Structural bioinformatics

Machine Learning Accelerates MD-based Binding-Pose Prediction between Ligands and Proteins

Kei Terayama¹, Hiroaki Iwata², Mitsugu Araki³, Yasushi Okuno³,⁴ and Koji Tsuda¹,⁵,⁶

¹Department of Computational Biology and Medical Science, Graduate School of Frontier Sciences, The University of Tokyo, Chiba 277-8561, Japan, ²Foundation for Biomedical Research and Innovation, Hyogo 650-0047, Japan, ³RIKEN Advanced Institute for Computational Science, Hyogo 650-0047, Japan, ⁴Department of Biomedical Data Intelligence, Graduate School of Medicine, Kyoto University, Kyoto 606-8507, Japan, ⁵Center for Materials Research by Information Integration, NIMS, Ibaraki 305-0047, Japan, and ⁶RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan.

To whom correspondence should be addressed.

Associate Editor: Dr. Burkhard Rost

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Fast and accurate prediction of protein-ligand binding structures is indispensable for structure-based drug design (SBDD) and accurate estimation of binding free energy of drug candidate molecules in drug discovery. Recently, accurate pose prediction methods based on short-MD simulations such as MM-PBSA and MM-GBSA among generated docking poses have been used. Since molecular structures obtained from MD simulation depends on the initial condition, taking the average over different initial conditions leads to better accuracy. Prediction accuracy of protein-ligand binding poses can be improved with multiple runs at different initial velocity.

Results: This paper shows that a machine learning method, called Best Arm Identification (BAI), can optimally control the number of MD runs for each binding pose. It allows us to identify a correct binding pose with a minimum number of total runs. Our experiment using three proteins and eight inhibitors showed that the computational cost can be reduced substantially without sacrificing accuracy. This method can be applied for controlling all kinds of molecular simulations to obtain best results under restricted computational resources.

Availability: Code and data are available on Github at: https://github.com/tsudalab/bpbi

Contact: terayama@cbms.k.u-tokyo.ac.jp or tsuda@k.u-tokyo.ac.jp

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Structure-based drug discovery is becoming an essential tool for assisting fast and cost-efficient lead discovery and optimization (Lionta et al., 2014). Fast and accurate prediction of protein-ligand binding structure is an indispensable process for structure-based drug design, especially when binding structure has not been resolved by x-ray or nuclear magnetic resonance (NMR). In addition, accurate prediction of binding structure enables us to estimate binding free energy $\Delta G_{\text{bind}}$ with high accuracy using existing methods such as Linear Interaction Energy (LIE) (Åqvist et al., 1994), Molecular Mechanics/Poisson-Boltzmann Surface Area (MM-PBSA) (Strinivasan et al., 1998; Kollman et al., 2000), Molecular Mechanics/Generalized Born Surface Area (MM-GBSA) (Ondrias et al., 2000), or Massively Parallel Computation of Absolute binding Free Energy with well-Equilibrated states (MP-CAFEE) (Fujitani et al., 2009). Docking programs are often used as scoring function for binding pose prediction, but their accuracy is still low (Lavecchia and Di Giovanni, 2013; Hou
Recently, MM-PBSA-based binding pose prediction methods have also been proposed, in which Molecular Dynamics (MD) and MM-PBSA calculations are performed to estimate $\Delta G_{\text{bind}}$ on generated docking poses and accurately identify the best ones (Thompson et al., 2008; Hou et al., 2011). A number of methods for binding pose prediction and binding free energy estimation based on MD calculation have been proposed (Okimoto et al., 2009; Coluzza et al., 2010; Proctor et al., 2012).

However, binding pose prediction based on the MM-PBSA method is computationally expensive: in order to improve accuracy, MD simulation and MM-PBSA calculation (MD and MM-PBSA run) must be repeated and averaged over multiple initialization conditions, for each pose candidate (Sadiq et al., 2010; Genheden and Ryde, 2010; Mikulsikis et al., 2012, Berham and Hansmann, 2013). This approach incurs a huge computational cost, as existing studies use the same number of initial conditions for all poses (uniform sampling) and unnecessary calculation are performed for unpromising pose candidates. Figure 1 (A) shows the pose prediction with MD and MM-PBSA through uniform sampling.

The recently introduced Best Arm Identification (BAI) problem consists of optimizing the allocation of limited resources in order to find the best slot out of many slots. In this problem, we select a slot, called arm, and get a reward according to a probability distribution associated with it. These probability distributions are not known to us. The purpose of the problem is to minimize the total number of selections and reward-getting processes to find the best arm. A number of effective algorithms to achieve this purpose have been proposed (Bubeck et al., 2009; Audibert and Bubeck, 2010; Gabillon et al., 2012).

The BAI problem has attracted attention in the field of machine learning (Bubeck et al., 2009; Audert et al., 2002) and has been applied to various fields such as the design of clinical trials (Villar et al., 2015), recommendation systems of news and goods (Lietal., 2010), and the game of Go (Coulom, 2006; Silver et al., 2016). In this paper, we propose an effective pose prediction method based on a BAI algorithm. Using such an algorithm to optimally control MD and MM-PBSA runs, we can reduce the total number of MD and MM-PBSA runs to the total number of runs with uniform sampling, as seen in Figure 1.

