Use of molecular dynamics fingerprints (MDFPs) in SAMPL6 octanol–water log P blind challenge

Journal Article

Author(s):
Wang, Shuzhe; Riniker, Sereina

Publication date:
2020-04

Permanent link:
https://doi.org/10.3929/ethz-b-000387894

Originally published in:
Journal of Computer-Aided Molecular Design 34(4), https://doi.org/10.1007/s10822-019-00252-6

Funding acknowledgement:
178762 - Passive Membrane-Permeability Prediction for Peptides and Peptidomimetics Using Computational Methods (SNF)

Use of Molecular Dynamics Fingerprints (MDFPs) in SAMPL6 Octanol-Water log$P$ Blind Challenge

Shuzhe Wang,[a] Sereina Riniker[a]*

[a] Department of Chemistry and Applied Biosciences, ETH Zurich, Vladimir-Prelog-Weg 2, 8093 Zurich, Switzerland. *Email: sriniker@ethz.ch

Abstract

The *in silico* prediction of partition coefficients is an important task in computer-aided drug discovery. In particular the octanol-water partition coefficient is used as surrogate for lipophilicity. Various computational approaches have been proposed, ranging from simple group-contribution techniques based on the 2D topology of a molecule to rigorous methods based molecular dynamics (MD) or quantum chemistry. In order to balance accuracy and computational cost, we recently developed the MD fingerprints (MDFPs), where the information in MD simulations is encoded in a floating-point vector, which can be used as input for machine learning (ML). The MDFP-ML approach was shown to perform similarily to rigorous methods while being substantially more efficient. Here, we present the application of MDFP-ML for the prediction of octanol-water partition coefficients in the SAMPL6 blind challenge. The underlying computational pipeline is made freely available in form of the MDFPtools package.

1 Introduction

Lipophilicity is an important concept in medicinal chemistry as it influences the ADMET (absorption, distribution, metabolism, elimination and toxicology) properties of a compound [1]. The Lipophilicity of a molecule can be estimated by measuring its partition coefficient (log$P$) between an aqueous and organic solvent phase. In a pharmaceutical setting, the organic solvent is typically octanol, i.e. log$P_{\text{oct/wat}}$. In 1899, the very first quantitative structural-activity relationship (QSAR) study involved the correlation between lipophilicity and the anaesthesia ability of molecules [2]. Since then QSAR models have become a common tool in modern cheminformatics and log$P_{\text{oct/wat}}$ are routinely measured in drug discovery campaigns as a key quantity to profile potential drug candidates.

To assist in virtual screening of large compound libraries, a wide spectrum of computational approaches have been developed to estimate log$P_{\text{oct/wat}}$ *in silico*, ranging from empirical to rigorous physics-based approaches [3]. Empirical approaches typically aim to correlate log$P_{\text{oct/wat}}$ with topological information of the molecules. For example, group-contribution or substructure-based methods sub-divide molecules into smaller fragments or individual atoms, each contributing a certain amount to the total log$P_{\text{oct/wat}}$. These contributions are obtained by fitting to a large set of measured data. Alternatively, other 2D descriptors such as size or hydrogen-bonding capacity can also be used to fit the existing data (for an overview of methods see e.g. Ref. [3]). Physics-based approaches to estimate log$P_{\text{oct/wat}}$ include force-field methods that calculate transfer free energies or solvation free energies (e.g. with free-energy perturbation [4]), or quantum-mechanical (QM) methods such as COSMO-RS [5, 6]. COSMO-RS calculates screening charge densities for surface segments that make up a molecule. These density profiles are in turn used to obtain chemical potentials for molecules in different solvent environment. From chemical potentials log$P_{\text{oct/wat}}$ can be estimated. Empirical based methods are generally computationally much less demanding than physics-based methods but they rely heavily on the quantity and quality of measured training data used for fitting. This is less of an issue for log$P_{\text{oct/wat}}$ where a large amount of experimental data is available due to its importance in medicinal chemistry. However, much less data is

publicly available for other solvent pairs, which are relevant for example in environmental chemistry. For log$P$ of other solvent pairs, physics-based methods are therefore often more suitable.

Recently, we developed a novel approach to predict physicochemical properties such as solvation free energy and partition coefficients using a combination of molecular dynamics (MD) and machine learning (ML) [7]. The information in MD simulations is encoded into so-called MD fingerprints (MDFP) that can be used as input features to train ML models against experimental data. The MDFP-ML approach compared well to more rigorous *in silico* schemes such as FEP and COSMO-RS for the prediction of solvation free energies, while being computationally less expensive and easier to implement. From the solvation free energy of a molecule in two solvents, the partition coefficient can be calculated analytically,

$$\log P_{\mathsf{S2/S1}} = \frac{\Delta G^{\mathsf{S1}}_{\mathsf{solv}} - \Delta G^{\mathsf{S2}}_{\mathsf{solv}}}{RTln(10)}, \tag{1}$$

