	12.7474	7.2226	106.4756	-4.7046	-88.7978
ILE_TYR	12.7106	7.9674	74.7862	-3.8266	-83.2669
ILE_VAL	13.7835	5.6578	61.4502	-2.6499	-70.0550
LEU_LEU	13.5945	5.6899	63.0049	-3.3027	-93.3937
LEU_LYS	11.2464	4.9581	128.3883	-3.5745	-87.2265
LEU_MET	13.5768	4.2700	87.6483	-3.3772	-96.8095
LEU_PHE	11.4064	7.9611	86.3356	-4.2916	-97.6166
LEU_PRO	18.5714	4.4984	80.7557	-3.1356	-83.7364
LEU_SER	12.5473	4.4305	73.6836	-2.9347	-90.3296
LEU_THR	16.9417	5.1981	52.6275	-3.6534	-90.9104
LEU_TRP	11.6967	7.1828	106.3209	-5.3009	-109.1064
LEU_TYR	11.6591	7.9309	74.5793	-4.4222	-103.5734
LEU_VAL	12.7080	5.6387	61.2403	-3.2638	-90.2831
LYS_LYS	8.6939	4.2781	171.3038	-3.5314	-45.4134
LYS_MET	11.0330	3.6081	124.4217	-3.3356	-78.2226
LYS_PHE	8.8668	7.2951	122.1350	-4.2423	-78.8246
LYS_PRO	15.6520	3.7526	128.2222	-3.2329	-76.2325
LYS_SER	10.0010	3.7651	112.4688	-2.8809	-75.3101
LYS_THR	14.4987	4.5370	93.5142	-3.5840	-82.7968
LYS_TRP	9.1591	6.5259	143.0397	-5.2445	-91.8499
LYS_TYR	9.1216	7.2670	110.4253	-4.3677	-84.9175
LYS_VAL	10.1856	4.9676	102.1771	-3.1953	-77.2144
MET_MET	13.4350	2.8200	103.0089	-3.2060	-86.3285
MET_PHE	11.2652	6.4971	101.7498	-4.1168	-87.2333
MET_PRO	18.0173	2.9833	97.5342	-3.1331	-75.5615
MET_SER	12.3963	2.9839	89.0948	-2.7589	-79.9252
MET_THR	16.8934	3.7504	67.9690	-3.5088	-81.1402
MET_TRP	11.5585	5.7433	121.6639	-5.1254	-98.6982
MET_TYR	11.5213	6.4843	90.0092	-4.2453	-93.1956
MET_VAL	12.5811	4.1817	76.5158	-3.0688	-79.8849
PHE_PHE	8.8690	10.0964	107.0504	-6.1741	-92.1391
PHE_PRO	15.4840	6.5747	101.4620	-5.0988	-79.9522
PHE_SER	10.0096	6.5360	94.1256	-4.7424	-84.2691
PHE_THR	14.2474	7.2030	72.8452	-5.4750	-83.0128
PHE_TRP	9.1665	9.3161	126.8904	-7.1916	-103.3919
PHE_TYR	9.1257	10.0721	95.2836	-6.3051	-98.1342
PHE_VAL	10.2077	7.7409	81.4624	-5.1171	-83.9398
PRO_PRO	17.5120	2.8868	72.8938	-2.2480	-57.3659
PRO_SER	12.2004	2.6706	63.0541	-2.2137	-60.5165
PRO_THR	16.6168	3.4533	41.8183	-2.7470	-61.0888
PRO_TRP	11.3682	5.4203	95.7614	-4.5660	-79.3417
PRO_TYR	11.3270	6.1699	64.1022	-3.7006	-73.9033
PRO_VAL	12.3954	3.8743	50.4475	-2.4789	-60.3810
SER_SER	11.2371	2.7488	61.0942	-1.7591	-55.9101
SER_THR	15.5318	3.5052	39.9507	-2.3943	-55.1979
SER_TRP	10.3993	5.5132	93.7906	-4.1119	-74.9205
SER_TYR	10.3531	6.2666	62.1423	-3.2397	-69.6242
SER_VAL	11.4523	3.9602	48.5044	-2.0396	-55.7095
THR_THR	19.2257	4.6269	1.2414	-2.7763	-34.8942
THR_TRP	13.9779	6.5994	54.8633	-4.5659	-52.6021

Table 5.12. (cont'd)

THR_TYR	13.9386	7.3544	23.1877	-3.6970	-47.1025
THR_VAL	15.0228	5.0458	9.7650	-2.4901	-33.8353
TRP_TRP	9.3833	8.5720	140.8008	-8.5027	-109.7283
TRP_TYR	9.3402	9.3300	109.1063	-7.6203	-104.3638
TRP_VAL	11.7296	6.6231	92.8557	-6.1057	-88.1176
TYR_TYR	9.3749	10.0634	83.1288	-6.6073	-102.9884
TYR_VAL	10.4433	7.7158	69.3328	-5.4419	-88.8127
VAL_VAL	11.6396	5.3037	21.1357	-2.5695	-45.7340

Table 5.13. Torsion and nonbond energies calculated by Amber software with ff14SB parameters for double amino acid test set.

	Energy difference / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA_ALA	6.5282	2.3091	117.6841	-1.4026	-96.1489
ALA_ARG	7.0170	3.6214	-155.1509	-2.6933	57.7762
ALA_ASN	17.1392	3.0909	49.9035	-1.8698	-111.5165
ALA_ASP	12.4040	3.1048	137.8546	-1.5127	-122.4424
ALA_CYS	8.5098	2.5578	115.5685	-1.7206	-92.5853
ALA_GLN	15.6404	3.3884	52.9085	-2.1389	-95.2611
ALA_GLU	12.3527	3.3713	131.5812	-2.1224	-104.8197
ALA_GLY	4.0146	2.1422	123.3091	-1.0269	-106.0097
ALA_HID	9.9729	2.4023	106.3461	-2.7109	-95.0422
ALA_ILE	11.0312	4.3062	118.5851	-1.9103	-102.1069
ALA_LEU	9.7864	4.0532	97.5104	-2.0542	-98.1581
ALA_LYS	7.4157	3.3160	161.3121	-2.3042	-90.5891
ALA_MET	9.7609	2.6382	122.2837	-2.1295	-101.7366
ALA_PHE	7.5973	6.3132	121.1507	-3.0366	-102.6976
ALA_PRO	14.0927	2.9197	112.2296	-1.5277	-86.3184
ALA_SER	8.7280	2.7884	108.3392	-1.6829	-95.0772
ALA_THR	13.1223	3.5711	87.2368	-2.1952	-95.7339
ALA_TRP	7.8912	5.5500	140.9576	-4.0233	-114.0243
ALA_TYR	7.8514	6.2958	109.4072	-3.1600	-108.6725
ALA_VAL	8.9269	3.9916	95.4977	-1.9359	-94.7606
ARG_ARG	7.9099	4.9417	-464.2442	-4.3340	251.9600
ARG_ASN	18.0242	4.4043	-271.9286	-3.4648	64.5757
ARG_ASP	13.2674	4.4491	-186.8442	-3.1060	19.1445
ARG_CYS	9.3891	3.8967	-205.3971	-3.3413	82.6710
ARG_GLN	16.5234	4.7258	-266.3313	-3.7600	78.5323
ARG_GLU	13.2153	4.7238	-191.7468	-3.7332	39.4827
ARG_GLY	4.8920	3.4106	-194.9587	-2.5090	66.8844
ARG_HID	10.8851	3.7338	-213.8494	-4.3274	81.2735
ARG_ILE	11.8945	5.6490	-197.8503	-3.5930	67.7062
ARG_LEU	10.6667	5.3734	-221.8038	-3.6715	76.2732
ARG_LYS	8.3086	4.6396	-148.6216	-3.9451	104.1923
ARG_MET	10.6430	3.9709	-195.9901	-3.7467	71.5163
ARG_PHE	8.4747	7.6495	-198.3387	-4.6529	70.9472
ARG_PRO	15.2757	4.0963	-191.2041	-3.6702	73.5045
ARG_SER	9.6097	4.1278	-207.9238	-3.2953	74.3420
ARG_THR	14.1231	4.8727	-226.8921	-4.0152	66.6317
ARG_TRP	8.7665	6.8905	-177.4064	-5.6572	57.8598
ARG_TYR	8.7297	7.6336	-210.0276	-4.7785	64.8528
ARG_VAL	9.7899	5.3299	-218.1479	-3.6102	72.3134
ASN_ASN	27.9848	3.7478	-20.8155	-2.8476	-126.6976
ASN_ASP	23.2596	3.7634	66.9171	-2.5902	-133.8253
ASN_CYS	19.3676	3.1900	44.8419	-2.7493	-107.1840
ASN_GLN	26.4813	4.0234	-17.9978	-3.1309	-109.6119
ASN_GLU	23.1974	4.0234	60.5460	-3.1982	-116.2772
ASN_GLY	14.9037	2.8193	52.7154	-1.9774	-120.5051
ASN_HID	20.7816	3.0441	35.3851	-3.6560	-110.5166
ASN_ILE	21.8834	4.9776	47.9255	-3.0698	-116.3440
ASN_LEU	20.6328	4.7050	26.7914	-3.0464	-112.9472
ASN_LYS	18.2463	3.9543	90.7562	-3.2184	-107.8911

Table 5.13. (cont'd)

ASN_MET	20.6039	3.2760	51.3890	-3.1217	-116.2577
ASN_PHE	18.4486	6.9534	50.3511	-4.0785	-117.1686
ASN_PRO	26.9224	3.4016	42.5755	-3.5515	-104.6450
ASN_SER	19.5868	3.4227	37.4880	-2.7506	-109.3075
ASN_THR	24.6287	4.0420	15.7161	-3.1990	-111.0184
ASN_TRP	18.7472	6.1894	70.2149	-5.1032	-128.4556
ASN_TYR	18.7036	6.9341	38.5993	-4.2090	-123.2130
ASN_VAL	19.7796	4.6664	24.8189	-3.0872	-108.9721
ASP_ASP	18.9974	3.6566	141.7366	-1.7714	-83.9031
ASP_CYS	15.0738	3.0641	120.7395	-1.9229	-103.0825
ASP_GLN	22.1672	3.9166	57.1639	-2.2846	-104.4838
ASP_GLU	18.9239	3.8998	134.6789	-2.3561	-70.8137
ASP_GLY	10.6717	2.7244	127.7975	-1.2064	-113.8517
ASP_HID	16.4164	2.9555	110.8062	-2.7735	-108.9838
ASP_ILE	17.6335	4.8453	122.4809	-2.3007	-107.8662
ASP_LEU	16.3341	4.6208	102.5543	-2.2113	-108.7919
ASP_LYS	13.9133	3.8490	165.4784	-2.3478	-136.9650
ASP_MET	16.2923	3.1735	126.2859	-2.2781	-111.5886
ASP_PHE	14.1537	6.8407	125.6969	-3.2740	-111.7951
ASP_PRO	22.4506	3.3064	115.1461	-2.5038	-81.7805
ASP_SER	15.2951	3.2852	111.4828	-1.9388	-101.6421
ASP_THR	20.6374	3.7314	87.0484	-2.1300	-99.0745
ASP_TRP	14.4583	6.0683	145.2645	-4.3018	-121.7983
ASP_TYR	14.4095	6.8174	113.8966	-3.4065	-117.7014
ASP_VAL	15.5282	4.5469	98.3435	-2.3158	-99.4606
CYS_CYS	10.9100	2.3817	104.5054	-1.9379	-79.8937
CYS_GLN	18.0295	3.2153	41.7720	-2.3502	-82.6228
CYS_GLU	14.7725	3.1898	119.4726	-2.3617	-89.3755
CYS_GLY	6.4300	2.0053	112.2827	-1.1687	-93.3515
CYS_HID	12.3455	2.2341	95.1971	-2.9113	-82.7579
CYS_ILE	13.4402	4.1245	107.5194	-2.1703	-89.1386
CYS_LEU	12.1810	3.8890	86.4840	-2.2630	-85.5892
CYS_LYS	9.7949	3.1438	151.0466	-2.5078	-80.0327
CYS_MET	12.1524	2.4671	111.1535	-2.3387	-89.1061
CYS_PHE	9.9926	6.1396	110.0459	-3.2636	-89.9577
CYS_PRO	16.2760	2.6776	102.4500	-2.4416	-75.3708
CYS_SER	11.1345	2.6103	97.1729	-1.9056	-82.0977
CYS_THR	15.4373	3.3564	75.9967	-2.5560	-81.5378
CYS_TRP	10.2921	5.3722	129.8671	-4.2635	-101.2073
CYS_TYR	10.2482	6.1211	98.2938	-3.3890	-95.9400
CYS_VAL	11.3381	3.8127	84.3965	-2.1921	-81.7574
GLN_GLN	25.1965	4.3243	-30.5645	-3.1760	-76.8420
GLN_GLU	21.8907	4.3089	46.4066	-3.1492	-86.7995
GLN_GLY	13.5617	3.0394	39.7990	-1.9483	-87.5960
GLN_HID	19.5480	3.3332	22.8073	-3.7448	-76.2269
GLN_ILE	20.5604	5.2530	35.2455	-2.9977	-84.3988
GLN_LEU	19.3382	4.9808	14.1123	-3.0886	-79.9226
GLN_LYS	16.9792	4.2608	79.4663	-3.3553	-72.6036
GLN_MET	19.3146	3.5733	38.8806	-3.1620	-83.4509
GLN_PHE	17.1451	7.2502	37.6421	-4.0719	-84.4789
GLN_PRO	23.9502	3.7046	32.2167	-3.0092	-72.0375

Table 5.13. (cont'd)