To show the effectiveness of the proposed method, we conducted a pose prediction experiment, using the MD and MM-PBSA methods, on a dataset consisting of three proteins, cyclin-dependent kinase 2 (CDK2), heat shock protein 90 alpha (HSP90A), and coagulation factor X (FVIII), and eight inhibitors. We prepared 20 binding poses that included one or more "correct" poses for each complex. We then investigated how much of a computational cost reduction can be obtained on runs of MD and MM-PBSA, using our proposed method, compared with the widely used uniform sampling approach (where the same number of MD and MM-PBSA runs is performed for each pose).

The number of MM-PBSA runs was reduced by a factor of 1.76 to 6.67 compared to uniform sampling, with a success probability of correct pose identification fixed to 95%. In particular, in cases where pose prediction was difficult due to the small difference between correct and incorrect poses, $\Delta G_{\text{bind}}$, the cost reduction was even greater. This result illustrates the ability of BAI algorithms to avoid unpromising poses at an early stage and concentrate computational resources on promising poses instead.

2 Methods

2.1 Binding Pose Prediction and Best Arm Identification

In order to predict binding poses, we need to estimate and compare the binding free energies, $\Delta G_{\text{bind}}$, of each generated poses, as shown in Figure 1. An accurate value of $\Delta G_{\text{bind}}$ for a pose can be obtained by performing MD and MM-PBSA runs with multiple initial velocities. Uniform sampling, in which the same number of runs with different initial conditions are conducted, is the current standard method to obtain the best results (Sadiq et al., 2010; Genheden and Ryde, 2010; Mikulsikis et al., 2012, Berman et al., 2000), however it comes with enormous calculation costs (Figure 1 (A)).

We propose an effective pose prediction method that control runs based on best arm identification (BAI). The BAI problem was formulated by Bubeck et al. (Bubeck et al., 2009) in the field of machine learning. Figure 2 (A) shows a flowchart of the BAI algorithm in general settings. The purpose of the BAI algorithm is to optimize allocation of limited resources in order to find the best slot (arm). At a first step, a forecaster pulls each arm once and observes a sample drawn from the reward distribution of the arm (Initialization). He repeatedly selects an arm according to the "scores" of the arms (Selection) and gets a reward of the arm, until the budget runs out (Pull). The selection way of arms and the definition of "score" depends on exploration algorithms. Finally, he selects the seemingly best arm from the reward of each arm (Final Selection). Since he does not know the reward distributions, he needs to explore entire arms and exploits the seemingly most rewarding arms, avoiding the seemingly bad rewarding arms. A number of effective algorithms for the BAI problem have been proposed in some settings, and theoretical analysis has been also conducted (Bubeck et al., 2009; Audibert and Bubeck, 2010; Gabillon et al., 2012; Kaufmann and Kalyanakrishnan, 2013).
The problem of binding pose prediction can be regarded as a BAI problem as shown in Figure 2 (B) by considering an arm as a binding pose, pulling an arm as calculation of $-\Delta G_{bind}$ by MD and MM-PBSA, and a reward as an estimate $-\Delta G_{bind}$. BAI algorithms can reduce the total number of MD and MM-PBSA runs because they optimally control the number of runs for each pose to identify the binding pose.

2.1.1 BAI algorithms used in our experiment

We adopt four BAI algorithms for our experiments: Upper Confidence Bound algorithm with quantile factor parameter $p$ (UCB($p$)) (Bubeck et al., 2009), Upper Confidence Bound Exploration (UCB-E) (Audibert and Bubeck, 2010), Successive Rejects (Audibert and Bubeck, 2010), and Unified Gap-based Exploration (UGapE) (Gabillon et al., 2012). Their effectiveness have been shown both theoretically and experimentally, and implementation is relatively easy.

To explain the details of these algorithms, we introduce standard BAI notation (see overview in Figure 3): Let $A = \{1, 2, ..., K\}$ be the set of arms and $n$ the number of rounds (or budget). For $t = 1, 2, ..., n$, at round $t$, the forecaster chooses an arm $A_t$ in $A$ and observes a reward. A reward of an arm $i$ is sampled from the reward distribution $\nu_i$ which is the unknown parameter for the forecaster. For each arm $i$ and round $t$, we denote by $T_i(t)$ the number of times arm $i$ was pulled from rounds 1 to $n$, and by $X_{i,1}, X_{i,2}, ..., X_{i,n}$ the sequence of associated rewards. We define the empirical mean for arm $i$ after $s$ pulls $\hat{X}_{i,s} = \frac{1}{s} \sum_{t=1}^{s} X_{i,t}$.

After $n$ rounds, the forecaster selects an arm, denoted by $J_n$.

The UCB($p$) and UCB-E algorithms were proposed by Bubeck et al. (Bubeck et al., 2009) and Audibert et al. (Audibert and Bubeck, 2010) respectively, based off the UCB algorithm (Auer et al., 2002), which is practical and widely used for the multi-armed bandit (MAB) problem.

\[ \text{Algorithm 1 (UCB($p$)) algorithm.} \]

1. For $t = 1, K$, do
2. Pull arm $A_t$ and get reward $X_{i,t}$
3. end for
4. for $t = K + 1, n$, do
5. for $i = 1, K$, do
6. Calculate UCB($p$) score $S_i$ of arm $i$
7. end for
8. $J_n \leftarrow \arg \max_{i \in A} S_i$ and get reward $X_{J_n,t}$
9. end for
10. return $J_n$.