where R is the gas constant and T the absolute temperature. When comparing ML models trained against solvation free energies versus log$P$ (using a dataset of 630 molecules), the latter gave a slightly smaller root-mean-square error (RMSE) but a larger tilt in the slope between experimental and predicted values. We therefore concluded that training against solvation free energies is more robust. In addition, it is also more flexible because only an ML model per solvent is needed to estimate partition coefficients between different pairs of solvents. The MDFP-ML approach was further validated by predicting cyclohexane-water distribution coefficients (log$D_{\mathsf{cyc/wat}}$) from the SAMPL5 blind challenge [8] retrospectively [7]. Note that while log$P$ is related to the transfer free energy of the neural form of a molecule between solvents, log$D$ requires the consideration of all protonation states of a molecule at a given pH. Although log$P$ values instead of log$D$ were predicted with the MDFP-ML approach, the resulting RMSE was smaller than with the null model by the SAMPL5 organisers, which in turn had outperformed all submitted entries of the SAMPL5 challenge [8].

Here, we present the results from the application of the MDFP-ML approach in the SAMPL6 blind challenge[9] to predict octanol-water partition coefficients log$P_{\mathsf{oct/wat}}$, which was among the top 10 of the submitted entries. In addition, improvements from post-competition analysis are presented that increases the performance further. The MDFP-ML approach has been compiled into an open-source toolkit termed MDFPtools, which enables the setup of an automated workflow to predict log$P$ in different solvent combinations as well as other physicochemical properties of interest.

2 Methods

2.1 MDFP-ML Setup

The setup for the MDFP-ML approach is described in detail in Ref. [7]. In brief, given a dataset of molecules with a measured property P, a short MD simulation (5 ns) of each molecule in the dataset solvated in a water box is performed. From the simulations, different terms are extracted to obtain the MD descriptors. In this study, the terms were the same as in Ref. [7], namely the solute-solute and solute-solvent interaction energies (split up into the electrostatic and Lennard-Jones contributions), the radius of gyration, and the solvent-accessible surface area. For each term, the distribution of values obtained from the trajectories is encoded in the fingerprint by the mean, median, and standard devation. These features form the MDFP together with 2D topological counts (e.g. for heavy atoms, rotatable bonds, and different elements). The MDFPs are used as input features for one or more supervised ML models that are trained against the experimental data to predict P for an unseen molecule.

Changes to the original implementation in Ref. [7] are the following: (i) the AMBER-like parm@frosst force field [10] was used to parameterise systems instead of the GAFF [11] force field, (ii) only solution simulations (i.e. solute in water) were performed, no gas-phase simulations, (iii) OpenMM [12] was used as MD engine instead of GROMOS [13], and (iv) long-range electrostatic interactions were treated with particle-mesh Ewald (PME) [14] instead of reaction field (RF) [15]. The third change ensures consistency with the protocol used to parametrise force fields of the AMBER family. However, the use of PME does not allow a direct decomposition of the energy terms into solute-solute, solute-solvent and

solvent-solvent contributions as needed for the construction of the MDFPs. Therefore, the energies were recalculated with the RF expression using the trajectories in a post-processing step.

This workflow was assembled in the open-source MDFPtools Python package (see below). For the SAMPL6 blind challenge, MDFPtools was used to build, simulate and extract MDFPs for the solvated system, i.e. one solute molecule in a box of water.

2.2 MDFPtools Package

To facilitate the development and dissemination of the MDFP-ML approach, we built and maintain the MDFPtools package (https://github.com/rinikerlab/mdfptools/) that lets users build their MDFP pipelines with ease. It is written in Python and relies on fully open-source tools. MDFPtools consists of the following parts:

1. *Parameteriser* to build and parameterise systems
2. *Simulator* to perform simulations of systems
3. *Composer* to construct MDFPs with the properties extracted from simulations
4. *Predictor* to train ML models and make predictions

Parameteriser generates the 3D starting coordinates of the system to be simulated and provides all necessary force-field parameters. For SAMPL6, the *SolutionParameteriser* was used. As input, it takes the SMILES string of the solute. A 3D conformation of the solute is generated and solvated in a solvent box. Conformer generation is done by either the open-source cheminformatics toolkit RDKit (ETKDG [16]) or the commercial package OpenEye (OMEGA [17]). Solvation is carried out by PDBFixer[18], with water box padding of 1.25 nm on all sides of the solute as default. The TIP3P water model was used as solvent.

The OpenForceField toolkit [19] determines all force-field parameters for the molecular system using chemical environment matching via SMARTS strings. The current parameters were translated from parm@frosst99. The OpenForceField toolkit can be run using either RDKit or OEChem from OpenEye as backend. The only parameters calculated on the fly are the partial charges of the solute. When using OpenEye as backend, oequacpac derives the semi-empirical AM1-BCC [20, 21] charges. When using RDKit as backend, Antechamber from ambermini (based on AmberTools16 [22]) is called by OpenForceField. Alternatively, we introduced the option to assign ML-predicted partial charges using the models from Ref. [23]. The ML-predicted charge assignment is implemented in our mlddec package (https://github.com/rinikerlab/mlddec) and it is integrated in the MDFPTools parameteriser. Finally, a fully parameterised system is written out as a compact ParmEd [24] object. For SAMPL6, OpenEye was used as backend for the parameterisation of the solutes.