GLN_SER	18.2733	3.7322	24.9567	-2.7114	-77.2824
GLN_THR	22.7461	4.4885	3.8052	-3.3946	-79.2634
GLN_TRP	17.4344	6.4919	57.5225	-5.0740	-95.9908
GLN_TYR	17.3992	7.2338	25.9013	-4.1971	-90.4286
GLN_VAL	18.4552	4.9370	12.2917	-3.0152	-77.1943
GLU_GLU	18.6487	4.1413	119.8938	-3.1346	-49.2682
GLU_GLY	10.2968	2.9371	113.2688	-1.9271	-84.1351
GLU_HID	16.2436	3.1839	96.6895	-3.7066	-75.6186
GLU_ILE	17.3121	5.0882	108.1523	-2.9666	-79.2074
GLU_LEU	16.1179	4.7079	88.2605	-2.8551	-77.9815
GLU_LYS	13.6894	4.1101	151.9113	-3.3092	-99.2071
GLU_MET	16.0368	3.4216	112.2724	-3.1272	-80.7798
GLU_PHE	13.8753	7.0947	111.4143	-4.0489	-81.4068
GLU_PRO	20.5160	3.4710	104.0921	-2.6301	-62.9401
GLU_SER	15.0077	3.5758	97.4635	-2.6833	-72.0663
GLU_THR	19.4001	4.3390	75.4564	-3.3651	-67.5732
GLU_TRP	14.1693	6.3326	131.0126	-5.0513	-91.7557
GLU_TYR	14.1292	7.0770	99.6367	-4.1748	-87.2148
GLU_VAL	15.2079	4.7771	84.1734	-2.9838	-71.0673
GLY_GLY	4.0571	1.6663	102.7669	-0.6665	-92.1614
GLY_HID	10.0137	1.9185	85.5641	-2.2972	-80.9031
GLY_ILE	11.0750	3.8193	98.1415	-1.4622	-88.3655
GLY_LEU	9.8317	3.5702	77.0255	-1.6373	-84.3646
GLY_LYS	7.4591	2.8330	143.0797	-1.8861	-78.8947
GLY_MET	9.8045	2.1546	101.6800	-1.7160	-87.7888
GLY_PHE	7.6422	5.8286	100.4684	-2.6192	-88.6522
GLY_PRO	14.1074	2.5911	96.6984	-1.2380	-77.0469
GLY_SER	8.7684	2.3052	87.7668	-1.2730	-81.1536
GLY_THR	13.1819	3.0889	66.6964	-1.7144	-81.9070
GLY_TRP	7.9352	5.0654	120.3817	-3.5973	-100.1118
GLY_TYR	7.8957	5.8108	88.7200	-2.7416	-94.6265
GLY_VAL	8.9702	3.5071	75.1746	-1.4934	-81.1613
HID_HID	13.8729	2.3383	81.3272	-5.2944	-78.5939
HID_ILE	14.8953	4.2373	93.7686	-4.6061	-85.2148
HID_LEU	13.6844	3.9917	72.6274	-4.6461	-81.4784
HID_LYS	11.3083	3.2481	137.4761	-4.9013	-74.4760
HID_MET	13.6590	2.5692	97.3191	-4.7082	-84.7790
HID_PHE	11.4937	6.2437	96.2157	-5.6565	-85.7706
HID_PRO	18.4450	2.8695	90.4082	-4.2075	-81.3837
HID_SER	12.6047	2.7181	83.3998	-4.2492	-78.3228
HID_THR	16.8114	3.7095	64.8340	-4.3792	-89.1386
HID_TRP	11.7805	5.4768	116.0556	-6.6675	-97.2560
HID_TYR	11.7450	6.2253	84.4679	-5.7820	-91.8666
HID_VAL	12.7876	3.9261	70.6911	-4.6109	-77.8877
ILE_ILE	15.8877	5.9732	84.3253	-2.6350	-77.1453
ILE_LEU	14.6445	5.7113	63.2066	-2.7042	-73.1167
ILE_LYS	12.2764	4.9827	130.1112	-2.9697	-68.6922
ILE_MET	14.6221	4.2991	87.8321	-2.7770	-76.5238
ILE_PHE	12.4562	7.9745	86.5202	-3.6956	-77.2997
ILE_PRO	19.2531	4.4099	84.5980	-2.5348	-67.5243
ILE_SER	13.5922	4.4533	73.8963	-2.3312	-70.0281

Table 5.13. (cont'd)

ILE_THR	17.9664	5.2206	52.7832	-3.0056	-70.5953
ILE_TRP	12.7503	7.2130	106.4955	-4.7046	-88.8111
ILE_TYR	12.7109	7.9575	74.7720	-3.8214	-83.2649
ILE_VAL	13.7825	5.6585	61.4601	-2.6494	-70.0597
LEU_LEU	13.5940	5.6858	63.0026	-3.3027	-93.3972
LEU_LYS	11.2465	4.9574	128.3958	-3.5757	-87.2341
LEU_MET	13.5761	4.2746	87.6406	-3.3781	-96.8174
LEU_PHE	11.4052	7.9490	86.3375	-4.2959	-97.6069
LEU_PRO	18.5746	4.5002	80.7675	-3.1374	-83.7435
LEU_SER	12.5444	4.4279	73.6941	-2.9369	-90.3152
LEU_THR	16.9427	5.2028	52.6326	-3.6550	-90.9020
LEU_TRP	11.6953	7.1875	106.3052	-5.3042	-109.1018
LEU_TYR	11.6601	7.9322	74.5923	-4.4227	-103.5727
LEU_VAL	12.7097	5.6310	61.2493	-3.2665	-90.3028
LYS_LYS	8.6952	4.2772	171.3113	-3.5313	-45.4004
LYS_MET	11.0333	3.6107	124.4326	-3.3344	-78.2293
LYS_PHE	8.8658	7.2889	122.1373	-4.2427	-78.8203
LYS_PRO	15.6536	3.7499	128.2239	-3.2305	-76.2389
LYS_SER	10.0019	3.7673	112.4833	-2.8832	-75.3285
LYS_THR	14.4996	4.5276	93.5190	-3.5844	-82.7711
LYS_TRP	9.1589	6.5282	143.0227	-5.2453	-91.8419
LYS_TYR	9.1210	7.2727	110.4476	-4.3681	-84.9413
LYS_VAL	10.1855	4.9699	102.1546	-3.1944	-77.2011
MET_MET	13.4340	2.8189	103.0132	-3.2066	-86.3307
MET_PHE	11.2657	6.4954	101.7523	-4.1213	-87.2293
MET_PRO	18.0155	2.9767	97.5353	-3.1302	-75.5661
MET_SER	12.3981	2.9768	89.0866	-2.7593	-79.9232
MET_THR	16.8929	3.7465	67.9793	-3.5080	-81.1601
MET_TRP	11.5571	5.7361	121.6680	-5.1263	-98.7293
MET_TYR	11.5201	6.4790	90.0090	-4.2468	-93.1910
MET_VAL	12.5809	4.1823	76.5132	-3.0690	-79.8933
PHE_PHE	8.8691	10.0936	107.0462	-6.1755	-92.1359
PHE_PRO	15.4860	6.5744	101.4619	-5.0999	-79.9568
PHE_SER	10.0113	6.5467	94.1375	-4.7430	-84.2849
PHE_THR	14.2484	7.1953	72.8453	-5.4724	-83.0104
PHE_TRP	9.1682	9.3181	126.8987	-7.1934	-103.3995
PHE_TYR	9.1249	10.0724	95.2835	-6.3031	-98.1220
PHE_VAL	10.2036	7.7438	81.4659	-5.1192	-83.9486
PRO_PRO	17.5127	2.8886	72.8967	-2.2462	-57.3684
PRO_SER	12.1990	2.6669	63.0465	-2.2128	-60.5143
PRO_THR	16.6148	3.4496	41.8326	-2.7460	-61.0971
PRO_TRP	11.3664	5.4249	95.7587	-4.5677	-79.3411
PRO_TYR	11.3261	6.1706	64.1092	-3.7004	-73.8992
PRO_VAL	12.3973	3.8713	50.4373	-2.4812	-60.3802
SER_SER	11.2379	2.7509	61.0849	-1.7581	-55.9137
SER_THR	15.5296	3.5013	39.9627	-2.3935	-55.1953
SER_TRP	10.3988	5.5147	93.8043	-4.1144	-74.9394
SER_TYR	10.3529	6.2659	62.1415	-3.2397	-69.6217
SER_VAL	11.4515	3.9557	48.5157	-2.0406	-55.7108
THR_THR	19.2258	4.6262	1.2434	-2.7759	-34.8859
THR_TRP	13.9776	6.6122	54.8779	-4.5681	-52.6080

Table 5.13. (cont'd)

THR_TYR	13.9395	7.3578	23.1922	-3.6967	-47.1002
THR_VAL	15.0208	5.0505	9.7613	-2.4891	-33.8293
TRP_TRP	9.3826	8.5740	140.7899	-8.5034	-109.7377
TRP_TYR	9.3407	9.3261	109.1256	-7.6196	-104.3716
TRP_VAL	11.7309	6.6227	92.8398	-6.1037	-88.1134
TYR_TYR	9.3730	10.0502	83.1507	-6.6067	-102.9952
TYR_VAL	10.4448	7.7262	69.3126	-5.4425	-88.8102
VAL_VAL	11.6387	5.3042	21.1595	-2.5726	-45.7387

Table 5.14. Comparisons of torsion and nonbond energies calculated by the encoded program and Amber software with ff14SB parameters for double amino acid test set.

	Energy difference / kcal/mol				
	Dihedral	1_4_VdW	1_4_EEL	VdW	EEL
ALA_ALA	-0.0002	0.0035	0.0024	0.0001	0.0008
ALA_ARG	-0.0004	0.0025	0.0314	0.0013	-0.0337
ALA_ASN	-0.0007	0.0073	0.0210	-0.0014	-0.0124
ALA_ASP	0.0002	0.0007	-0.0118	-0.0001	0.0345
ALA_CYS	0.0007	-0.0048	-0.0187	0.0012	-0.0015
ALA_GLN	0.0006	-0.0030	-0.0078	0.0011	0.0099
ALA_GLU	0.0012	0.0034	0.0123	-0.0010	-0.0178
ALA_GLY	0.0011	0.0003	-0.0004	0.0000	0.0081
ALA_HID	0.0003	0.0032	0.0092	-0.0008	-0.0109
ALA_ILE	0.0023	0.0035	0.0061	-0.0025	-0.0055
ALA_LEU	-0.0003	-0.0019	-0.0056	-0.0012	0.0145
ALA_LYS	-0.0009	-0.0052	-0.0093	-0.0007	-0.0115
ALA_MET	-0.0012	0.0041	0.0039	0.0006	0.0007
ALA_PHE	-0.0006	0.0047	0.0005	0.0008	0.0124
ALA_PRO	0.0007	0.0067	0.0149	0.0030	-0.0188
ALA_SER	0.0004	-0.0025	-0.0130	0.0006	0.0071
ALA_THR	0.0007	0.0001	-0.0121	0.0003	-0.0009
ALA_TRP	-0.0001	0.0000	-0.0036	0.0001	-0.0063
ALA_TYR	-0.0007	-0.0080	-0.0069	0.0014	0.0101
ALA_VAL	-0.0011	0.0090	-0.0102	-0.0007	0.0177
ARG_ARG	-0.0006	0.0001	-0.0356	0.0032	0.0199
ARG_ASN	0.0004	0.0068	0.0183	0.0035	-0.0131
ARG_ASP	0.0002	0.0005	0.0341	0.0063	-0.0343
ARG_CYS	0.0012	-0.0039	0.0584	0.0026	-0.0402
ARG_GLN	-0.0009	0.0025	-0.0008	0.0008	-0.0012
ARG_GLU	-0.0022	0.0007	0.0388	0.0026	-0.0034
ARG_GLY	-0.0006	0.0020	0.0126	0.0011	-0.0016
ARG_HID	-0.0013	0.0016	0.0155	-0.0003	0.0186
ARG_ILE	0.0014	0.0053	-0.0113	0.0021	0.0021
ARG_LEU	-0.0005	-0.0012	0.0284	0.0014	-0.0186
ARG_LYS	-0.0002	0.0043	0.0364	0.0016	-0.0194
ARG_MET	-0.0004	-0.0016	-0.0287	0.0027	-0.0038
ARG_PHE	-0.0009	-0.0052	-0.0289	0.0012	-0.0023
ARG_PRO	-0.0029	0.0042	0.0128	0.0035	-0.0039
ARG_SER	-0.0008	-0.0047	0.0361	-0.0002	-0.0051
ARG_THR	0.0000	-0.0007	0.0346	-0.0005	-0.0074
ARG_TRP	-0.0009	0.0017	-0.0434	0.0024	-0.0196
ARG_TYR	0.0017	0.0175	-0.0246	0.0011	0.0046
ARG_VAL	-0.0019	0.0024	-0.0183	0.0022	0.0123
ASN_ASN	-0.0003	-0.0012	-0.0030	-0.0041	0.0141
ASN_ASP	0.0015	0.0083	0.0098	-0.0025	-0.0083
ASN_CYS	0.0016	0.0044	-0.0158	0.0006	-0.0055
ASN_GLN	0.0003	0.0048	-0.0014	-0.0015	0.0144
ASN_GLU	-0.0010	0.0021	-0.0143	-0.0043	0.0343
ASN_GLY	0.0002	0.0001	0.0028	0.0009	0.0044
ASN_HID	-0.0014	-0.0031	-0.0158	-0.0008	-0.0027
ASN_ILE	-0.0024	0.0038	0.0105	0.0016	-0.0091
ASN_LEU	0.0006	0.0040	-0.0031	-0.0017	0.0090
ASN_LYS	0.0003	-0.0035	-0.0139	-0.0038	0.0203

Table 5.14. (cont'd)