Algorithm 1 and 2 show the UCB($p$) and the UCB-E algorithms. The basic idea of these algorithms is to select the most promising arm based on each score function. These algorithms approximately calculate upper confident bounds (UCB) of arms according to some criteria and select an arm based on the UCBs. When using these methods, it is necessary to decide the total number of rounds (or budget) $n$, and the exploration parameters $p$ and $c$ beforehand. In general, when the value of an exploration parameter is large, it is extensively searched, and when the value is small, a small number of candidates are intensively searched. Users need to determine

---

1. In this study, we multiplied MM-PBSA estimates by $\lambda$ before applying BAI algorithms. This is because BAI algorithms output the best (maximum) arm, whereas the binding pose that has smallest binding energy should be explored.

2. The MAB problem is a problem to optimize allocation of limited resources based on existing knowledge in order to maximize the cumulative sum of rewards obtained by multiple slots (arms) (Robbins, 1952; Auer et al., 2002; Agrawal and Goyal, 2012). The BAI problem is a variant of the multi-armed bandit problem.
a proper value of the parameter in order to find the best arm effectively without missing it.

Audibert et al. also proposed the Successive Rejects (SR) algorithm (Algorithm 3) that is exploration-parameter free. The basic idea of SR is to reject unpromising arms and pull the remaining arms. SR has the same number of phases as the number of arms and discards one arm which is seemingly bad per phase. The number of poses calculated in each phase is determined by \( n_k \) in Algorithm 3. The sum of \( n_k \) for \( k = 1, 2, \ldots, K \) is less than or equal to \( n \). We also implemented UGapE algorithm (Algorithm 4) that outperformed existing algorithms including UCB-E in some settings (Gabillon et al., 2012). UGapE was proposed as an algorithm to find the best arm. We fix \( n = 1 \) in this paper. In the selection process of UGapE, we first find the "promising" arm \( \tilde{I}_t \) based on a confident interval \( \beta_k(t-1) \) and then select an arm among the arm \( \tilde{I}_t \) and the "best" one \( u_{\tilde{I}_t} \) in the arms except for \( \tilde{I}_t \). This algorithm also uses the exploration parameter \( \alpha \).

It is difficult to predict which algorithm is suitable for a given problem in advance, because the effectiveness of exploration depends on the distribution of rewards on the problem and exploration parameters. We experimentally verify which algorithm is appropriate for the binding-pose prediction problem.

### 2.1.2 Automatic estimation of exploration parameters

One practical issue with the algorithms covered in this paper, is the selection of exploration parameters that affect the effectiveness of exploration and are needed to be determined by users. We employ automatic adjustment methods for the free parameters \( \epsilon \) and \( \alpha \) in UCB-E and UGapE, as stated by Audibert et al. (Audibert and Bubeck, 2010) and Gabillon et al. (Gabillon et al., 2011).

For these two algorithms, the bounds of probability of error \( \epsilon_a \), i.e., the probability that the finally selected arm \( J_n \) is not the best one with round \( n \), have been proved as follows:

\[
e_{\alpha} \leq 2nK \exp\left(-\frac{\alpha - K}{18\epsilon_a}\right) \quad (1)
\]

for UCB-E and

\[
e_{\alpha} \leq 2nK \exp\left(-\frac{\alpha - K}{2n}\right) \quad (2)
\]

for UGapE where

\[
\epsilon = \frac{25}{36} \frac{n - K}{H_{\alpha}} \quad \text{and} \quad \alpha = \frac{n - K}{4H_{\alpha}} \quad (3)
\]

The variables \( H_{\epsilon} \) and \( H_{\alpha} \), called complexities of a problem, are calculated by using the the reward distribution \( v_1, v_2, \ldots, v_K \). However, such distribution is unknown in advance. As automatic parameter tuning methods (UCB-E auto and UGapE auto), we calculate suitable estimates \( \hat{H}_{\epsilon} \) and \( \hat{H}_{\alpha} \) of \( H_{\epsilon} \) and \( H_{\alpha} \) from observations at each round \( t \) and select an arm using the estimates. We show the details of these methods in Supplemental Method.

### 2.2 Preparation of Docking Poses

It is necessary to prepare the correct (stable) binding pose of a protein and a ligand in order to calculate the binding free energy between them. In this paper, we consider that a pose is correct if the root mean square deviation (RMSD) of the binding pose is less than or equal to 2.0Å from the experimentally-observed conformation.

We prepared 20 docking poses for each protein-ligand complex which consists of three proteins, CDK2, HSP20A, and FA10 and eight ligands (see Supplemental Method for details and Supplemental data). The basic information of the dataset is listed in Table 1. We show the RMSDs and thd docking scores of the prepared docking poses for each of all the compounds in Supplementary Table S1 and S2. In our dataset, one to three correct poses (\( \leq 2.0\)Å) are included in each pose set.