Simulator carries out the simulation of a parameterised system inputted as ParmEd object. For SAMPL6, the *SolutionSimulator* was used. The simulation parameters detailed in Section 2.1 are set as default (simulation length of 5 ns, Andersen thermostat at 298 K, Monte Carlo barostat at 1 atm, PME with a long-range cut-off of 1.0 nm). The user can modify them if needed. At the backend of the simulator is the open-source OpenMM package, which is optimised for single-thread GPU platforms. We intend to add a python wrapper for the GROMACS MD engine [25] in the future.

Composer takes as input the parmeterised ParmEd system and the simulated trajectory (by default written in the binary hdf5 format). For SAMPL6, the *SolutionComposer* was used. The composer works by first calling the relevant extractor class in MDFPtools to obtain the quantities of interest from every frame of the simulated trajectory. This is done using features inside OpenMM, ParmEd and MDTraj [26]. Next, the composer calculates the statistical moments from the distributions of each quantity to be stored in the MD feature vector. Lastly, it combines them with 2D topological features calculated via RDKit.

Predictor includes some basic supervised ML methods from Scikit-Learn [27] together with their corresponding hyperparameters. As the best hyperparameters for a given learning task strongly depend on the amount and quality of data at hand, we recommend the users to adjust the hyperparameters for

their dataset. If desired, the predictions of different ML models can be combined using the meta-learner approach from the MLEns package [28].

For SAMPL6, the following versions of software packages were used: OpenForceField 0.0.4, OpenMM 7.1.1, Scikit-Learn 0.19.1, MLEns 0.2.3, and MDFPtools 0.0.1. After the SAMPL6 competition, additional packages xgboost 1.0.0 and PyTorch 1.10 were used to experiment with different learning models.

2.3 Meta-learner

Meta-learner is a form of ensemble modelling which exploits consensus amongst ML models that have different sources of errors. Here, a diverse set of supervised ML models (also known as base predictors) were first trained independently, each with a cross-validation on the entire training dataset. This serves as the first layer of training. Next, the independent predictions from all base predictors are inputted into a second (meta) layer of ML model, which determines how to optimally combine the individual predictions. The entire training set was used at once for training the meta layer. A grid-search approach was used to determine the hyperparameters of the base predictors. For each predictor, 100 picks were taken from the pre-defined hyperparameter ranges, and for each hyperparameter combination a five-fold cross-validation was performed. This yielded 500 models per base predictor.

2.4 Datasets

For SAMPL6, two different datasets were used. The first dataset consists of 670 molecules with experimental $\Delta G^{\text{wat}}_{\text{solv}}$ values from the FreeSolv [29, 30] database, and 480 molecules with experimental $\Delta G^{\text{oct}}_{\text{solv}}$ from the Minnesota [31] database. The second dataset consists of 15'784 molecules from OChem [32] with experimental $\log P_{\text{oct/wat}}$ values. For the latter a subset was used as well, which contains the 4'304 molecules, for which the MD simulations on the cluster finished first.

For OChem, the dataset obtained contain some molecules with multiple measurements. Only those molecules with a single measured value or multiple values that are the same were used. The list of experimental $\log P_{\text{oct/wat}}$ values can be obtained from ochem.eu/properties/show.do after free registration. The SMILES codes of the $\log P_{\text{oct/wat}}$ dataset used in this study are provided in the Supporting Information.

2.5 Performance Assessment

The performance of the ML models was assessed by calculating the root-mean-square error (RMSE), mean absolute error (MAE), and line of best fit for a held-out test set or the cross-validation. The line of best fit is obtained by minimising the sum of squared distances between all points in the dataset and the line (i.e. linear regression).

3 Results and Discussion

3.1 Submissions to the SAMPL6 Blind Challenge

We submitted four entries to the SAMPL6 blind challenge, summarised in Table 1. Entry 1 follows closely the approach in Ref. [7], where ML models for $\Delta G^{\text{wat}}_{\text{solv}}$ and $\Delta G^{\text{oct}}_{\text{solv}}$ were trained separately and combined analytically using Eq. (1) to give $\log P_{\text{oct/wat}}$. Compared to the three other submissions, where models are trained directly on $\log P_{\text{oct/wat}}$ data, this approach appears to be more robust and offers more flexibility as one can estimate $\log P$ between arbitrary pairs of solvents. However, the amount of experimental solvation free energies that is available is much smaller than for $\log P_{\text{oct/wat}}$, which will affect the performance of entry 1.

Table 1: Four submissions with the MDFP-ML approach to the SAMPL6 blind challenge for $\log P_{\text{oct/wat}}$ prediction and their associated submission codes. The submission files can be found at https://github.com/MobleyLab/SAMPL6/tree/master/physical_properties/logP/predictions/submission_files.