ASN_MET	0.0006	-0.0056	-0.0207	0.0011	0.0018
ASN_PHE	-0.0006	-0.0011	-0.0245	-0.0007	-0.0012
ASN_PRO	0.0008	0.0012	-0.0289	0.0038	0.0239
ASN_SER	0.0009	-0.0017	0.0067	-0.0022	-0.0271
ASN_THR	0.0007	0.0053	-0.0203	-0.0027	0.0094
ASN_TRP	0.0020	-0.0076	-0.0134	0.0044	0.0039
ASN_TYR	0.0007	0.0046	0.0145	0.0008	-0.0073
ASN_VAL	0.0019	0.0049	0.0248	0.0005	-0.0243
ASP_ASP	-0.0001	-0.0054	0.0023	-0.0005	-0.0215
ASP_CYS	0.0029	-0.0046	-0.0202	-0.0004	0.0269
ASP_GLN	-0.0009	-0.0088	-0.0203	0.0021	-0.0080
ASP_GLU	-0.0007	-0.0020	-0.0086	-0.0058	0.0123
ASP_GLY	0.0023	-0.0047	0.0048	-0.0061	0.0221
ASP_HID	0.0025	0.0004	0.0108	-0.0013	-0.0068
ASP_ILE	-0.0052	-0.0043	-0.0049	0.0067	-0.0143
ASP_LEU	-0.0004	0.0000	-0.0094	-0.0042	0.0097
ASP_LYS	-0.0019	0.0016	0.0008	0.0026	0.0097
ASP_MET	-0.0030	0.0050	0.0054	-0.0033	0.0047
ASP_PHE	-0.0013	-0.0151	-0.0077	0.0021	-0.0118
ASP_PRO	0.0025	0.0043	-0.0089	0.0010	0.0085
ASP_SER	0.0010	-0.0005	-0.0128	0.0038	0.0143
ASP_THR	0.0003	-0.0079	-0.0060	-0.0037	-0.0082
ASP_TRP	0.0013	0.0046	0.0117	0.0049	-0.0041
ASP_TYR	-0.0025	-0.0019	-0.0043	0.0020	-0.0026
ASP_VAL	-0.0011	0.0007	-0.0071	-0.0017	0.0048
CYS_CYS	-0.0010	0.0003	-0.0103	-0.0018	0.0116
CYS_GLN	-0.0006	-0.0021	0.0153	0.0012	-0.0062
CYS_GLU	-0.0022	0.0010	0.0101	-0.0030	-0.0045
CYS_GLY	-0.0018	0.0029	-0.0037	0.0002	-0.0043
CYS_HID	0.0015	-0.0018	0.0105	-0.0034	-0.0007
CYS_ILE	0.0003	-0.0014	0.0008	-0.0006	-0.0154
CYS_LEU	-0.0009	0.0000	0.0088	0.0016	0.0039
CYS_LYS	-0.0019	0.0015	-0.0083	-0.0022	0.0154
CYS_MET	-0.0012	-0.0026	0.0008	0.0021	0.0135
CYS_PHE	0.0014	0.0041	-0.0016	-0.0013	-0.0094
CYS_PRO	0.0005	0.0019	-0.0057	-0.0017	0.0093
CYS_SER	0.0020	0.0019	0.0123	0.0022	0.0139
CYS_THR	0.0023	-0.0014	-0.0012	-0.0022	-0.0077
CYS_TRP	-0.0028	-0.0023	0.0076	0.0038	0.0041
CYS_TYR	-0.0019	-0.0025	-0.0092	0.0025	0.0115
CYS_VAL	0.0011	0.0012	-0.0081	-0.0043	0.0041
GLN_GLN	0.0000	-0.0062	0.0001	0.0002	-0.0119
GLN_GLU	0.0004	-0.0008	0.0096	0.0001	-0.0123
GLN_GLY	0.0001	-0.0101	0.0017	0.0009	0.0086
GLN_HID	-0.0019	-0.0018	-0.0197	-0.0004	-0.0001
GLN_ILE	-0.0014	0.0042	-0.0054	0.0005	-0.0053
GLN_LEU	0.0020	-0.0045	0.0024	-0.0040	0.0000
GLN_LYS	-0.0012	-0.0019	-0.0060	0.0002	-0.0067
GLN_MET	-0.0008	0.0018	0.0257	0.0011	-0.0110
GLN_PHE	0.0006	0.0051	-0.0089	0.0026	0.0081
GLN_PRO	0.0023	0.0023	-0.0154	-0.0025	0.0059

Table 5.14. (cont'd)

GLN_SER	-0.0022	0.0065	0.0055	0.0001	-0.0148
GLN_THR	-0.0015	0.0050	0.0114	-0.0001	-0.0060
GLN_TRP	0.0004	0.0090	-0.0034	0.0013	-0.0027
GLN_TYR	-0.0022	-0.0071	-0.0085	0.0022	-0.0160
GLN_VAL	0.0024	0.0068	0.0143	-0.0006	0.0013
GLU_GLU	-0.0018	-0.0003	-0.0107	-0.0003	0.0028
GLU_GLY	0.0013	0.0046	-0.0047	0.0005	0.0027
GLU_HID	-0.0020	-0.0016	0.0139	-0.0017	0.0034
GLU_ILE	0.0022	0.0066	0.0101	-0.0015	0.0122
GLU_LEU	0.0006	0.0038	0.0160	0.0017	-0.0035
GLU_LYS	-0.0022	-0.0002	-0.0116	-0.0006	0.0086
GLU_MET	0.0019	-0.0034	-0.0071	0.0007	-0.0250
GLU_PHE	-0.0026	0.0007	0.0052	0.0014	0.0033
GLU_PRO	0.0042	-0.0016	-0.0005	-0.0042	0.0114
GLU_SER	0.0005	0.0052	0.0112	-0.0018	0.0186
GLU_THR	0.0008	-0.0013	-0.0181	-0.0004	0.0260
GLU_TRP	0.0013	-0.0066	0.0010	0.0010	-0.0019
GLU_TYR	-0.0020	0.0048	-0.0051	-0.0029	0.0206
GLU_VAL	-0.0006	0.0005	-0.0036	-0.0002	0.0404
GLY_GLY	-0.0003	-0.0035	-0.0019	-0.0005	0.0043
GLY_HID	-0.0025	0.0035	0.0138	-0.0015	-0.0167
GLY_ILE	0.0008	-0.0019	-0.0110	0.0002	-0.0024
GLY_LEU	0.0001	-0.0054	-0.0138	0.0018	0.0036
GLY_LYS	0.0004	-0.0005	0.0019	-0.0013	-0.0006
GLY_MET	0.0005	-0.0025	-0.0245	0.0006	-0.0002
GLY_PHE	0.0003	-0.0020	-0.0003	-0.0001	-0.0008
GLY_PRO	0.0010	-0.0043	-0.0015	0.0026	-0.0125
GLY_SER	0.0005	-0.0014	0.0026	-0.0003	-0.0053
GLY_THR	-0.0004	0.0025	0.0026	0.0012	-0.0085
GLY_TRP	-0.0005	0.0072	-0.0032	-0.0005	0.0112
GLY_TYR	0.0005	-0.0026	-0.0001	-0.0002	0.0133
GLY_VAL	-0.0003	0.0001	0.0068	-0.0004	0.0042
HID_HID	-0.0006	-0.0023	-0.0056	0.0024	-0.0128
HID_ILE	-0.0015	-0.0055	0.0125	-0.0024	-0.0081
HID_LEU	-0.0007	-0.0005	-0.0004	0.0021	0.0018
HID_LYS	-0.0014	0.0011	0.0010	-0.0022	0.0030
HID_MET	0.0005	-0.0027	0.0044	0.0008	0.0006
HID_PHE	-0.0003	0.0002	0.0100	-0.0003	-0.0133
HID_PRO	0.0030	0.0003	0.0124	0.0021	0.0091
HID_SER	0.0017	-0.0004	0.0014	-0.0013	-0.0082
HID_THR	0.0002	0.0079	0.0138	-0.0013	-0.0128
HID_TRP	-0.0022	-0.0048	0.0017	0.0010	0.0009
HID_TYR	0.0040	-0.0133	-0.0087	-0.0024	-0.0072
HID_VAL	-0.0010	-0.0003	-0.0041	0.0056	-0.0101
ILE_ILE	0.0016	-0.0077	-0.0255	0.0000	0.0262
ILE_LEU	0.0012	-0.0029	-0.0061	-0.0021	0.0007
ILE_LYS	0.0003	-0.0006	-0.0176	-0.0004	0.0169
ILE_MET	-0.0020	0.0038	0.0136	0.0007	-0.0017
ILE_PHE	-0.0007	0.0103	0.0033	-0.0017	-0.0059
ILE_PRO	0.0016	-0.0018	-0.0110	0.0026	0.0132
ILE_SER	-0.0007	0.0002	-0.0025	0.0012	0.0120

Table 5.14. (cont'd)

ILE_THR	0.0014	0.0105	-0.0162	-0.0001	0.0002
ILE_TRP	-0.0029	0.0096	-0.0199	0.0000	0.0133
ILE_TYR	-0.0003	0.0099	0.0142	-0.0052	-0.0020
ILE_VAL	0.0010	-0.0007	-0.0099	-0.0005	0.0047
LEU_LEU	0.0005	0.0041	0.0023	0.0000	0.0035
LEU_LYS	-0.0001	0.0007	-0.0075	0.0012	0.0076
LEU_MET	0.0007	-0.0046	0.0077	0.0009	0.0079
LEU_PHE	0.0012	0.0121	-0.0019	0.0043	-0.0097
LEU_PRO	-0.0032	-0.0018	-0.0118	0.0018	0.0071
LEU_SER	0.0029	0.0026	-0.0105	0.0022	-0.0144
LEU_THR	-0.0010	-0.0047	-0.0051	0.0016	-0.0084
LEU_TRP	0.0014	-0.0047	0.0157	0.0033	-0.0046
LEU_TYR	-0.0010	-0.0013	-0.0130	0.0005	-0.0007
LEU_VAL	-0.0017	0.0077	-0.0090	0.0027	0.0197
LYS_LYS	-0.0013	0.0009	-0.0075	-0.0001	-0.0130
LYS_MET	-0.0003	-0.0026	-0.0109	-0.0012	0.0067
LYS_PHE	0.0010	0.0062	-0.0023	0.0004	-0.0043
LYS_PRO	-0.0016	0.0027	-0.0017	-0.0024	0.0064
LYS_SER	-0.0009	-0.0022	-0.0145	0.0023	0.0184
LYS_THR	-0.0009	0.0094	-0.0048	0.0004	-0.0257
LYS_TRP	0.0002	-0.0023	0.0170	0.0008	-0.0080
LYS_TYR	0.0006	-0.0057	-0.0223	0.0004	0.0238
LYS_VAL	0.0001	-0.0023	0.0225	-0.0009	-0.0133
MET_MET	0.0010	0.0011	-0.0043	0.0006	0.0022
MET_PHE	-0.0005	0.0017	-0.0025	0.0045	-0.0040
MET_PRO	0.0018	0.0066	-0.0011	-0.0029	0.0046
MET_SER	-0.0018	0.0071	0.0082	0.0004	-0.0020
MET_THR	0.0005	0.0039	-0.0103	-0.0008	0.0199
MET_TRP	0.0014	0.0072	-0.0041	0.0009	0.0311
MET_TYR	0.0012	0.0053	0.0002	0.0015	-0.0046
MET_VAL	0.0002	-0.0006	0.0026	0.0002	0.0084
PHE_PHE	-0.0001	0.0028	0.0042	0.0014	-0.0032
PHE_PRO	-0.0020	0.0003	0.0001	0.0011	0.0046
PHE_SER	-0.0017	-0.0107	-0.0119	0.0006	0.0158
PHE_THR	-0.0010	0.0077	-0.0001	-0.0026	-0.0024
PHE_TRP	-0.0017	-0.0020	-0.0083	0.0018	0.0076
PHE_TYR	0.0008	-0.0003	0.0001	-0.0020	-0.0122
PHE_VAL	0.0041	-0.0029	-0.0035	0.0021	0.0088
PRO_PRO	-0.0007	-0.0018	-0.0029	-0.0018	0.0025
PRO_SER	0.0014	0.0037	0.0076	-0.0009	-0.0022
PRO_THR	0.0020	0.0037	-0.0143	-0.0010	0.0083
PRO_TRP	0.0018	-0.0046	0.0027	0.0017	-0.0006
PRO_TYR	0.0009	-0.0007	-0.0070	-0.0002	-0.0041
PRO_VAL	-0.0019	0.0030	0.0102	0.0023	-0.0008
SER_SER	-0.0008	-0.0021	0.0093	-0.0010	0.0036
SER_THR	0.0022	0.0039	-0.0120	-0.0008	-0.0026
SER_TRP	0.0005	-0.0015	-0.0137	0.0025	0.0189
SER_TYR	0.0002	0.0007	0.0008	0.0000	-0.0025
SER_VAL	0.0008	0.0045	-0.0113	0.0010	0.0013
THR_THR	-0.0001	0.0007	-0.0020	-0.0004	-0.0083
THR_TRP	0.0003	-0.0128	-0.0146	0.0022	0.0059

Table 5.14. (cont'd)

THR_TYR	-0.0009	-0.0034	-0.0045	-0.0003	-0.0023
THR_VAL	0.0020	-0.0047	0.0037	-0.0010	-0.0060
TRP_TRP	0.0007	-0.0020	0.0109	0.0007	0.0094
TRP_TYR	-0.0005	0.0039	-0.0193	-0.0007	0.0078
TRP_VAL	-0.0013	0.0004	0.0159	-0.0020	-0.0042
TYR_TYR	0.0019	0.0132	-0.0219	-0.0006	0.0068
TYR_VAL	-0.0015	-0.0104	0.0202	0.0006	-0.0025
VAL_VAL	0.0009	-0.0005	-0.0238	0.0031	0.0047
Maximum	0.0042	0.0175	0.0584	0.0067	0.0404
Minimum	-0.0052	-0.0151	-0.0434	-0.0061	-0.0402

Table 5.15. A brief comparison between FFENCODER and Amber with ff94 and ff14SB parameter sets.

		ff94		ff14SB	
		Maximum (kcal/mol)	Minimum (kcal/mol)	Maximum (kcal/mol)	Minimum (kcal/mol)
Single amino acid test set	Dihedral	0.0010	-0.0011	0.0033	-0.0011
	1_4_VdW	0.0065	-0.0051	0.0063	-0.0104
	1_4_EEL	0.0076	-0.0177	0.0061	-0.0191
	VdW	0.0021	-0.0010	0.0036	-0.0015
	EEL	0.0244	-0.0085	0.0247	-0.0098
Dipeptide test set	Dihedral	0.0047	-0.0098	0.0042	-0.0052
	1_4_VdW	0.0143	-0.0152	0.0175	-0.0151
	1_4_EEL	0.0807	-0.0357	0.0584	-0.0434
	VdW	0.0086	-0.0063	0.0067	-0.0061
	EEL	0.0485	-0.0519	0.0404	-0.0402

Table 5.16. Accuracy and native ranking comparison between RF models based on ff94 and ff14SB with other scoring functions.

		accuracy			Native ranking		
		average	highest	lowest	average	highest	lowest
RF models with ff94	IMP_100	0.988	0.999	0.963	5.88	17.17	1.47
	IMP_200	0.987	1.000	0.963	5.93	17.87	1.11
	IMP_300	0.989	0.999	0.958	4.40	14.66	1.28
	IMP_400	0.988	0.999	0.965	5.01	15.06	1.26
	IMP_500	0.992	0.999	0.986	3.52	6.19	1.23
RF models with ff14SB	IMP_100	0.987	1.000	0.973	5.16	10.45	1.11
	IMP_200	0.987	0.999	0.962	5.83	16.85	1.17
	IMP_300	0.991	1.000	0.972	4.28	14.28	1.04
	IMP_400	0.987	0.999	0.965	4.80	11.11	1.51
	IMP_500	0.990	0.996	0.973	4.06	9.34	1.85
RF models with KECSA2	IMP_100	0.963	0.987	0.931	10.62	21.64	4.86
	IMP_200	0.972	0.989	0.947	8.01	11.17	6.05
	IMP_300	0.976	0.993	0.933	8.69	13.31	3.92
	IMP_400	0.977	0.965	0.956	6.06	15.34	2.36
	IMP_500	0.981	0.994	0.965	7.95	17.59	2.77
RWplus	-	0.916	-	-	23.43	-	-
DFIRE	-	0.886	-	-	31.35	-	-
dDFIRE	-	0.904	-	-	26.49	-	-
GOAP	-	0.917	-	-	23.09	-	-

Table 5.17. 1st decoy RMSD and TM-score comparison between RF models based on ff94 and ff14SB with other scoring functions.