---

\(^3\) This criteria is widely used in binding pose prediction (Thompson et al., 2008; Cheng et al., 2009; Hou et al., 2011).
2.3 Molecular Dynamics (MD) Simulations of protein-ligand complexes

In molecular mechanics (MM) minimization and MD simulations, the Amber99SB-ILDN force field (Lindorff-Larsen et al., 2010) was used for proteins and the general AMBER force field (GAFF) (Wang et al., 2004) was used for ligands. The TIP3P water model (Jorgensen et al., 1983) was used for water molecules. Water molecules were placed around the complex model with an encompassing distance of 8Å, including roughly 13,000 water molecules. Charge-neutralization ions were added to neutralize the system. All MD simulations were carried out in periodic boundary conditions using the GROMACS 4 program (Hess et al., 2008) on the K-computer (RIKEN, Japan). Energy was first minimized for the initial configuration using the steepest descent method, then the Particle Mesh Ewald (PME) method (Darden et al., 1993) was used to calculate the long-range electrostatic interactions. Hydrogen atoms are constrained using the LINCS (Hess, 2008) algorithm. After minimization, the system was equilibrated for 100 ps under constant volume (NVT), and 300 ps under constant pressure and temperature (NPT) with positional restraints on protein heavy atoms and ligand atoms. Initial velocities were assigned from a Maxwell distribution at 298 K. Then a 1 ns production run was conducted under the NPT condition without positional restraints. In this procedure, the temperature was maintained at 298 K and the pressure was maintained at 1 atm.

20 sets of 1 ns production runs were performed with different initial velocities for each docking pose. The total simulation time was 3.2 µs (1 ns × 20 runs with different velocities × 20 poses × 8 complexes). All MD runs were carried out with time steps of 2 fs and snapshots for the MM-PBSA analysis were taken every 10 ps.

2.4 MM-PBSA Calculations

In the MM-PBSA method, the total free energy of a biomolecular system is expressed as follows (Kollman et al., 2000):

\[
G = E_{\text{MM}} + G_{\text{prot}} + G_{\text{SA}} - T S_{\text{Solvent}}
\]  

\[
E_{\text{MM}} = E_{\text{int}} + E_{\text{vdw}}
\]  

where \( E_{\text{MM}} \) is molecular mechanical (MM) energy term consisting of internal \( E_{\text{int}} \) (from bonds, angles, and dihedral angles), non-bonded electrostatic \( E_{\text{ele}} \) and van der Waals \( E_{\text{vdw}} \) energies. The polar and non-polar contributions to the solvation free energies are expressed as a continuum solvent Poisson-Boltzmann (PB) model and a solvent accessible surface area (SASA)-dependent nonpolar solvation term respectively. The last term in (4) is the absolute temperature \( T \) multiplied by the entropy \( S \).

The binding free energy \( \Delta G_{\text{bind}} \) is expressed as follows:

\[
\Delta G_{\text{bind}} = G_{\text{complex}} - (G_{\text{protein}} + G_{\text{ligand}}).
\]

To calculate the binding free energy, the typical procedure is to average \( \Delta G_{\text{bind}} \) in a set of conformation ensembles of the given complex structure taken from molecular dynamics (MD) simulations (Kollman et al., 2000; Hou et al., 2010). In this study, conformation ensembles in 1 ns MD production were used, as it was suggested that MM-PBSA calculation based on short (≈1 ns) MD simulations is appropriate for \( \Delta G_{\text{bind}} \) prediction and pose rescoring (Hou et al., 2011; Xu et al., 2013; Hou et al., 2010).

MM-PBSA calculations were carried out using the MMPBSA.py module (Miller III et al., 2012) in the Amber12 package (Case et al., 2012). In the calculation, we used the single-trajectory protocol, which is much faster than the original separate-trajectory protocol (Hou and Yu, 2007).

The snapshots for MM-PBSA calculations were taken every 10 ps over the last 500 ps period in each MD production run, resulting in a total of 50 snapshots per MD run. The \( \Delta G_{\text{bind}} \) estimate for each MD run was calculated by averaging the 50 MM-PBSA estimates for the snapshots. The nonpolar solvation free energy (\( G_{\text{SA}} \)) was determined by the solvent accessible surface area (SASA) using equation:

\[
G_{\text{SA}} = \gamma \text{SASA} + \beta
\]

where the surface tension \( \gamma \) and the offset \( \beta \) were set to the standard values of 0.00542 kcal mol\(^{-1}\) Å\(^{-2}\) and 0.92 kcal mol\(^{-1}\), respectively. It has been suggested many times that the change of conformation entropy \( T S_{\text{Solvent}} \) term can be omitted in pose prediction as it does not always improve the prediction results (Hou et al., 2010, 2011; Yang et al., 2011; Kumar et al., 2014), and therefore, in light of its high computational cost, this term was not considered here.

2.5 Calculated \( \Delta G_{\text{bind}} \) distributions of the prepared poses

For the generated conformation ensembles in the MD productions, we calculated \( \Delta G_{\text{bind}} \) according to the above MM-PBSA calculation procedure. For each pose, we calculated the average \( \Delta G_{\text{bind}} \) and standard deviation (SD) of the 20 \( \Delta G_{\text{bind}} \) values with MD and MM-PBSA calculations. In our dataset, the \( \Delta G_{\text{bind}} \) values were identified by the docking score.