	Entry 1	Entry 2	Entry3	Entry4
Submission code	5mahv	w6jta	0a7a8	gnxuu
Model keyword	ΔG_{solv} model	$\log P$ subset	$\log P$ meta-learner	$\log P$ all data
#Data Points	670 $\Delta G^{\text{wat}}_{\text{solv}}$ values, 480 $\Delta G^{\text{oct}}_{\text{solv}}$ values	4'304 $\log P_{\text{oct/wat}}$ values	4'304 $\log P_{\text{oct/wat}}$ values	15'784 $\log P_{\text{oct/wat}}$ values
Model description	Two ensemble models (same as in entry 2) were trained, one against $\Delta G^{\text{wat}}_{\text{solv}}$ and the other against $\Delta G^{\text{oct}}_{\text{solv}}$. Prediction: ΔG_{solv} values in the two solvents were obtained for each SAMPL6 molecule and the $\log P_{\text{oct/wat}}$ value was calculated using Eq. (1).	Ensemble of LASSO [33] and gradient tree boosting (GTR) [34] models. GTR model: n_estimators = 100, max_depth = 3. The outcomes from the two ML models were averaged to give the final prediction.	A set of different ML models were trained independently, and the predictions from these models were used as input for an additional layer of learning model (i.e. meta-learner) that optimises the combination of individual predictions.	See entry 2

Using the $\log P_{\text{oct/wat}}$ data from OChem, the ML models in entries 2-4 could be trained on a much larger dataset. Entries 2 and 3 were trained using the subset of 4'304 molecules, for which the MD simulations on the cluster finished first. We consider this to be random subset of the complete dataset. Entry 2 uses the same ML ensemble approach as in entry 1, only the training set and target property differs. For entry 3, a larger set of ML methods was employed together a meta-learner, which exploits consensus among diverse ML models. However, the meta-learner as implemented in MLEns requires a large amount of memory and scales poorly with the number of data points due to the grid-search approach for determining the hyperparameters of the base predictors. The current MLEns implementation retains all model parameters during training of the first layer and only filters for the best base predictors after completion of the training. For the $\log P_{\text{oct/wat}}$ subset with 4'304 data points, hundreds gigabytes of memory and hours of runtime were required.

The ML models in entry 4 were trained on the complete set of 15'784 molecules with experimental $\log P_{\text{oct/wat}}$ values. The ML approach is the same as in entry 1 and 2. A meta-learner was not attempted with the larger dataset because of the associated computational cost. Figure 1 shows the deviation between predicted and experimental $\log P_{\text{oct/wat}}$ values for the four submissions using different validation techniques. Information on the line of best fit, MAE and RMSE are listed in Table 2. One should keep in mind the data difference in the four validation campaigns, but the general trend suggests entry 3 to perform the best, giving the lowest MAE and RMSE value as well as a line of best fit close to $y = x$. The slope of the best-fit line for entry 1 is closer to one, while the slopes for entries 2 and 4 deviate more, which was already observed in Ref. [7] for the models directly fitted against $\log P$ data compared to ΔG_{solv}.