		1st decoy RMSD			1st decoy TM-score		
		average	highest	lowest	average	highest	lowest
RF models with FF94	IMP_100	4.55	6.03	3.38	0.614	0.679	0.550
	IMP_200	4.83	5.51	4.05	0.618	0.654	0.551
	IMP_300	4.84	5.92	3.88	0.618	0.704	0.561
	IMP_400	4.67	5.50	3.67	0.621	0.669	0.564
	IMP_500	4.93	6.36	3.8	0.611	0.672	0.536
RF models with FF14SB	IMP_100	4.74	5.47	3.79	0.621	0.688	0.563
	IMP_200	4.83	5.76	4.48	0.620	0.656	0.587
	IMP_300	4.87	5.83	3.78	0.609	0.639	0.572
	IMP_400	5.22	5.66	4.57	0.594	0.628	0.564
	IMP_500	4.80	5.96	4.15	0.616	0.660	0.591
RF models with KECSA2	IMP_100	4.62	5.49	3.77	0.634	0.685	0.574
	IMP_200	5.06	5.71	4.18	0.598	0.656	0.561
	IMP_300	4.78	5.43	3.56	0.604	0.679	0.546
	IMP_400	4.74	5.48	4.00	0.629	0.684	0.582
	IMP_500	4.57	5.72	3.52	0.614	0.695	0.536
RWplus	-	4.53	-	-	0.622	-	-
DFIRE	-	4.51	-	-	0.623	-	-
dDFIRE	-	4.44	-	-	0.625	-	-
GOAP	-	4.45	-	-	0.674	-	-

Table 5.18. Distribution of decoys' lowest RMSD values in the combined decoy set.

Protein name	Lowest RMSD	Protein name	Lowest RMSD	Protein name	Lowest RMSD	Protein name	Lowest RMSD	Protein name	Lowest RMSD
1r69	0.195	1gaf	0.797	1nbv	1.166	1opd	1.931	1nkl	5.325
1di2	0.202	256b	0.803	1mcp	1.169	1cew	1.954	1dxt	5.401
1o2f	0.204	1gyv	0.804	1c8c	1.173	1c9o	2.007	1trl	5.401
1no5	0.205	1jnu	0.805	1mla	1.185	1gpt	2.012	1ew4	5.703
1a32	0.212	1tfi	0.812	6fab	1.191	1mfa	2.044	1pgb	5.874
1org	0.220	1thx	0.817	1of9	1.212	1hbg	2.049	2mta	5.944
1hbk	0.224	2cro	0.823	1fai	1.225	1ubi	2.116	1bg8	6.050
2reb	0.227	1shf	0.831	1tif	1.230	1ash	2.254	1kpe	6.244
1cy5	0.265	1bm8	0.843	1kjs	1.233	1cc8	2.264	2vik	6.256
2cr7	0.271	1fpt	0.847	1yuh	1.254	4sdh	2.287	1gky	6.275
1b72	0.290	1mlb	0.873	1fbi	1.280	2chf	2.334	1bbh	6.478
1abv	0.290	1hil	0.878	1mrd	1.283	1mn8	2.447	1cei	6.504
1cqk	0.343	1vcc	0.887	1ikf	1.308	1scj	2.457	1eyv	6.515
1aoy	0.346	2cgr	0.893	1ctf	1.319	1b0n	2.466	1a68	6.620
1mky	0.403	1igf	0.910	1sn3	1.341	2cmd	2.528	5cro	6.627
1bq9	0.419	1frg	0.917	1kem	1.358	1fc2	2.547	1dkt	6.713
1kvi	0.425	1igc	0.917	1dfb	1.366	1gdm	2.609	1cpc	6.873
1hda	0.429	1hbh	0.919	2a0b	1.371	1hdd	2.722	1gvp	6.957
1myg	0.433	1dbb	0.931	1opg	1.371	1bba	2.788	1beo	7.045
1csp	0.438	1for	0.938	1fig	1.378	1gnu	2.816	1vie	7.152
1ogw	0.457	3icb	0.961	4rxn	1.379	1b3a	2.883	1bk2	7.157
2f3n	0.500	2pcy	0.962	1gig	1.410	1hlb	2.884	1jwe	7.879
1ah9	0.510	3hfm	0.963	4pti	1.421	1a19	2.889	1tit	8.190
1elw	0.513	1ggi	0.968	1ngq	1.429	1hlm	2.994	1bgf	8.395
1ten	0.536	1jel	0.970	1ecd	1.478	1fna	3.009	1dhn	8.422
2dhb	0.541	1egx	0.973	1b4b	1.491	2lhb	3.024	1bkr	8.521
1pgx	0.552	1fgv	0.974	1mam	1.493	1acf	3.120	1cg5	8.545
1nps	0.559	1n0u	0.980	1nmb	1.493	1igd	3.171	smd3	8.612
1myj	0.563	1plg	1.004	1fvc	1.494	1cid	3.348	1rnb	8.778
1af7	0.573	1myt	1.004	1ibg	1.501	1eaf	3.455	811b	9.143
1g1c	0.576	1igm	1.028	1sfp	1.532	1cau	3.469	1who	9.170
1itp	0.615	1iai	1.041	1eap	1.545	1c2r	3.490	1lis	9.269
1sro	0.632	2fbj	1.046	1rmf	1.545	1bl0	3.641	1ptq	9.403
2pgh	0.641	1aiu	1.052	1igi	1.555	1wit	3.712	1fkb	9.682
1fad	0.644	1jhl	1.053	1bbd	1.580	1mdc	3.793	2acy	9.781
1dvf	0.645	1flr	1.065	1vge	1.602	1onc	3.807	2ci2	9.794
1dtj	0.656	1ail	1.071	1ith	1.614	1utg	3.900	2pna	10.803
1tig	0.670	1ne3	1.083	2gfb	1.630	1vls	3.984	1tul	11.621
1emy	0.690	1baf	1.087	1iib	1.662	2ovo	3.989	1lga	12.349
1bab	0.694	1dcj	1.111	2fb4	1.675	1eh2	4.002	1urn	12.423
1tet	0.735	1vfa	1.135	1ig5	1.704	1e6i	4.131	1col	12.427
1hkl	0.737	1nsn	1.146	1mbs	1.715	1dtk	4.333	1hz6	12.468
1fvd	0.772	1acy	1.148	1flp	1.719	4ubp	4.724	4sbv	14.050
1hsy	0.772	1ncb	1.149	1enh	1.745	4icb	4.781	1mup	14.293
1fo5	0.782	1gjx	1.156	1mba	1.824	1lou	4.974	2sim	15.671
1lht	0.784	1ucb	1.159	7fab	1.850	1fca	5.190	2afn	19.744
1bbj	0.791	1ind	1.162	8fab	1.913	1ugh	5.311		

Table 5.19. Accuracy comparisons between scoring functions with and without RF refinement.

		Accuracy		
		average	highest	lowest
RF models with FF94	IMP_100	0.988	0.999	0.963
	IMP_500	0.992	0.999	0.986
RF models with FF14SB	IMP_100	0.987	1.000	0.973
	IMP_500	0.990	0.996	0.973
FF94 without RF refinement	IMP_100	0.626	-	-
	IMP_500	0.564	-	-
FF14SB without RF refinement	IMP_100	0.656	-	-
	IMP_500	0.660	-	-

Table 5.20. Accuracy comparisons between scoring functions with and without force field parameters.

		Accuracy		
		average	highest	lowest
RF models with FF94	IMP_100	0.988	0.999	0.963
	IMP_500	0.992	0.999	0.986
RF models with FF14SB	IMP_100	0.987	1.000	0.973
	IMP_500	0.990	0.996	0.973
RF models without force field potentials	IMP_100	0.679	0.768	0.593
	IMP_500	0.712	0.780	0.572

APPENDIX B: FIGURES

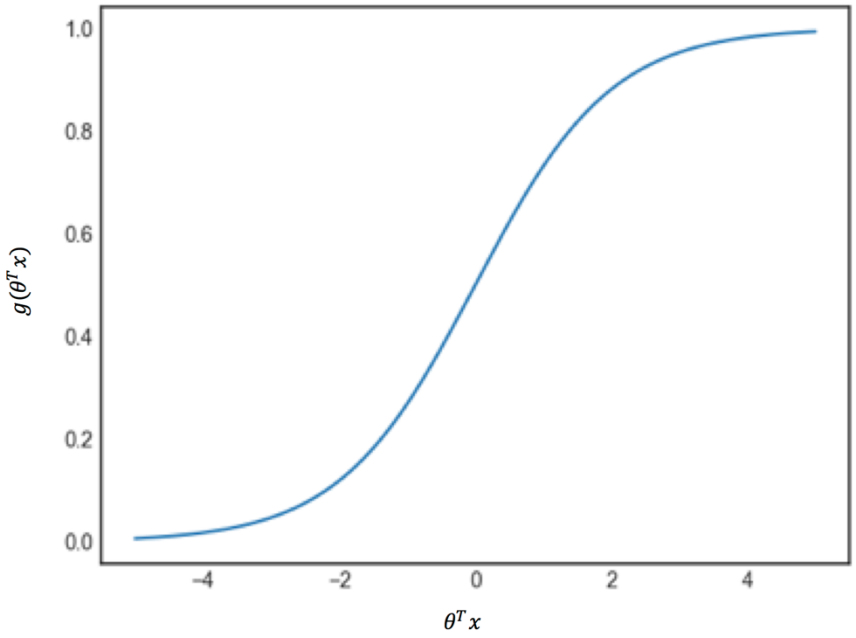

Figure 1.1. The shape of the sigmoid function showed in equation (1.3).

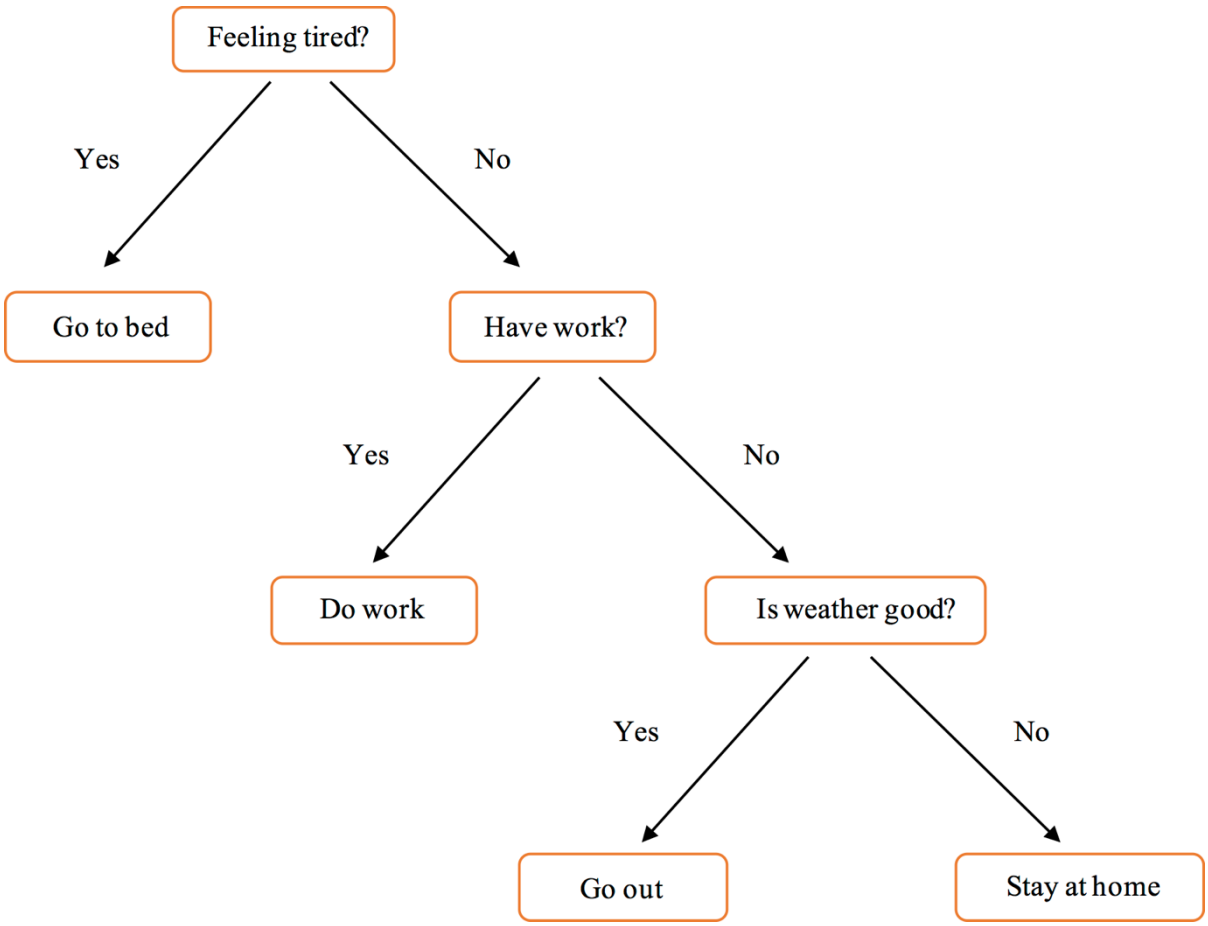

Figure 1.2. An example of a simple decision tree.