3 Results and Discussion

To show the effectiveness of the proposed method, we performed a pose prediction experiment on our dataset. In the experiment, a pose prediction trial is a process to choose a pose from 20 pose candidates by calculating \( \Delta G_{\text{bind}} \) values with MD and MM-PBSA for each pose. In the pose prediction trials, the number of MD and MM-PBSA runs of each pose is controlled by different allocation algorithms: uniform sampling (baseline method), UCSF (UCSF E-antibody, SR, and UGExpE-antibody) (see Methods for more details). Note that a pose prediction trial may succeed, i.e., the pose chosen by the trial is a correct pose, or fail while using the same algorithm due to variations of \( \Delta G_{\text{bind}} \) depending on the initial velocity distribution in MD, as shown in Figure 4. Figure 5 illustrates the reductions of MD and MM-PBSA runs in a pose prediction trial for 3DQ-RRC and 2VCI-2GJ. The reduction results
Here, we first performed a pose prediction trial using uniform sampling ($k = 10$). Green bars show the number of runs per pose. The total number of runs was 200 ($10 \times 20$ poses). Blue, purple, red, and orange bars show the number of runs per pose using UGapE auto, UCB-E auto, SR and UCB($p$) ($p = 4$). Total runs of UGapE auto, UCB-E auto, and UCB($p$) were 50, and those of SR was 75. Black lines are the averaged binding free energies ($\Delta G_{\text{bind}}$) in Supplementary Table S1 and S2. As shown in Figure 5, the computational resources for the BAI algorithms are concentrated on small $\Delta G_{\text{bind}}$ poses. And the total numbers of runs using the BAI algorithms are reduced from 200 ($10 \times 20$ pose) runs using uniform sampling to 50 and 75 without reducing the number of runs for promising poses which have small $\Delta G_{\text{bind}}$ values. In all the trials in Figure 5, correct poses were successfully chosen. However, such trials do not always succeed due to the large fluctuations of $\Delta G_{\text{bind}}$ (see Figure 4).

To evaluate the performance of the BAI algorithms, we estimated the probability of correct pose prediction, i.e. the probability that an algorithm selects the correct pose among pose candidates under fixed parameters. For a given allocation algorithm and total number of runs, we repeated pose prediction trials and calculated the ratio of succeeded trials to total trials. Instead of actually performing the MD and MM-PBSA runs, we estimated the probability by sampling without replacement a binding free energy $\Delta G_{\text{bind}}$ from the energies calculated in Section 2.3, 2.4, and 2.5 and shown in Figure 4. We show the result of probability of correct pose prediction using uniform sampling (baseline) for all the complexes in Supplementary Figure S2. Figure 6 shows the probabilities of correct pose prediction using BAI algorithms (UGapE auto, UCB-E auto, SR, and UCB($p$) ($p = 4$)) increasing the total number of MD and MM-PBSA runs. For all the algorithms, the probabilities rose with increasing the number of total runs. For all the compounds, the total number of MD and MM-PBSA runs using BAI algorithms were reduced without sacrificing the probability of correct pose prediction. We summarized the total number of MD and MM-PBSA runs to reach the correct pose-prediction probability of 95% for exploration parameter free algorithms in Table 2. Table 3 shows the probabilities of correct pose prediction with the total number of runs fixed to 40, 60, and 100. From Table 2, the average numbers of runs using UGapE auto and UCB-E auto were approximately reduced by a factor of 3.5 compared to uniform sampling. The average probabilities of correct pose prediction using UGapE auto and UCB-E auto reached 90% only 40 total runs, although the probability of uniform sampling was 76% (Table 3).

In particular, when pose prediction is difficult due to a small difference in $\Delta G_{\text{bind}}$ between correct and incorrect poses, such as for 3DDQ-RRC and 2VCI-2GJ, computational cost using UGapE auto can be greatly reduced by a factor of 5.00 and 6.67, respectively (Figure 6 and Table 2). For example, the difference of $\Delta G_{\text{bind}}$ values between pose 1 (-41.9 kcal/mol, correct) and pose 2 (-41.6 kcal/mol, incorrect) of 2VCI-2GJ is 0.37 kcal/mol (see Supplementary Table S2) whereas the standard deviations of poses 1 and 2 were relatively large (2.23 and 1.00 kcal/mol). When the difference is small, a large number of runs for promising poses (e.g. poses 1 and 2 in 2VCI-2GJ) is needed to explore the difference and predict a correct pose, leading to unnecessary calculation of many unpromising poses (e.g. poses 3 to 20 in 2VCI-2GJ) under uniform sampling. On the other hand, BAI algorithms avoid such unpromising poses at an early stage and computational resource can be concentrated on only promising poses, leading to an even greater reduction in computational cost.

\footnote{When all the calculated binding free energies $\Delta G_{\text{bind}}$ of a pose were used up, pose selection was done from the remaining poses except for the pose.}
Machine Learning Accelerates MD-based Binding-Pose Prediction

Fig. 5. Reductions of MD and MM-PBSA runs per pose by BAI algorithms in a pose prediction trial. Green bars show the numbers of runs ($k = 10$) per pose by uniform sampling in a pose prediction trial. Blue, purple, red, and orange bars show the number of runs per pose using UGapE auto, UCB-E auto, SR and UCB($p = 4$). Black lines are the averaged binding free energies ($\Delta G_{20\text{vel}}$). The total numbers of runs by BAI algorithms are reduced from 200 (10 x 20 pose) by uniform sampling to 50 and 75 without reducing the number of runs for promising poses which have small $\Delta G_{20\text{vel}}$ values.