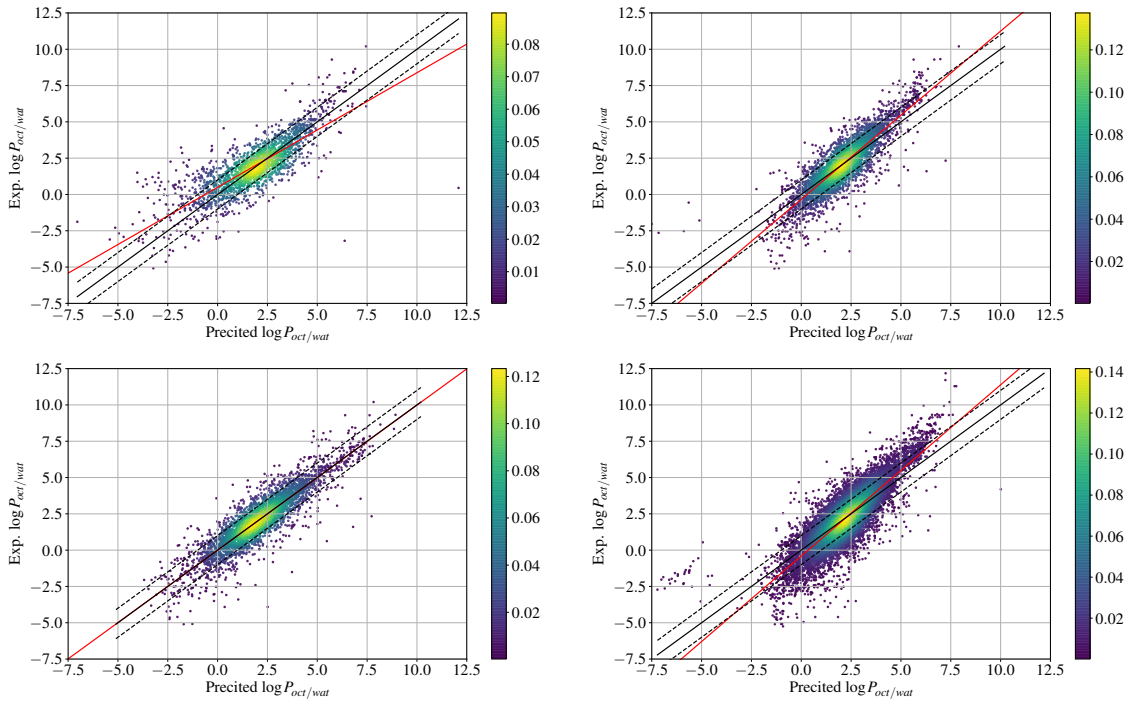

Figure 1: Experimental versus predicted $\log P_{\text{oct/wat}}$ values for a validation set (entry 1) or from cross-validation (entries 2-4). The validation set for entry 1 (top left) consists of 1'999 compounds unseen in the training set. For entries 2 and 4, a 100-fold cross-validation was performed, for entry 3 a 5-fold cross-validation. The colour indicates the density of points at a region, the lighter the colour the higher the density. The solid black line shows $y = x$, while the dashed lines indicate one log unit offsets on either side. The red line shows the line of best fit.

Table 2: RMSE, MAE and line of best fit (slope, intercept) for a validation set (entry 1) or from cross-validation (entries 2-4). The validation set for entry 1 consists of 1'999 compounds unseen in the training set. For entries 2 and 4, a 100-fold cross-validation was performed, for entry 3 a 5-fold cross-validation.

	Entry 1	Entry 2	Entry 3	Entry 4
RMSE	1.16	0.95	0.81	1.02
MAE	0.83	0.71	0.58	0.77
Slope	0.789	1.158	0.999	1.179
Intercept	0.50	-0.33	-0.001	-0.40
R^2	0.653	0.752	0.805	0.734

The generation of MDFPs is certainly computationally more expensive compared to simpler topological descriptors. However, as shown in Ref. [7] the same MDFPs can be used to train ML models to predict different solvation free energies or partition coefficients. Furthermore, the MDFPs present an orthogonal description of molecules compared to topological fingerprints, thus the combination of the two can be beneficial. In this study, a separate simulation was performed for each of the 15'784 molecules. Depending on the size of the molecule and the corresponding amount of solvent, a simulation with OpenMM took 1 - 5 hours on 4 CPU cores. Note that OpenMM is mainly optimised for GPU platforms, thus the same simulations would take about 20 minutes on a high-end NVIDIA GPU.

3.2 Post-Competition Analysis

The results for the 11 molecules from the SAMPL6 blind challenge are given in Table 3 for the four entries together with the experimental values revealed after the end of the competition. The standard

deviation was determined by repeated training of the ML models with different random number seeds and calculating the average over the predictions. For entries 1, 2 and 4, 100 repetitions were performed (note that similar results were obtained with ten repetitions). Due to the computational cost of the meta-learner, the standard deviation of entry 3 was calculated from only six repetitions. Figure 2 shows the deviations of the four entries from the true experimental values for each SAMPL6 molecule.

As could be expected from the validation results, entry 3 performed best, ranking 10th in terms of MAE ($= 0.43$ log units). Although there is the danger of overfitting with a meta-learner and its computational cost is high, it clearly improved the performance. Entry 1 on the other hand was the least performant with a MAE $= 0.62$, ranking 25th out of total 93 submissions. This was also to be expected considering the much smaller training set compared to the other models. It can be seen in Figure 2 that the per-molecule error is somewhat "bipolar" for entry 1, i.e. it performs either very well or very badly. In fact, entry 1 is the top performer (amongst our four entries) for five of the 11 molecules but also the worst performer for another five molecules. This can also be seen in Figure 3, where the spread of the absolute error for entry 1 is the biggest. We hypothesised that the SAMPL6 molecules with very low prediction errors are relatively "close" to some molecules in the ΔG_{solv} training sets. However, this is not supported by neither topological fingerprint similarity or MDFP similarity between SAMPL6 molecules and the compounds in the training sets. Interestingly, while the molecules in the ΔG_{solv} training set are generally smaller than the SAMPL6 molecules, this is no longer the case in the $\log P_{\text{oct/wat}}$ training set with 15'784 molecules. It would therefore be interesting to see the performance of the ΔG_{solv} based models when more training data is available.

For all four entries, there is no correlation between the standard error of a prediction (Table 3) and the deviation from the experimental value (Figure 2). Recently, studies in neural networks for the prediction of QM energies found that the standard deviation obtained by training several models on different portions of the entire training set correlated with the prediction confidence [35]. However, such a correlation could not be observed in this study. This suggests that standard errors obtained in this manner cannot be used generally as an estimate for applicability domain.