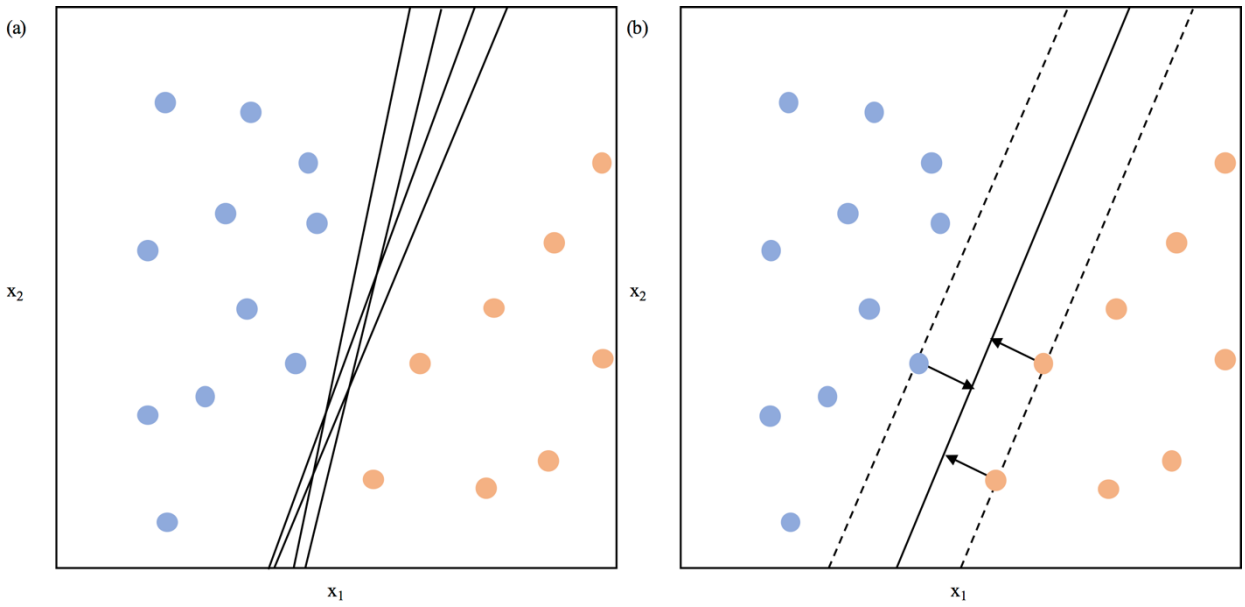

Figure 1.3. The comparison between general hyperplane and hyperplane generated by the maximum margin classifier. (a) shows there are infinite hyperplanes can be used to separate the data set. (b) shows the hyperplane with the largest margin of separation width.

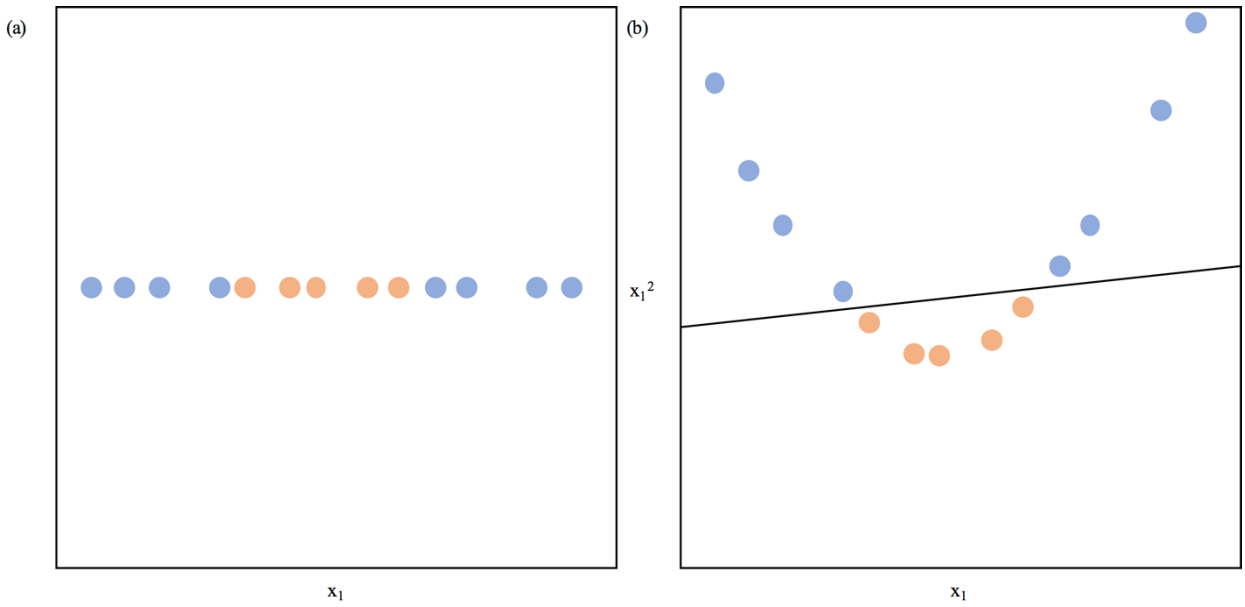

Figure 1.4. An example of a support vector machine algorithm. (a) An example of a dataset that cannot be separated by a hyperplane. The observations in the data set are one dimensional points. (b) A polynomial kernel equation is used to change those one dimensional points to 2D points. A hyperplane (black line) can be used to separate those observations.

descriptor	x_1	x_2	x_3	x_4	x_5	x_6
example1						
example2						
example3						
example4						
example5						
example6						

Figure 1.5. An example of bagging.

descriptor	x_1	x_2	x_3	x_4	x_5	x_6
example1						
example2						
example3						
example4						
example5						
example6						

Figure 1.6. An example of RF.

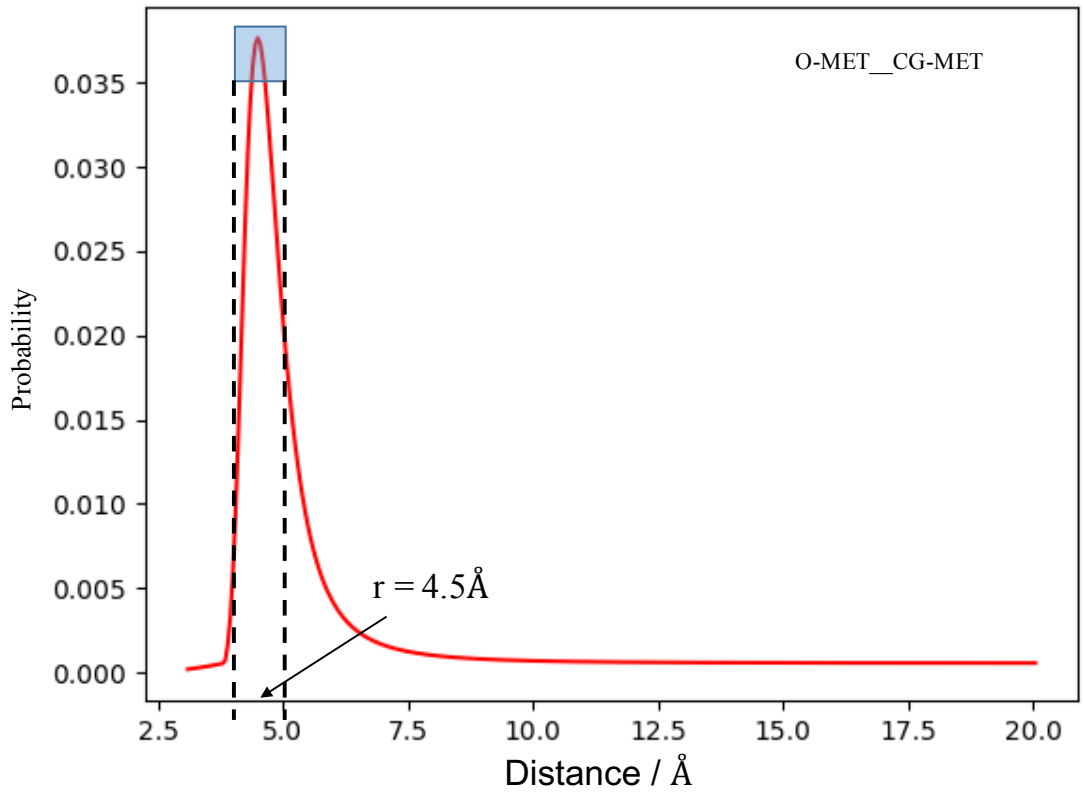

Figure 2.1. Probability versus distance plot for atom pair O-MET__CG-MET, shaded region is the averaged region with the length of 1 Å.

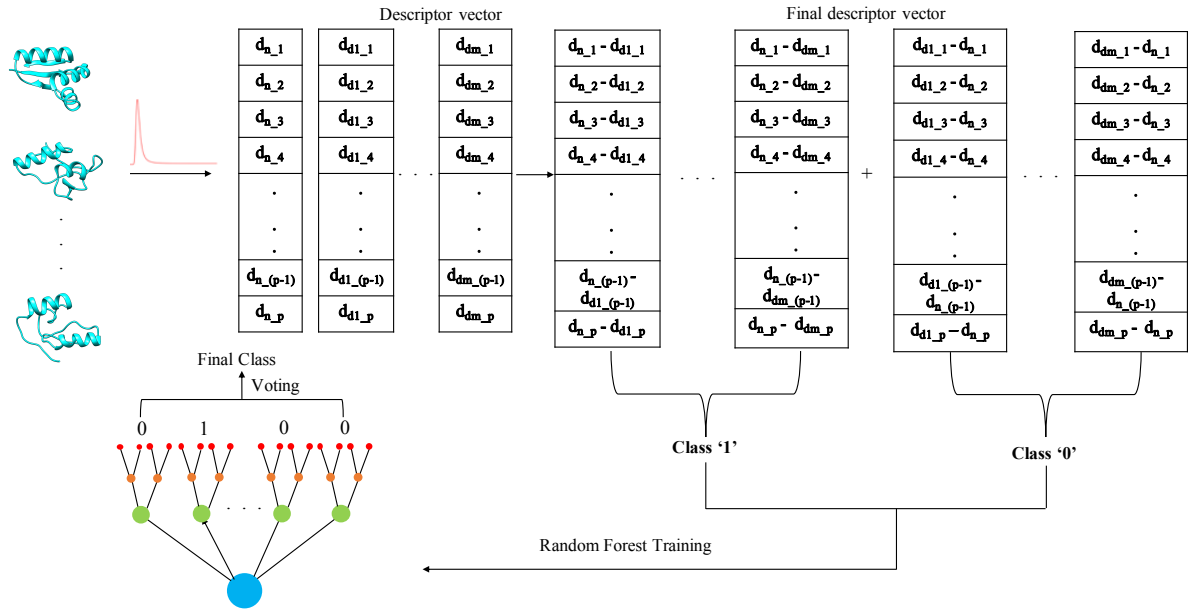

Figure 2.2. The protocol used to build up the Random Forest model. Parameter p (equals to 16029) represents the total number of atom pairs in KECSA2. Parameter *n* represents the native structure, $d_1, \ldots, d_m$ are the 1st , …, mth decoy structures.

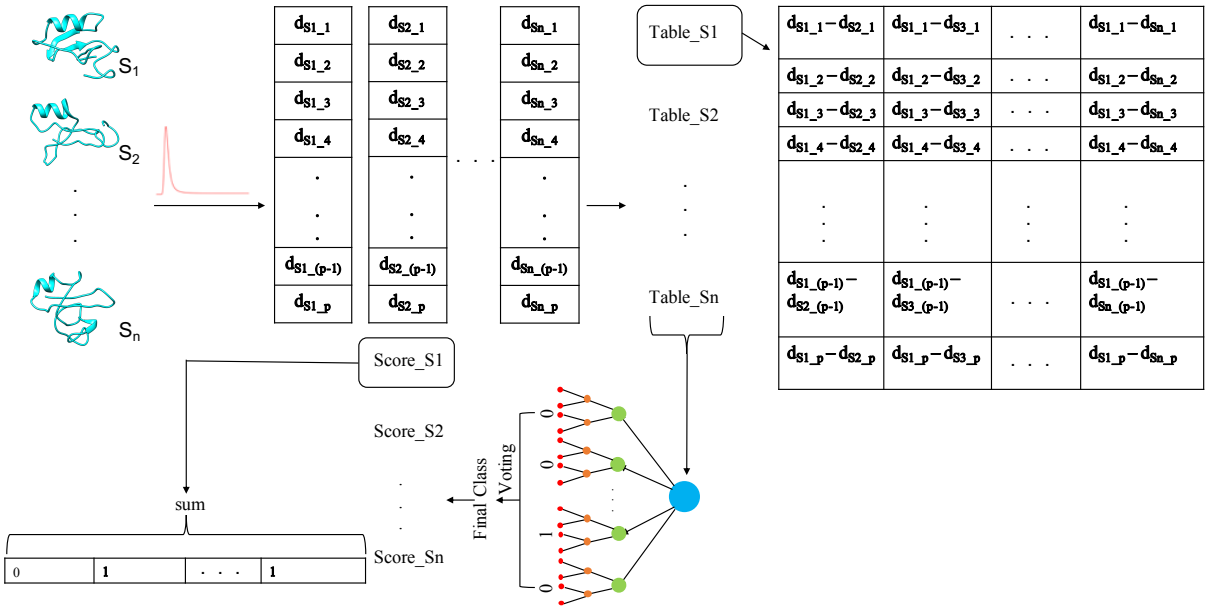

Figure 2.3. Protocol for generating the ranking list for the Random Forest model. Parameter *p* (equal to 16029) represents the total number of atom pairs in KECSA2. S_1, S_2, … , S_n are the 1st , 2nd , … , nth protein structures with the same residue sequence.