Fig. 6. The probabilities of correct pose prediction by the proposed methods and uniform sampling (baseline) at different numbers of MD and MM-PBSA runs for eight complexes. The total numbers of MD and MM-PBSA runs (computational cost) were reduced using the BAI algorithms (UGapE auto, UCB-E auto, UCB($p = 4$), and SR) without sacrificing accuracy compared to uniform sampling (green). The probabilities of UGapE auto (blue) and UCB-E auto (purple), whose exploration parameters were automatically adjusted, are higher than those of uniform sampling (green). Although UCB($p = 4$) showed almost the same high performance as UGapE auto and UCB-E auto under the exploration parameter $p = 4$, the result varies depending on the parameter as shown in Supplementary Figure S3. The result using SR (red) are a little worse than other BAI algorithms.

In order to assess the robustness of these algorithms, especially for the exploration parameters $p$, $c$, and $a$ for UCB($p$), UCB-E, and UGapE algorithms, we calculated the probabilities of correct pose prediction by changing these parameters. Supplementary Figures S3, S4, and S5 show the probability curves of correct pose predictions with different parameters. From these results, the optimal value of an exploration parameter for each complex may change depending on its $\Delta G_{\text{bind}}$ distributions shown in Figure 4. In Figures S3 and S4, the automatic parameter adjustment methods UCB-E auto and UGapE auto are not necessarily the best result for all complexes. However, these explore-parameter free algorithms are practical because it is difficult to decide the optimal parameter among different complexes and takes cost to search it by calculating the
Table 2. The number of MD and MM-PBSA runs to reach the correct pose prediction probability of 95% using exploration-parameter-free algorithms. The reduction effects obtained by dividing the numbers of runs of the BAI algorithms by ones of uniform sampling are shown in parentheses.

| Algorithm       | Total runs | 2R3J-SCJ | 109P-N20 | 1KE6-LS2 | 3DDQ-RRC | 2WYG-461 | 2YH4-G15 | 2VCI-2GJ | 3VHD-VHE | average |
|-----------------|------------|----------|----------|----------|----------|----------|----------|----------|----------|---------|
| Uniform sampling| 50         | 60       | 60       | 160      | 260      | 60       | 100      | 360      | 100      | 153     |
| UGapE auto      | 24 (2.50)  | 26 (2.31)| 75 (2.13)| 52 (5.00)| 29 (2.07)| 57 (2.81)| 54 (6.67)| 28 (3.57)| 43.1 (3.54) |
| UCB-E auto      | 24 (2.50)  | 26 (2.31)| 53 (3.02)| 60 (4.33)| 34 (1.76)| 51 (3.14)| 81 (4.44)| 28 (3.57)| 44.6 (3.42) |
| SR              | 27 (2.22)  | 29 (2.07)| 65 (2.46)| 92 (2.83)| 34 (1.76)| 68 (2.35)| 124 (2.90)| 38 (2.63)| 59.6 (2.56) |

Table 3. The probabilities (%) of correct pose prediction with the total number of runs fixed to 40, 60, and 100.

| Algorithm       | Total runs | 2R3J-SCJ | 109P-N20 | 1KE6-LS2 | 3DDQ-RRC | 2WYG-461 | 2YH4-G15 | 2VCI-2GJ | 3VHD-VHE | average |
|-----------------|------------|----------|----------|----------|----------|----------|----------|----------|----------|---------|
| Uniform sampling| 40         | 93.9     | 88.9     | 76.5     | 62.6     | 94.2     | 61.5     | 53.1     | 77.5     | 76.0    |
| UGapE auto      | 40         | 100      | 100      | 86.5     | 89.5     | 97.4     | 89.1     | 76.9     | 100      | 92.4    |
| UCB-E auto      | 40         | 99.9     | 99.3     | 86.2     | 79.9     | 96.6     | 81.9     | 63.6     | 96.9     | 88.0    |
| SR              | 40         | 96.7     | 95.5     | 81.6     | 69.9     | 96.6     | 71.5     | 57.7     | 90.1     | 82.7    |
| Uniform sampling| 60         | 100      | 100      | 98.0     | 98.8     | 98.6     | 95.7     | 100      | 100      | 97.7    |
| UGapE auto      | 60         | 100      | 100      | 96.9     | 95.0     | 99.6     | 97.5     | 66.5     | 100      | 94.4    |
| UCB-E auto      | 60         | 100      | 100      | 93.4     | 85.5     | 98.8     | 90.8     | 67.6     | 99.4     | 91.9    |
| SR              | 60         | 100      | 100      | 99.5     | 89.7     | 78.7     | 99.9     | 83.8     | 63.2     | 97.2    | 89.0    |
| Uniform sampling| 100        | 100      | 100      | 99.8     | 100.0    | 99.8     | 99.5     | 99.4     | 100      | 99.8    |
| UGapE auto      | 100        | 100      | 100      | 99.9     | 100.0    | 100.0    | 100.0    | 100.0    | 100      | 100.0   |
| UCB-E auto      | 100        | 100      | 100      | 99.9     | 100.0    | 100.0    | 100.0    | 100.0    | 100      | 100.0   |
| SR              | 100        | 100      | 100      | 99.9     | 96.7     | 100.0    | 100.0    | 99.9     | 84.1     | 100.0   |

4 Conclusions

In the present study, we have proposed an efficient binding pose prediction method based on BAI algorithms to estimate binding free energy between ligands and proteins. Our results on the test datasets, showed that the proposed method reduced the computational cost and improved accuracies compared to uniform sampling, in particular for small differences in binding free energies between correct and incorrect poses.