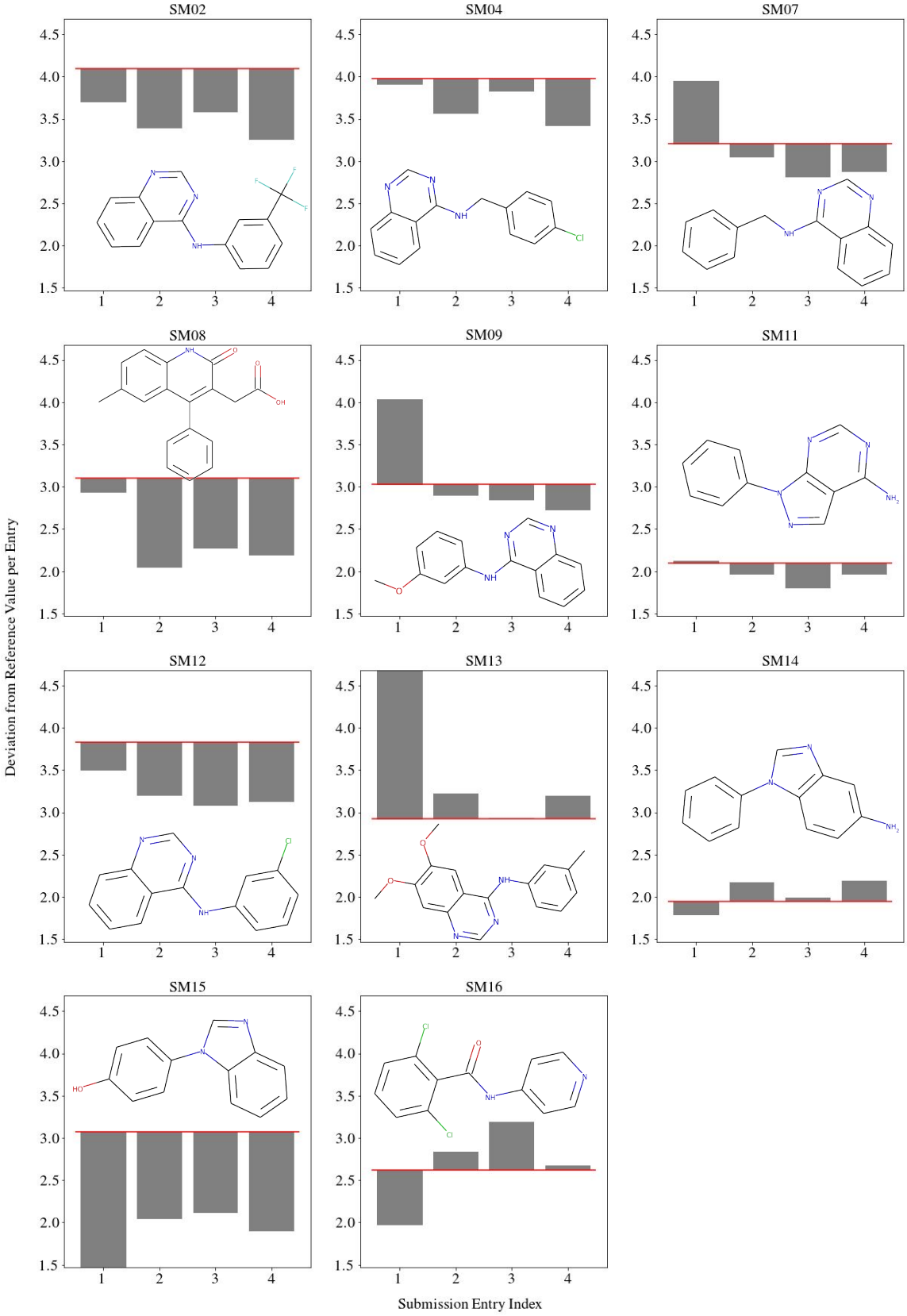

Figure 2: Prediction error from the experimental value for the 11 SAMPL6 molecules and our four entries. The red horizontal line corresponds to the experimental $\log P_{\text{oct/wat}}$ value.

Table 3: Predicted and experimental $\log P_{\text{oct/wat}}$ value and standard error for the 11 SAMPL6 molecules. The mean and standard deviation for entries 1, 2 and 4 were obtained from 100 repeated trainings of the ML models with different random number seeds. For entry 3, only six repetitions were performed due to the computational cost. The experimental measurements were obtained from potentiometric method with at least three replicates. The mean and the standard deviation averaged over the replicas are reported.

Molecule Index	Entry 1	Entry 2	Entry 3	Entry 4	Exp.
SM02	3.70 ± 0.25	3.39 ± 0.07	3.58 ± 0.15	3.25 ± 0.06	4.09 ± 0.03
SM04	3.91 ± 0.20	3.56 ± 0.06	3.82 ± 0.16	3.41 ± 0.05	3.98 ± 0.03
SM07	3.95 ± 0.18	3.04 ± 0.05	2.81 ± 0.18	2.87 ± 0.04	3.21 ± 0.04
SM08	2.94 ± 0.20	2.04 ± 0.05	2.27 ± 0.15	2.19 ± 0.05	3.10 ± 0.03
SM09	4.03 ± 0.21	2.89 ± 0.05	2.84 ± 0.12	2.72 ± 0.04	3.03 ± 0.07
SM11	2.12 ± 0.19	1.96 ± 0.05	1.80 ± 0.15	1.96 ± 0.05	2.10 ± 0.04
SM12	3.50 ± 0.18	3.20 ± 0.06	3.08 ± 0.23	3.12 ± 0.04	3.83 ± 0.03
SM13	4.68 ± 0.24	3.22 ± 0.05	2.93 ± 0.37	3.20 ± 0.04	2.92 ± 0.04
SM14	1.79 ± 0.17	2.17 ± 0.07	1.99 ± 0.08	2.19 ± 0.05	1.95 ± 0.03
SM15	1.47 ± 0.16	2.04 ± 0.05	2.11 ± 0.12	1.90 ± 0.04	3.07 ± 0.03
SM16	1.98 ± 0.16	2.84 ± 0.06	3.19 ± 0.07	2.67 ± 0.04	2.62 ± 0.01
MAE	0.62	0.46	0.43	0.51	
RMSE	0.85	0.56	0.53	0.61	