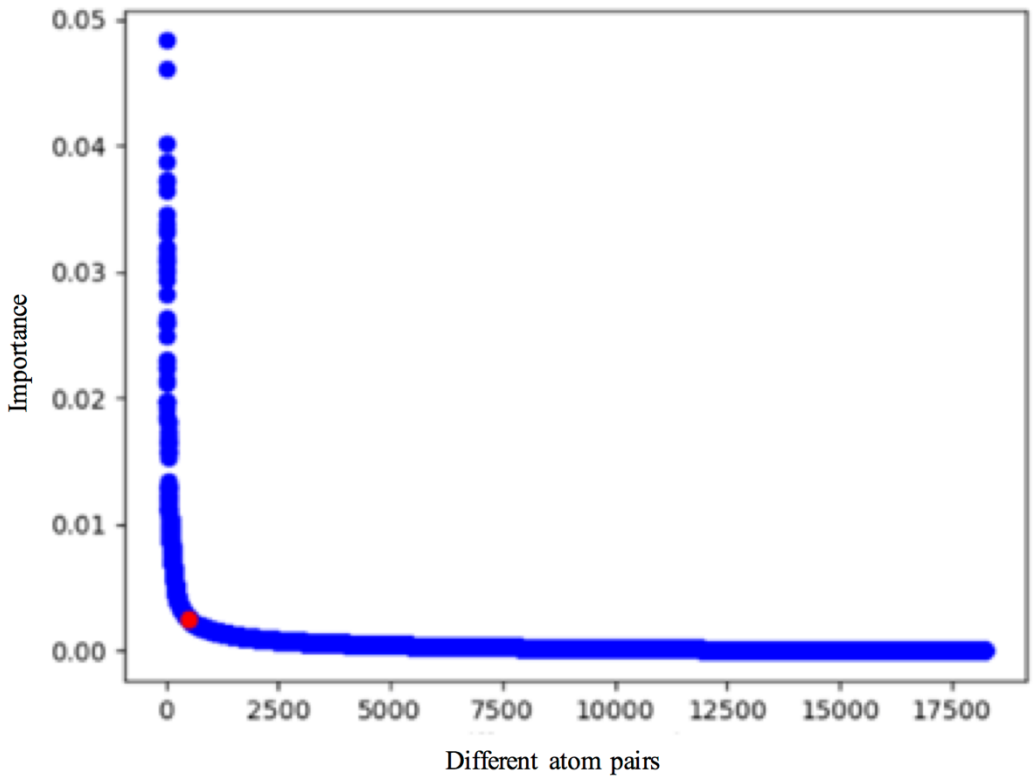

Figure 3.1. Feature importance analysis results for the overall decoy set. The red point represents the 500th atom pair.

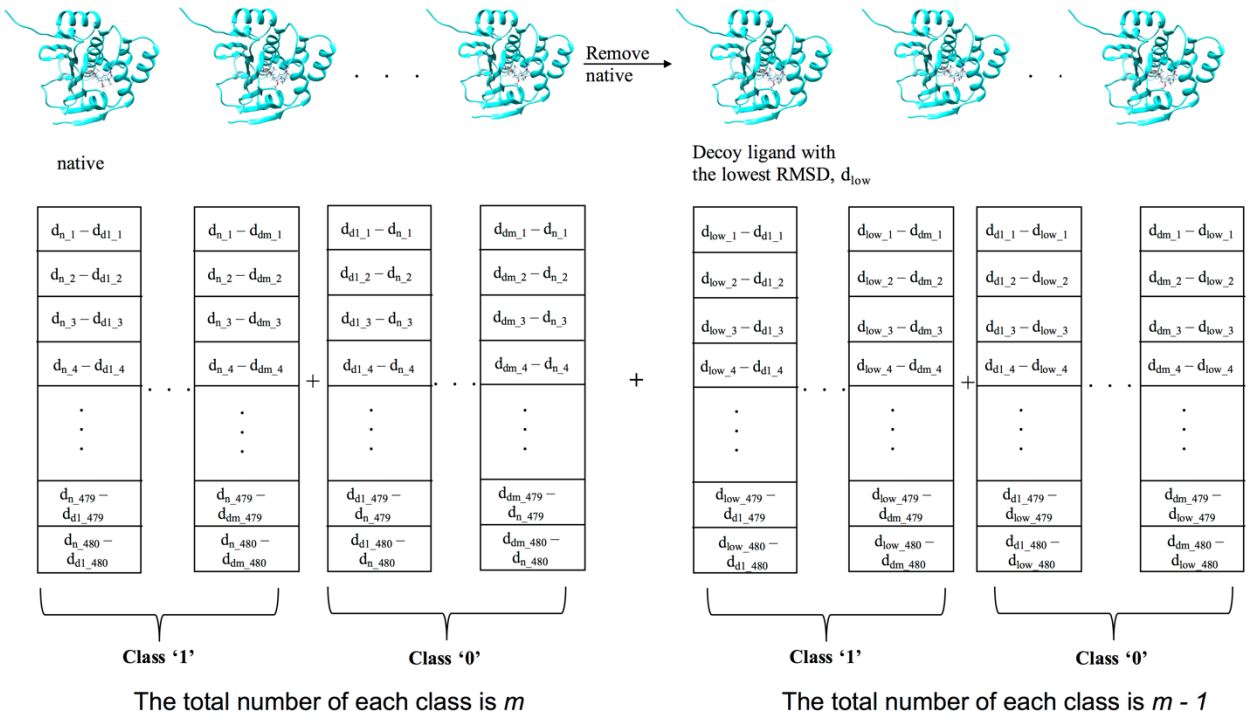

Figure 4.1. The protocol used to include the comparison information between best decoy binding pose and other decoy poses.

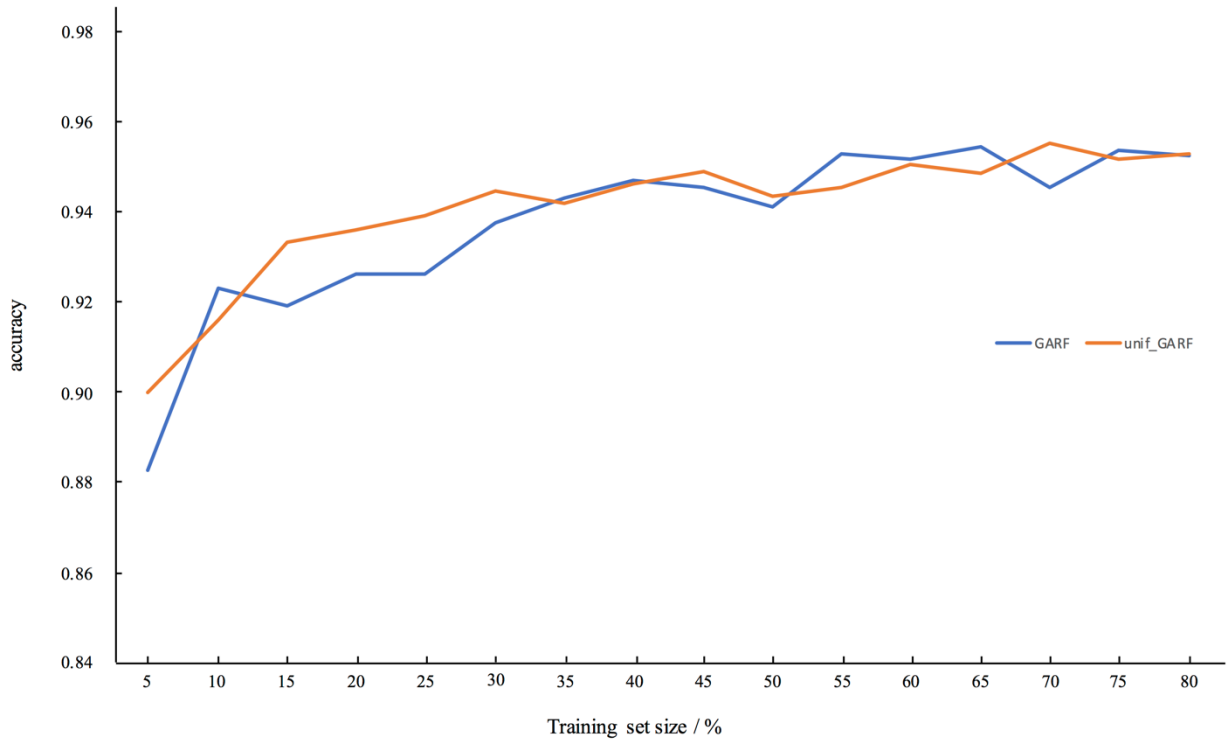

Figure 4.2. Accuracy trend from RF models based on original(blue line) and uniform(orange line) GARF data sets.

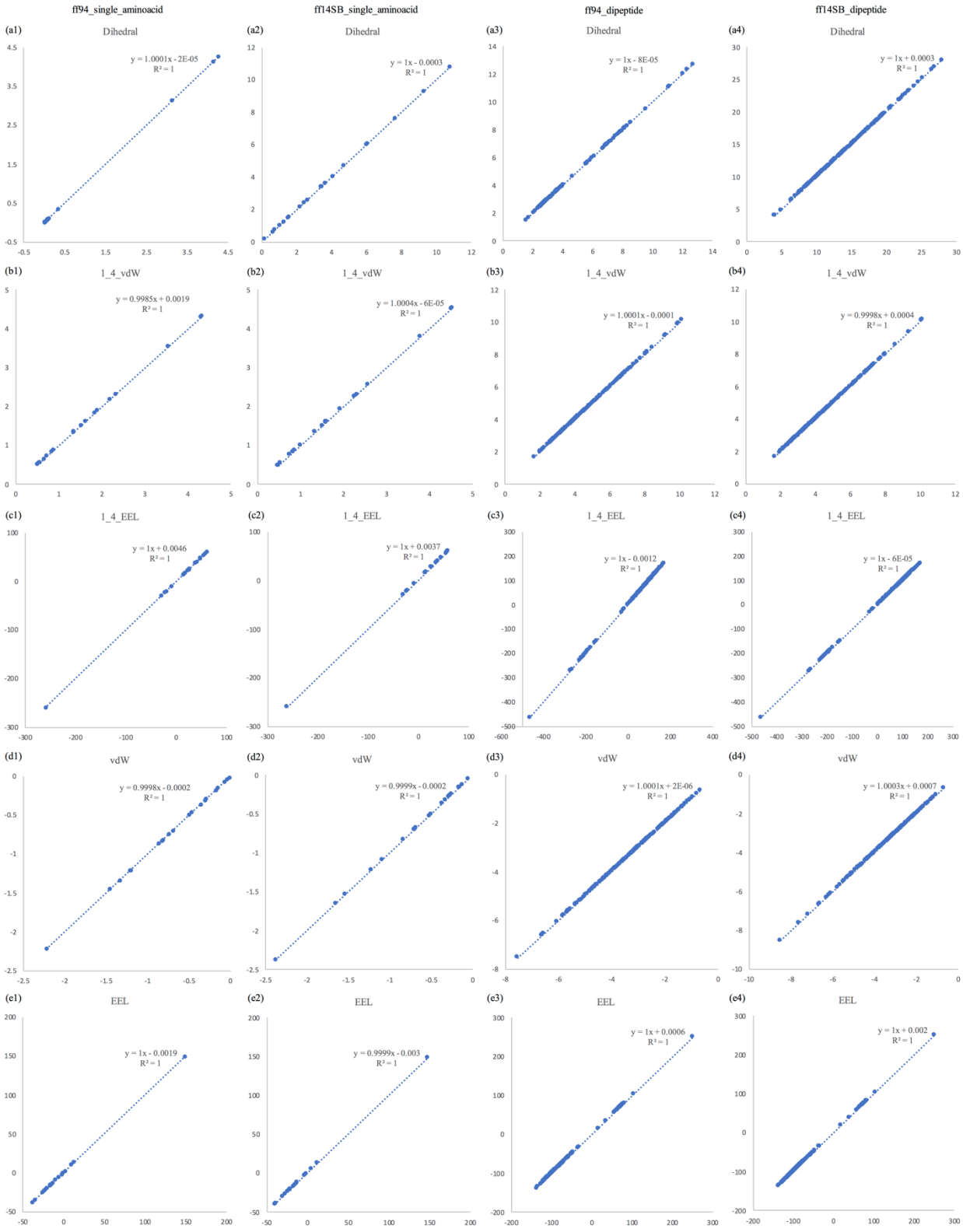

Figure 5.1. Comparisons between energies calculated by FFENCODER and the Amber software package. (a) - (e) are results for dihedral, 1_4 Van der Waals, 1_4 electrostatics, Van der Waals,

Figure 5.1. (cont'd)
and electrostatic energies. Columns (1) and (3) are comparisons of single amino acid and dipeptide test sets for ff94. Columns (2) and (4) are comparisons of single amino acid and dipeptide test sets for ff14SB.

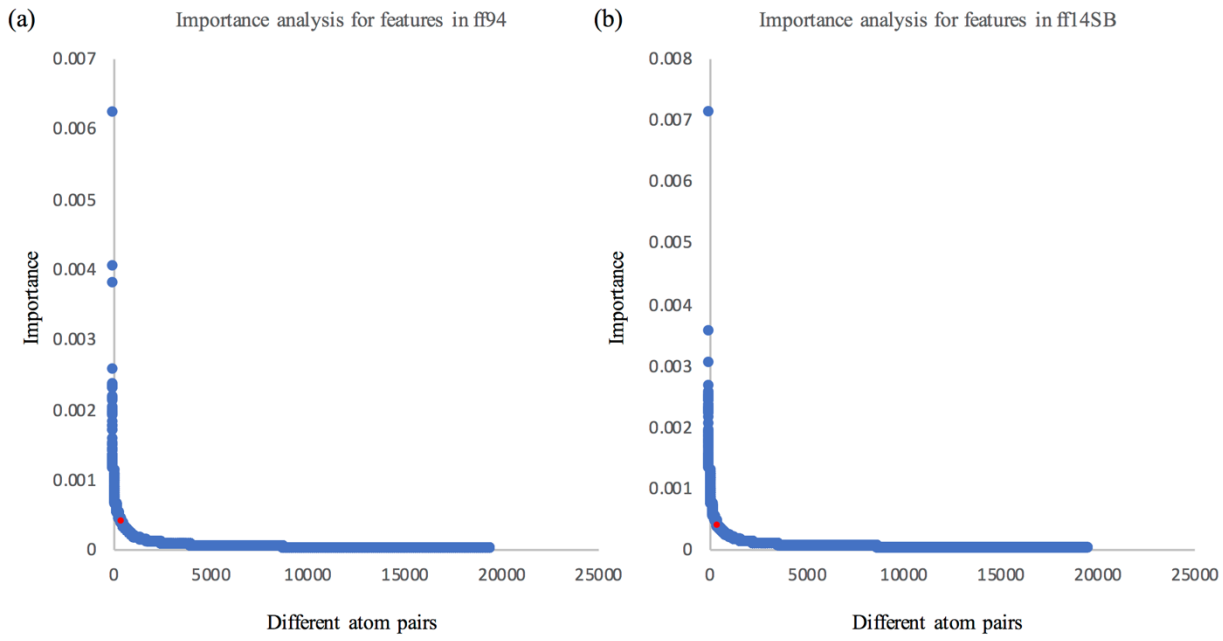

Figure 5.2. Importance analysis for features in ff94 and ff14SB parameter set. The red point in each plot represents the 500th most important feature in each parameter set.