While we did confirm the effectiveness of BAI algorithms with the MM-PBSA method, we believe they could be just as effective at controlling multiarms in multiarmed bandit algorithms. We would also like to investigate methods using Bayesian optimization (Shahriari et al., 2016) in order to search for optimal poses from many candidates without limiting their number.

Acknowledgements

This research was supported by MEXT as “Priority Issue on Post-K computer” (Building Innovative Drug Discovery Infrastructure Through Functional Control of Biomolecular Systems). This research used computational resources on the K computer provided by the RIKEN Advanced Institute for Computational Science through the HPCI Research Project (Project ID: hp160213 and hp150272), Innovative Drug Discovery Infrastructure through Functional Control of Biomolecular Systems, and Subsidy for Kobe Biomedical Innovation Cluster promoting projects. We thank T. Shimada and K. Kaimazawa (Tecnin Pharma Ltd., Tokyo, Japan), and K. Hattori and K. Yasuo (Shionogi & Co., Ltd., Osaka, Japan) for providing the docked complex protein complexes toward HS90a and FA10 proteins using GLIDE (Schrödinger, LLC, NY).

Funding

This work was supported by Core Research for Evolutional Science and Technology (CREST), JST [grant number JPMJCR1502 to K. Tsuda, and JPMJCR1311 to Y.O. and H.I.].

References

Agrawal, S. and Goyal, N. (2012). Analysis of thompson sampling for the multi-armed bandit problem. In Conference on Learning Theory, pages 39.1–39.26.
Aügust, J., Medina, C., and Samuelsson, J.-E. (1994). A new method for predicting binding affinity in computer-aided drug design. Protein engineering, 7(3), 385–391.
Aubert, J.-Y. and Bubeck, S. (2010). Best Arm Identification in Multi-Armed Bandits. In Conference on Learning Theory, pages 13; Haifa, Israel.
Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2-3), 235–256.
Bairami, W. M. and Hansmann, U. H. (2013). The stability of cylindrin barrel amyloid oligomer models-a molecular dynamics study. Protein Structure, Function, and Bioinformatics, 81(9), 1542–1555.
Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. Nucleic acids research, 28(1), 235–242.
Bubbec, S., Mamon, R., and Stoltz, G. (2009). Pure exploration in multi-armed bandit problems. In International Conference on Algorithmic Learning Theory, pages 23–37. Springer.

Case, D., Darden, T., Cheatham III, T., Simmerling, C., Wang, J., Duke, R., Luo, R., Walker, R., Zhang, W., Merz, K., Roberts, B., Hayik, S., Roitberg, A., Seabra, G., Swails, J., Gohlke, P., Kolossváry, I., Wang, K., Parinami, F., Vanicek, J., Wolf, R., Liu, J., Xu, W., Borelli, S., Steibrecht, T., Gohlke, H., Cai, Q., Ye, X., Wang, J., Hashi, M.-J., Cui, G., Roe, D., Mathews, D., Seritz, M., Salomon-Ferrer, R., Sagu, C., Babun, V., Latchko, T., Gisarov, S., Kovalenko, A., and Kollman, P. (2012). Amber 12. University of California, San Francisco.

Cheng, T., Li, X., Li, Y., Liu, Z., and Wang, B. (2009). Comparative assessment of scoring functions on a diverse test set. Journal of chemical information and modeling, 4(9), 1079–1093.

Colalillo, F., Protao, R., Scapozza, L., Recanatini, M., and Cavalli, A. (2010). Single-molecule pulling simulations can discern active from inactivating enzyme inhibitors. Journal of the American Chemical Society, 132(21), 7361–7371.

Coulom, R. (2006). Efficient selection and backup operators in monte-carlo tree search. In International Conference on Computers and Games, pages 72–83. Springer.

Darden, T., York, D., and Pedersen, L. (1993). Particle mesh ewald: An nâ€…log (n) method for ewald sums in large systems. The Journal of chemical physics, 98(12), 10089–10092.

Fujitani, H., Tanida, Y., and Matsuura, A. (2009). Massively parallel computation of absolute binding free energy with well-equilibrated states. Physical Review E, 79(2), 021914.

Gabillon, V., Ghavamzadeh, M., Lazaric, A., and Bubeck, S. (2011). Multi-bandit best arm identification. In Advances in Neural Information Processing Systems, pages 2222–2230.

Gabillon, V., Ghamamzadeh, M., Lazaric, A., and Bubeck, S. (2012). Best arm identification: A unified approach to fixed budget and fixed confidence. In Advances in Neural Information Processing Systems, pages 5212–5220.

Ghedin, S. and Rude, U. (2010). How to obtain statistically converged mmpba results. Journal of computational chemistry, 31(4), 837–846.

Hess, B. (2008). Prince: A parallel linear constraint solver for molecular simulation. Journal of Chemical Theory and Computation, 4(1), 116–122.

Hess, B., Kutzner, C., Van Der Spoel, D., and Lindahl, E. (2008). Gromacs 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. Journal of chemical theory and computation, 4(3), 455–465.

Hou, T. and Yu, R. (2007). Molecular dynamics and free energy studies on the wild-type and double mutant hiv-1 protease complexes with amprine and two amprenavir-related inhibitors: mechanism for binding and drug resistance. Journal of medicinal chemistry, 50(6), 1177–1188.