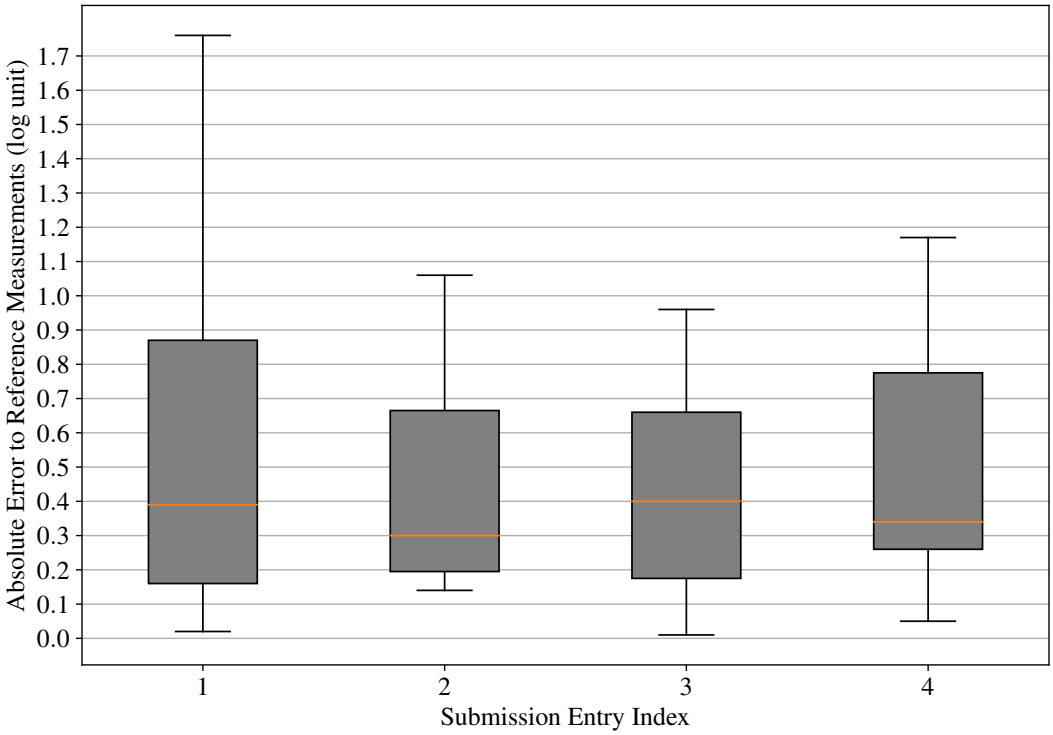

Figure 3: Distribution of absolute errors shown as box plots for the four submission entries for the 11 SAMPL6 molecules. The red line corresponds to the median error.

3.3 Post-Competition Improvements

Can the log$P$ based predictors be improved further? For this, we decided to focus on entries 2 and 4 due to considerations of computational cost, even though entry 3 performed best in terms of both MAE and RMSE. Interestingly, entry 2 with a training set of 4'304 molecules performed slightly better on the SAMPL6 molecules than entry 4 with a training set of 15'784 molecules. A reason for this could be that the ML model hyperparameters were not adjusted for the increased size of the training sets. Another factor is the ML method. The LASSO model was found to be no longer contributing to the ensemble model, i.e. the GTR model alone was equally performant. Could the performance be improved with a neural network? Based on these considerations three different post-competitions models were investigated using the dataset with 15'784 molecules:

- Entry 5: GTR model with 10'000 estimators and *max estimator depth* = 50.
- Entry 6: Fully-connected neural net (FCNN) with three hidden layers size of [500, 100, 30] (no fine-tuning of hyperparameters).
- Entry 7: GTR model with 2'500 estimators and *max estimator depth* = 2.

Note that for the post-competition models the xgboost package[36] was used instead of GTR from Scikit-Learn to improve efficiency. To mimic the competition process, the model performance was assessed first in terms of overall RMSE and MAE from a 100-fold cross-validation on the training set.

Increasing both hyperparameters was found to reduce the model uncertainty by more than 0.2 log units (top left panel in Figure 4). Using a FCNN model instead of GTR further improved the performance (top right panel in Fig. 4). In entry 7, the hyperparameters of the GTR model were optimised such that the best-fit line is closest to $y = x$. This resulted in a GTR model with 2'500 estimators and a *max estimator depth* of two (bottom panel in Figure 4). The RMSE, MAE and best-fit line of the three post-competition models from a cross-validation are listed in Table 4.