APPENDIX C: COPYRIGHT NOTICE

Chapter 2, 3, 4, and 5 of this dissertation (include its supporting information) are adapted with permissions from several publications listed below:

(1) Adapted with permission from ref 117. Copyright 2019 American Chemistry Society.
(2) Adapted with permission from ref 124. Copyright 2019 American Chemistry Society.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. John, B.; Sali, A. Comparative protein structure modeling by iterative alignment, model building and model assessment. Nucleic Acids Res. **2003**, *31*, 3982−3992.

2. Zhou, H.; Skolnick, J. GOAP: A Generalized Orientation- Dependent, All-Atom Statistical Potential for Protein Structure Prediction. Biophys. J. **2011**, *101*, 2043−2052.

3. Skolnick, J. In quest of an empirical potential for protein structure prediction. Curr. Opin. Struct. Biol. **2006**, *16*, 166−171.

4. Sippl, M. J. Knowledge-based potentials for proteins. Curr. Opin. Struct. Biol. **1995**, *5*, 229−235.

5. Jernigan, R. L.; Bahar, I. Structure-derived potentials and protein simulations. Curr. Opin. Struct. Biol. **1996**, *6*, 195−209.

6. Moult, J. Comparison of database potentials and molecular mechanics force fields. Curr. Opin. Struct. Biol. **1997**, *7*, 194−199.

7. Lazaridis, T.; Karplus, M. Effective energy functions for protein structure prediction. Curr. Opin. Struct. Biol. **2000**, *10*, 139−145.

8. Gohlke, H.; Klebe, G. Statistical potentials and scoring functions applied to protein-ligand binding. Curr. Opin. Struct. Biol. **2001**, *11*, 231−235.

9. Russ, W. P.; Ranganathan, R. Knowledge-based potential functions in protein design. Curr. Opin. Struct. Biol. **2002**, *12*, 447− 452.

10. Buchete, N. V.; Straub, J. E.; Thirumalai, D. Development of novel statistical potentials for protein fold recognition. Curr. Opin. Struct. Biol. **2004**, *14*, 225−232.

11. Poole, A. M.; Ranganathan, R. Knowledge-based potentials in protein design. Curr. Opin. Struct. Biol. **2006**, *16*, 508−513.

12. Zhou, Y.; Zhou, H.; Zhang, C.; Liu, S. What is a desirable statistical energy functions for proteins and how can it be obtained? Cell Biochem. Biophys. **2006**, *46*, 165−174.

13. Ma, J. Explicit Orientation Dependence in Empirical Potentials and Its Significance to Side-Chain Modeling. Acc. Chem. Res. **2009**, *42*, 1087−1096.

14. Gilis, D.; Biot, C.; Buisine, E.; Dehouck, Y.; Rooman, M. Development of Novel Statistical Potentials Describing Cation−π Interactions in Proteins and Comparison with Semiempirical and Quantum Chemistry Approaches. J. Chem. Inf. Model. **2006**, *46*, 884− 893.

15. Hendlich, M.; Lackner, P.; Weitckus, S.; Floeckner, H.; Froschauer, R.; Gottsbacher, K.; Casari, G.; Sippl, M. J. Identification of native protein folds amongst a large number of incorrect models. The calculation of low energy conformations from potentials of mean force. J. Mol. Biol. **1990**, *216*, 167−180.

16. Hoppe, C.; Schomburg, D. Prediction of protein thermo- stability with a direction- and distance-dependent knowledge-based potential. Protein Sci. **2005**, *14*, 2682−2692.

17. Jones, D. T.; Taylort, W. R.; Thornton, J. M. A new approach to protein fold recognition. Nature **1992**, *358*, 86−89.

18. Kolinński,A.;Bujnicki,J.M.Generalizedproteinstructure prediction based on combination of fold-recognition with de novo folding and evaluation of models. Proteins: Struct., Funct., Genet. **2005**, *61*, 84−90.

19. Miyazawa, S.; Jernigan, R. L. Estimation of Effective Interresidue Contact Energies from Protein Crystal Structures: Quasi-Chemical Approximation. Macromolecules **1985**, *18*, 534−552.

20. DeBolt, S. E.; Skolnick, J. Evaluation of atomic level mean force potentials via inverse folding and inverse refinement of protein structures: atomic burial position and pairwise non-bonded interactions. Protein Eng., Des. Sel. **1996**, *9*, 637−655.

21. Zhang, C.; Vasmatzis, G.; Cornette, J. L.; DeLisi, C. Determination of atomic desolvation energies from the structures of crystallized proteins. J. Mol. Biol. **1997**, *267*, 707−726.

22. Tobi, D.; Elber, R. Distance-dependent, pair-potential for protein folding: Results from linear optimization. Proteins: Struct., Funct., Genet. **2000**, *41*, 40−46.

23. Wu, Y.; Lu, M.; Chen, M.; Li, J.; Ma, J. OPUS-Ca: A knowledge-based potential function requiring only Cα positions. Protein Sci. **2007**, *16*, 1449−1463.

24. Zhang, Y.; Kolinski, A.; Skolnick, J. TOUCHSTONE II: A New Approach to Ab Initio Protein Structure Prediction. Biophys. J. **2003**, *85*, 1145−1164.

25. Sippl, M. J. Calculation of Conformational Ensembles from Potential of Mean Force. J. Mol. Biol. **1990**, *213*, 859−883.

26. Lu, H.; Skolnick, J. A distance dependent atomic knowledge-based potential for improved protein structure selection. Proteins: Struct., Funct., Genet. **2001**, *44*, 223−232.

27. Lu, M.; Dousis, A. D.; Ma, J. (2008). OPUS-PSP: An Orientation-dependent Statistical All-atom Potential Derived from Side-chain Packing. J. Mol. Biol. **2008**, *376*, 288−301.

28. Samudrala, R.; Moult, J. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. J. Mol. Biol. **1998**, *275*, 895−916.

29. Shen, M.-Y.; Sali, A. Statistical potential for assessment and prediction of protein structures. Protein Sci. **2006**, *15*, 2507−2524.

30. Yang, Y.; Zhou, Y. Specific interactions for ab initio folding of protein terminal regions with secondary structures. Proteins: Struct., Funct., Genet. **2008**, *72*, 793−803.

31. Skolnick, J.; Kolinski, A.; Ortiz, A. Derivation of protein-specific pair potentials based on weak sequence fragment similarity. Proteins: Struct., Funct., Genet. **2000**, *38*, 3−16.

32. Zhang, J.; Zhang, Y. A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. PLoS One **2010**, *5*, No. e15386.

33. Zhou, H.; Zhou, Y. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Sci. **2002**, *11*, 2714−2726.

34. Benkert, P.; Tosatto, S. C. E.; Schomburg, D. QMEAN: A comprehensive scoring function for model quality assessment. Proteins: Struct., Funct., Genet. **2008**, *71*, 261−277.

35. Cao, R.; Bhattacharya, D.; Hou, J.; Cheng, J. DeepQA: Improving the estimation of single protein model quality with deep belief networks. BMC Bioinf. **2016**, *17*, 1−9.

36. Uziela, K.; Shu, N.; Wallner, B.; Elofsson, A. ProQ3: Improved model quality assessments using Rosetta energy terms. Sci. Rep. **2016**, *6*, 1−10.

37. Uziela, K.; Hurtado, D. M.; Shu, N.; Wallner, B.; Elofsson, A. ProQ3D: Improved model quality assessments using deep learning. Bioinformatics **2017**, *33*, 1578−1580.

38. Manavalan, B.; Lee, J. SVMQA: support-vector-machine-based protein single-model quality assessment. Bioinformatics **2017**, *33*, 2496−2503.

39. Olechnovic̆, K.; Venclovas, Č. VoroMQA: Assessment of protein structure quality using interatomic contact areas. Proteins: Struct., Funct., Genet. **2017**, *85*, 1131−1145.

40. Hurtado, D. M.; Uziela, K.; Elofsson, A. Deep transfer learning in the assessment of the quality of protein models. arXiv:1804.06281, **2018**

41. Simons, K. T.; Kooperberg, C.; Huang, E.; Baker, D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J. Mol. Biol. **1997**, *268*, 209−225.

42. Balakrishnan, S.; Kamisetty, H.; Carbonell, J. G.; Lee, S.-I.; Langmead, C. J. Learning generative models for protein fold families. Proteins: Struct., Funct., Genet. **2011**, *79*, 1061−1078.

43. Leaver-Fay, A.; Tyka, M.; LEWIS, S. M.; Lange, O. F.; Thompson, J.; Jacak, R.; Kaufman, K. W.; Renfrew, P. D.; Smith, C. A.; Sheffler, W.; Davis, I. W.; Cooper, S.; Treuille, A.; Mandell,

D. J.; Richter, F.; Ban, Y. A.; Fleishman, S. J.; Corn, J. E.; Kim, D. E.; Lyskov, S.; Berrondo, M.; Mentzer, S.; Popovic, Z.; Havranek, J. J.; Karanicolas, J.; Das, R.; Meiler, J.; Kortemme, T.; Gray, J. J.; Kuhlman, B.; Baker, D.; Bradley, P. ROSETTA3:anobject-oriented-softwaresuiteforthe imulation and design of macromolecules. Methods Enzymol. **2011**, *487*, 545−574.

44. Wood, C. W.; Bruning, M.; Ibarra, A. A.; Bartlett, G. J.; Thomson, A. R.; Sessions, R. B.; Brady, R. L.; Woolfson, D. V. CCBuilder:aninteractiveweb-basedtoolforbuilding, designing and as- sessing coiled-coil protein assemblies. Bioinformatics **2014**, *30*, 3029− 3035.

45. Negron, C.; Keating, A. E. Multistate protein design using CLEVER and CLASSY. Methods Enzymol. **2013**, *523*, 171−190.

46. Smadbeck, J.; Peterson, M. B.; Khoury, G. A.; Taylor, M. S.; Floudas, C. A. Protein WISDOM: a workbench for in silico de novo design of biomolecules. J. Visualized Exp. **2013**, *77*, No. e50476.

47. Dahiyat, B. I.; Mayo, S. L. Protein design automation. Protein Sci. **1996**, *5*, 895−903.

48. Dahiyat, B. I.; Mayo, S. L. De novo protein design: fully automated sequence selection. Science **1997**, *278*, 82−87.

49. Kamisetty, H.; Ovchinnikov, S.; Baker, D. Assessing the utility of coevolution-based residue−residue contact predictions in a sequence- and structure-rich era. Proc. Natl. Acad. Sci. U. S. A. **2013**, *110*, 15674−15679.

50. Ovchinnikov, S.; Kamisetty, H.; Baker, D. Robust and accurate prediction of residue−residue interactions across protein interfaces using evolutionary information. eLife **2014**, *3*, No. e02030.

51. Ovchinnikov, S.; Kinch, L.; Park, H.; Liao, Y.; Pei, J.; Kim, D. E.; Kamisetty, H.; Grishin, N. V.; Baker, D. Large-scale determination of previously unsolved protein structures using evolutionary information. eLife **2015**, *4*, No. e09248.

52. Källberg, M.; Wang, H.; Wang, S.; Peng, J.; Wang, Z.; Lu, H.; Xu, J. Template-based protein structure modeling using the RaptorX web server. *Nature Protocols*, **2012**, *7*(8), 1511–1522.

53. Kuhlman, B.; Dantas, G.; Ireton, G. C.; Varani, G.; Stoddard, B. L.; Baker, D. Design of a novel globular protein fold with atomic-level accuracy. Science **2003**, *302*, 1364−1368.

54. Huang, P.-S.; Ban, Y.-E. A.; Richter, F.; Andre, I.; Vernon, R.; Schief, W. R.; Baker, D. RosettaRemodel:ageneralizedframeworkfor- flexible backbone protein design. PLoS One **2011**, *6*, No. e24109.

55. Harbury, P. B.; Plecs, J. J.; Tidor, B.; Alber, T.; Kim, P. S. High- resolution protein design with backbone freedom. Science **1998**, *282*, 1462−1467.

56. Thomson, A. R.; Wood, C. W.; Burton, A. J.; Bartlett, G. J.; Sessions, R. B.; Brady, R. L.; Woolfson, D. N. Computational design of water-solubleα-helical barrels. Science **2014**, *346*, 485−488.

57. Grigoryan, G.; DeGrado, W. F. Probing designability via a generalized model of helical bundle geometry. J. Mol. Biol. **2011**, *405*, 1079−1100.

58. Huang, P.-S.; Oberdorfer, G.; Xu, C.; Pei, X. Y.; Nannenga, B. L.; Rogers, J. M.; DiMaio, F.; Gonen, T.; Luisi, B.; Baker, D. High thermodynamic stability of parametrically designed helical bundles. Science **2014**, *346*, 481−485.

59. Regan, L.; DeGrado, W. F. Characterization of a helical protein designed from first principles. Science **1988**, *241*, 976−978.

60. Lin, Y.-R.; Koga, N.; Tatsumi-Koga, R.; Liu, G.; Clouser, A. F.; Montelione, G. T.; Baker, D. Control over overall shape and size in de novo designed proteins. Proc. Natl. Acad. Sci. U. S. A. **2015**, *112*, E5478−E5485.

61. Koga, N.; Tatsumi-Koga, R.; Liu, G.; Xiao, R.; Acton, T. B.; Montelione, G. T.; Baker, D. Principles for designing ideal protein structures. Nature **2012**, *491*, 222−227.

62. MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.; Prodhom, B.; Reiher, W. E., III; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe,M.;Wiorkiewicz-Kuczera,J.;Yin,D.;Karplus,M.All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins. J. Phys. Chem. B **1998**, *102*, 3586−3616.

63. Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. CHARMM: A program for macro- molecular energy, minimization, and dynamics calculations. J. Comput. Chem. **1983**, *4*, 187−217.

64. Weiner, S. J.; Kollman, P. A.; Nguyen, D. T.; Case, D. A. An all atom force field for simulations of proteins and nucleic acids. J. Comput. Chem. **1986**, *7*, 230−252.

65. Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber biomolecular simulation programs. J. Comput. Chem. **2005**, *26*, 1668−1688.

66. Arnautova, Y. A.; Jagielska, A.; Scheraga, H. A. A New Force Field (ECEPP-05) for Peptides, Proteins, and Organic Molecules. J. Phys. Chem. B **2006**, *110*, 5025−5044.

67. Ponder, J. W.; Case, D. A. Force fields for protein simulations. Adv. Protein Chem. **2003**, *66*, 27−85.

68. Jagielska, A.; Wroblewska, L.; Skolnick, J. Protein model refinement using an optimized physics-based all-atom force field. Proc. Natl. Acad. Sci. U. S. A. **2008**, *105*, 8268−8273.

69. Meng, E. C.; Shoichet, B. K.; Kuntz, I. D. Automated Docking with Grid-Based Energy Evaluation. *J. Comput. Chem.* **1992**, *13*, 505− 524.

70. Makino, S.; Kuntz, I. D. Automated Flexible Ligand Docking: Method and Its Application for Database Search. *J. Comput. Chem.* **1997**, *18*, 1812−1825.

71. Goodsell, D. S.; Morris, G. M.; Olson, A. J. Automated Docking of Flexible Ligands: Applications of AutoDock. *J. Mol. Recog.* **1996,** *9,* 1− 5.

72. Ortiz, A. R.; Pisabarro, M. T.; Gago, F.; Wade, R. C. Prediction of Drug Binding Affinities by Comparative Binding Energy Analysis. *J. Med. Chem*. **1995**, *38*, 2681−2691.

73. Yin, S.; Biedermannova, L.; Vondrasek, J.; Dokholyan, N. V. MedusaScore: An Accurate Force Field-Based Scoring Function for Virtual Drug Screening. *J. Chem. Inf. Model.* **2008**, *48*, 1656−1662.

74. Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Development and Validation of a Genetic Algorithm for Flexible Docking. *J. Mol. Biol.* **1997**, *267*, 727−748.

75. Aqvist, J.; Medina, C.; Samuelsson, J. E. New Method for Predicting Binding Affinity in Computer-Aided Drug Design. *Protein Eng.* **1994**, *7*, 385−391.

76. Almlof, M.; Brandsdal, B. O.; Aqvist, J. Binding Affinity Prediction with Different Force Fields: Examination of the Linear Interaction Energy Method. *J. Comput. Chem.* **2004**, *25*, 1242−1254.

77. Carlson, H. A.; Jorgensen, W. L. Extended Linear Response Method for Determining Free Energies of Hydration. *J. Phys. Chem.* **1995**, *99*, 10667−10673.

78. Jones-Hertzog, D. K.; Jorgensen, W. L. Binding Affinities for Sulfonamide Inhibitors with Human Thrombin Using Monte Carlo Simulations with a Linear Response Method. *J. Med. Chem*. **1997**, *40*, 1539−1549.

79. Kollman, P. A.; Massova, I.; Reyes, C.; Kuhn, B.; Huo, S.; Chong, L.; Lee, M.; Lee, T.; Duan, Y.; Wang, W.; Donini, O.; Cieplak, P.; Srinivasan, J.; Case, D. A.; Cheatham, T. E. Calculating Structures and Free Energies of Complex Molecules: Combining Molecular Mechanics and Continuum Models. *Acc. Chem. Res*. **2000**, *33*, 889− 897.

80. DeWitte, R. S.; Shakhnovich, E. I. SMoG: de Novo Design Method Based on Simple, Fast, and Accurate Free Energy Estimates. 1. Methodology and Supporting Evidence. *J. Am. Chem. Soc*. **1996**, *118*, 11733−11744.

81. Grzybowski, B. A.; Ishchenko, A. V.; Shimada, J.; Shakhnovich, E. I. From Knowledge-Based Potentials to Combinatorial Lead Design in Silico. *Acc. Chem. Res.* **2002**, *35*, 261−269.

82. Muegge, I.; Martin, Y. C. A General and Fast Scoring Function for Protein-Ligand Interactions: A Simplified Potential Approach. *J. Med. Chem.* **1999**, *42*, 791−804.

83. Muegge, I. A Knowledge-Based Scoring Function for Protein-Ligand Interactions: Probing the Reference State. *Perspect. Drug Discovery Des.* **2000,** *20*, 99−114.

84. Muegge, I. Effect of Ligand Volume Correction on PMF Scoring. *J. Comput. Chem.* **2001**, *22*, 418−425.

85. Gohlke, H.; Hendlich, M.; Klebe, G. Knowledge-Based Scoring Function to Predict Protein-Ligand Interactions. *J. Mol. Biol*. **2000**, *295*, 337−356.

86. Velec, H. F. G.; Gohlke, H.; Klebe, G. DrugScore(CSD): Knowledge-Based Scoring Function Derived from Small Molecule Crystal Data with Superior Recognition Rate of Near-Native Ligand Poses and Better Affinity Prediction. *J. Med. Chem.* **2005**, *48*, 6296− 6303.

87. Neudert, G.; Klebe, G. DSX: A Knowledge-Based Scoring Function for the Assessment of Protein-Ligand Complexes. *J. Chem. Inf. Model.* **2011**, *51*, 2731−2745.

88. Huang, S.-Y.; Zou, X. An Iterative Knowledge-Based Scoring Function to Predict Protein-Ligand Interactions: I. Derivation of Interaction Potentials. *J. Comput. Chem.* **2006**, *27*, 1865−1875.

89. Huang, S. Y.; Zou, X. An Iterative Knowledge-Based Scoring Function to Predict Protein-Ligand Interactions: II. Validation of the Scoring Function. *J. Comput. Chem.* **2006**, *27*, 1876−1882.

90. Huang, S. Y.; Zou, X. Inclusion of Solvation and Entropy in the Knowledge-Based Scoring Function for Protein-Ligand Interactions. *J. Chem. Inf. Model.* **2010**, *50*, 262−273.

91. Zheng, Z.; Merz, K. M. Development of the Knowledge-Based and Empirical Combined Scoring Algorithm (KECSA) To Score Protein−Ligand Interactions. *J. Chem. Inf. Model.* **2013**, *53*, 1073− 1083.

92. Wang, R.; Lai, L.; Wang, S. Further Development and Validation of Empirical Scoring Functions for Structure-Based Binding Affinity Prediction. *J. Comput. Aided. Mol. Des*. **2002**, *16*, 11−26.

93. Bohm, H. J. The Development of A Simple Empirical Scoring Function to Estimate the Binding Constant for a Protein-Ligand Complex of Known Three-Dimensional Structure. *J. Comput. Aided. Mol. Des.* **1994**, *8*, 243−256.

94. Verkhivker, G.; Appelt, K.; Freer, S. T.; Villafranca, J. E. Empirical Free Energy Calculations of Ligand-Protein Crystallographic Complexes. I. Knowledge-Based Ligand-Protein Interaction Potentials Applied to the Prediction of Human Immunodeficiency Virus 1 Protease Binding Affinity. *Protein Eng.* **1995**, *8*, 677−691.

95. Eldridge, M. D.; Murray, C. W.; Auton, T. R.; Paolini, G. V.; Mee, R. P. Empirical Scoring Functions: I. The Development of a Fast Empirical Scoring Function to Estimate the Binding Affinity of Ligands in Receptor Complexes. *J. Comput. Aided. Mol. Des.* **1997**, *11*, 425−445.

96. Murray, C. W.; Auton, T. R.; Eldridge, M. D. Empirical Scoring Functions. II. The Testing of an Empirical Scoring Function for the Prediction of Ligand-Receptor Binding Affinities and the Use of Bayesian Regression to Improve the Quality of the Model. *J. Comput. Aided. Mol. Des.* **1998**, *12*, 503−519.

97. Friesner, R. A.; Banks, J. L.; Murphy, R. B.; Halgren, T. A.; Klicic, J. J.; Mainz, D. T.; Repasky, M. P.; Knoll, E. H.; Shelley, M.; Perry, J. K.; Shaw, D. E.; Francis, P.; Shenkin, P. S. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J. Med. Chem.* **2004**, *47 (7),* 1739− 1749.

98. Friesner, R. A.; Murphy, R. B.; Repasky, M. P.; Frye, L. L.; Greenwood, J. R.; Halgren, T. A.; Sanschagrin, P. C.; Mainz, D. T. Extra Precision Glide: Docking and Scoring Incorporating a Model of Hydrophobic Enclosure for Protein-Ligand Complexes. *J. Med. Chem.* **2006**, *49*, 6177−6196.

99. Jiménez, J.; Škalič, M.; Martínez-Rosell, G.; De Fabritiis, G.. KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *J. Chem. Inf. Model.* **2018,** *58(2)*, 287–296.

100. Cang, Z., Wei, G. W. Integration of Element Specific Persistent Homology and Machine Learning for Protein-Ligand Binding Affinity Prediction. *Int. J. Numer Meth. Biomed. Engng.* **2018,** *34(2)*, 1–17.

101. Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein-Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model.* **2017,** *57(4),* 942–957.

102. Deng, W.; Breneman, C.; Embrechts, M. J. Predicting Protein-Ligand Binding Affinities Using Novel Geometrical Descriptors and Machine-Learning Methods. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 699− 703.

103. Zhang, S.; Golbraikh, A.; Tropsha, A. Development of Quantitative Structure-Binding Affinity Relationship Models Based on Novel Geometrical Chemical Descriptors of the Protein-Ligand Interfaces. *J. Med. Chem.* **2006**, *49*, 2713−2724.

104. Durrant, J. D.; McCammon, J. A. NNScore: A Neural-Network- Based Scoring Function for the Characterization of Protein-Ligand Complexes. *J. Chem. Inf. Model.* **2010**, *50*, 1865−1871.

105. Durrant, J. D.; McCammon, J. A. NNScore 2.0: A Neural- Network Receptor-Ligand Scoring Function. *J. Chem. Inf. Model.* **2011**, *51*, 2897−2903.

106. Ballester, P. J.; Mitchell, J. B. O. A Machine Learning Approach to Predicting Protein−Ligand Binding Affinity with Applications to Molecular Docking. *Bioinformatics* **2010**, *26*, 1169−1175.

107. Ballester, P. J.; Schreyer, A.; Blundell, T. L. Does a More Precise Chemical Description of Protein−Ligand Complexes Lead to More Accurate Prediction of Binding Affinity*? J. Chem. Inf. Model.* **2014**, *54*, 944−955.

108. Zilian, D.; Sotriffer, C. A. SFCscore^{RF}: A Random Forest-Based Scoring Function for Improved Affinity Prediction of Protein-Ligand Complexes. *J. Chem. Inf. Model.* **2013**, *53*, 1923−1933.

109. Li, G. B.; Yang, L. L.; Wang, W. J.; Li, L. L.; Yang, S. Y. ID- Score: A New Empirical Scoring Function Based on a Comprehensive Set of Descriptors Related to Protein−Ligand Interactions. *J. Chem. Inf. Model.* **2013**, *53*, 592−600.

110. Deng, Z.; Chuaqui, C.; Singh, J. Structural Interaction Fingerprint (SIFt): A Novel Method for Analyzing Three-Dimensional Protein-Ligand Binding Interactions. *J. Med. Chem.* **2004**, *47*, 337− 344.

111. Dangetti, P. Journey from Statistics for Machine Learning. In *Statistical for Machine Learning,* Editing, S., Pagare, V., Singh, A., Pawanikar, M., Pawar, D. Ltd: Packt Publishing, Birmingham, United Kingdom, 2017; pp 9.

112. Li, Y.; Liu, Z.; Li, J.; Han, L.; Liu, J.; Zhao, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. *J. Chem. Inf. Model.* **2014,** *54(6)*, 1700–1716.

113. Liu, Z.; Li, Y.; Han, L.; Li, J.; Liu, J.; Zhao, Z.; Nie, W.; Liu, Y.; Wang, R. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics* **2015**, 31 (3): 405-412

114. Zheng, Z.; Pei, J.; Bansal, N.; Liu, H.; Song, L. F.; Merz, K. M. Generation of Pairwise Potentials Using Multidimensional Data Mining. *J. Chem. Theory Comput.* **2018,** *14(10),* 5045–5067.

115. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

116. Breiman, L. Random forests. *Machine Learning* **2001**, *45*, 5–32.

117. Pei, J.; Zheng, Z.; Merz, K. M. Random Forest Refinement of the KECSA2 Knowledge-Based Scoring Function for Protein Decoy Detection. *J. Chem. Inf. Model.* **2019,** *59,* 1919-1929.

118. D.A. Case, I. Y. B.-S., S.R. Brozell, D.S. Cerutti, T.E. Cheatham, III, V.W.D. Cruzeiro, T.A. Darden,; R.E. Duke, D. G., M.K. Gilson, H. Gohlke, A.W. Goetz, D. Greene, R Harris, N. Homeyer, Y. Huang,; S. Izadi, A. K., T. Kurtzman, T.S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, D.J.; Mermelstein, K. M. M., Y. Miao, G. Monard, C. Nguyen, H. Nguyen, I. Omelyan, A. Onufriev, F. Pan, R.; Qi, D. R. R., A. Roitberg, C. Sagui, S. Schott-Verdugo, J. Shen, C.L. Simmerling, J. Smith, R. SalomonFerrer, J. Swails, R.C. Walker, J. Wang, H. Wei, R.M. Wolf, X. Wu, L. Xiao, D.M. York and P.A. Kollman, AMBER 2018, University of California, San Francisco. **2018**.

119. Tsui, V.; Case, D. A., Theory and applications of the generalized Born solvation model in macromolecular simulations. *Biopolymers* **2000,** *56* (4), 275-91.

120. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J. Am. Chem. Soc.*, **1995**, *117*(19), 5179–5197.

121. Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *J. Chem. Theory Comput.*, **2015**, *11*(8), 3696–3713.

122. Ho, T. K. Random Decision Forests. Proc. Third Int. Conf. **1995**,*1*, 278−282.

123. Liaw, A.; Wiener, M.; Breiman, L.; Cutler, A. Package 'randomForest'. **2015**.

124. Pei, J.; Zheng, Z.; Kim, H.; Song, L. F.; Walworth, S.; Merz, M. R.; Merz, K. M. Random Forest Refinement of Pairwise Potentials for Protein–Ligand Decoy Detection. *J. Chem. Inf. Model.*, **2019**, *59*(7), 3305–3315