Hou, T., Wang, J., Yu, L., and Wang, W. (2010). Assessing the performance of the mmpba and mmpb methods. 1. the accuracy of binding free energy calculations based on molecular dynamics simulations. Journal of chemical information and modeling, 50(1), 69–82.

Hou, T., Wang, J., Li, Y., and Wang, W. (2011). Assessing the performance of the molecular mechanics/poisson boltzmann surface area and molecular mechanics/generic Born surface area methods: the accuracy of ranking poses generated from docking. Journal of computational chemistry, 32(5), 866–877.

Jorgensen, W. L., Chandrasekhar, J., Madura, J. D., Impey, R. W., and Klein, M. L. (1983). Comparison of simple potential functions for simulating liquid water. The Journal of chemical physics, 79(2), 926–935.

Kaufmann, E. and Kalyanakrishnan, S. (2013). Information complexity in bandit problems. In Conference on Learning Theory, pages 228–251.

Kollman, P. A., Massova, I., Reyes, C., Kuhn, B., Huo, S., Chong, L., Lee, M., Lee, T., Duan, Y., and Wang, W. (2000). Calculating structures and free energies of complex molecules. Combining molecular mechanics and continuum models. Accounts of chemical research, 33(12), 889–897.

Kumar, R., Kumar, R., and Lynn, A. (2014). g_mmpba a gromacs tool for high-throughput mmpba calculations. Journal of chemical information and modeling, 54(7), 1951–1962.

Lavecchia, A. and De Giovanni, C. (2013). Virtual screening strategies in drug discovery: a critical review. Current medicinal chemistry, 20(23), 2839–2860.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In International conference on World Wide Web, pages 661–670. ACM.

Lindorff-Larsen, K., Piana, S., Palmo, K., Maragakis, P., Klepeis, J. L., Dore, R. O., and Shaw, D. E. (2010). Improved side-chain torsion potentials for the amber ff99sb protein force field. Proteins: Structure, Function, and Bioinformatics, 78(8), 1950–1958.

Lianto, E., Spyrow, K. V., Vasilisai, D., and Comina, Z. (2014). Structure-based virtual screening for drug discovery: principles, applications and recent advances. Current topics in medicinal chemistry, 14(16), 1923–1938.

Mikalakis, P., Gregersen, S., Ryberg, P., Sandberg, L., Olsen, L., and Ryle, U. (2012). Binding affinities in the mpba3x3x3 test and host-guest binding test estimated with the mmpba and mmpb methods. Journal of computer-aided molecular design, 26(1), 527–541.

Miller III, B. B., McGee Jr, T. D., Swails, J. M., Homeyen, N., Gohlke, H., and Roitberg, A. E. (2012). Mmpb mp: an efficient program for end-state free energy calculations. Journal of Chemical Theory and Computation, 8(9), 3314–3321.

Okimori, N., Fusitagi, N., Fuji, H., Suzuki, A., Morimoto, G., Yama, O., Ohno, Y., Narumi, T., and Tabi, M. (2009). High-performance drug discovery: computational screening by combining docking and molecular dynamics simulations. PLoS computational biology, 5(10), e1000528.

Onufriev, A., Bashford, D., and Case, D. A. (2000). Modification of the generalized born model suitable for macro-molecules. The Journal of Physical Biology, 3(4), 3712–3720.

Provost, A. E., Yin, S., Troppa, A., and Dohidhayon, N. V. (2012). Discrete molecular dynamics distinguishes native-like binding poses from decoys in difficult targets. Biophysical journal, 102(1), 144–151.

Robbins, H. (1952). Some aspects of the sequential design of experiments. Bull. Amer. Math. Soc., 58(5), 527–535.

Sadj, S. K., Wright, D. W., Kenway, O. A., and Coveny, P. V. (2010). Accurate ensemble molecular dynamics binding free energy ranking of multivalent-resistant hit-1 proteases. Journal of chemical information and modeling, 50(8), 896–905.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE, 104(1), 148–175.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G.,Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lament, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. Nature, 529(7587), 484–489.

Sivasivas, J., Cheatham III, T. E., Cieplak, P., Kollman, P. A., and Case, D. A. (1998). Continuum solvent studies of the stability of dna, rna, and phosphotriester-dna helicases. Journal of the American Chemical Society, 120(37), 9401–9409.

Thompson, D. C., Humbert, C., and Joseph-McCarthy, D. (2008). Investigation of mmpb rescoring of docking poses. Journal of chemical information and modeling, 48(5), 1081–1091.

Villar, S. S., Bowden, J., and Wason, J. (2015). Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. Statistical science: a review journal of the Institute of Mathematical Statistics, 30(2), 1992.

Walker, R., Zhang, W., Merz, K., Roitberg, A. E., and Case, D. A. (2004). Development and testing of a general amber force field. Journal of chemical theory and computation, 25(9), 1157–1174.

Xu, L., Sun, H., Li, Y., Wang, J., and Hou, T. (2013). Assessing the performance of mmpba and mmpb methods. 3. the impact of force fields and ligand charge models. The Journal of Physical Chemistry B, 117(28), 8406–8421.

Yang, T., Wu, J. C., Yan, C., Wang, Y., Luo, R., Santos, M. B., Dalby, K. N., and Ren, P. (2011). Virtual screening using molecular simulations. Proteins: Structure, Function, and Bioinformatics, 79(6), 1940–1951.