Table 4: RMSE, MAE and line of best fit (slope, intercept) from 100-fold cross-validation for the three post-competition models. Entry 5 is a GTR model with 10'000 estimators and *max estimator depth* = 50. Entry 6 is a FCNN model with three hidden layers size of [500, 100, 30] (hyperparameters not fine-tuned). Entry 7 is a GTR model with 2'500 estimators and *max estimator depth* = 2.

	Entry 5	Entry 6	Entry 7
RMSE	0.77	0.54	0.82
MAE	0.53	0.33	0.60
Slope	1.032	1.007	1.003
Intercept	-0.05	-0.03	-0.01
R^2	0.840	0.919	0.817

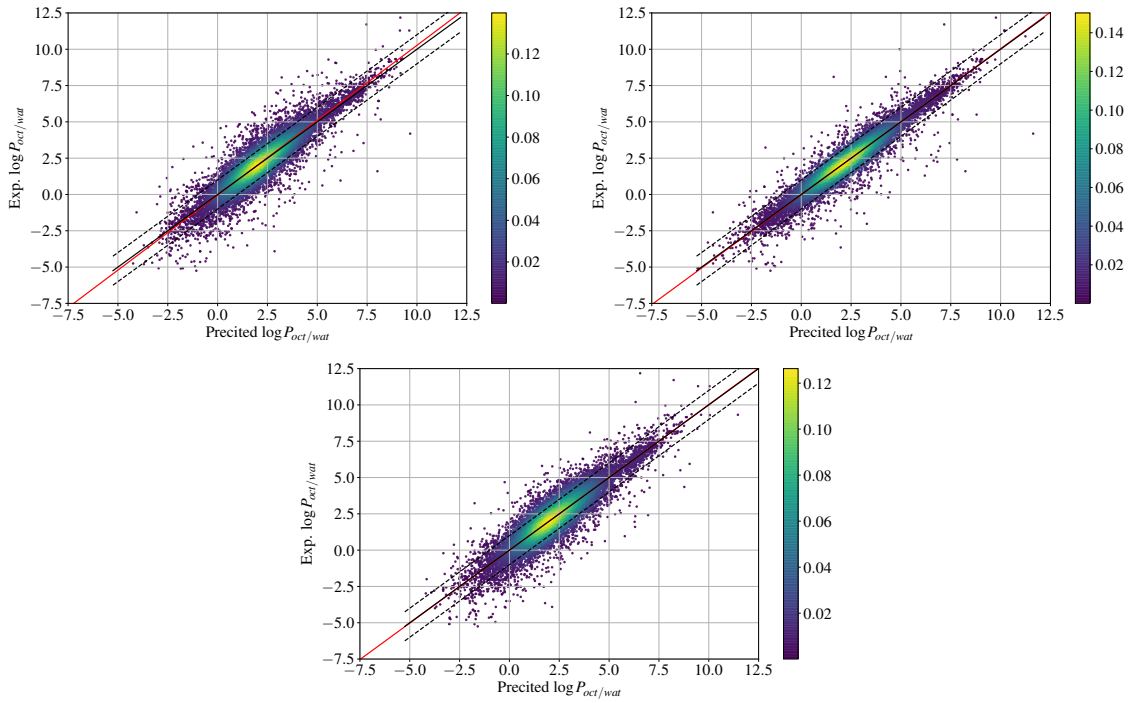

Figure 4: Experimental versus predicted $\log P_{\text{oct/wat}}$ values from 100-fold cross-validation for the three post-competition models. Entry 5 (top left) is a GTR model with 10'000 estimators and *max estimator depth* $= 50$. Entry 6 (top right) is a FCNN model with three hidden layers size of [500, 100, 30] (hyperparameters not fine-tuned). Entry 7 (bottom) is a GTR model with 2'500 estimators and *max estimator depth* $= 2$. The colour indicates the density of points at a region, the lighter the colour the higher the density. The solid black line shows $y = x$, while the dashed lines indicate one log unit offsets on either side. The red line shows the line of best fit.

In a second step, the model performances were evaluated on the SAMPL6 molecules (Table 5). Interestingly, entries 5 and 6 perform substantially worse than entries 2 and 4. For entry 6, this is likely due to overfitting of the FCNN model to the training set (the number of parameters in the FCNN model is high compared to the size of the MDFPs and the size of the training set), which is reflected by the high standard deviation of its predictions (Table 5). Entry 7 on the other hand gives with a MAE $= 0.38$ on the SAMPL6 molecules, which would have corresponded to rank 5 in the competition.

If the same ML model as in entry 7 was trained on the 4'304 molecules subset instead of the full dataset, the performance did not differ significantly in validation and test (not shown). Although the 4'304 molecules are a random subset, the chemical diversity is of course smaller than for the full $\log P_{\text{oct/wat}}$ dataset. We thus experimented with randomly picking subsets of 5'000 molecules from the full dataset. However, while the cross-validation error remained nearly constant for the different subsets, the performance on the SAMPL6 molecules changed wildly. One possible explanation for this observation is the source of the OChem dataset, which was collected and curated from a large corpus of literature. Each of the studies had likely a (slightly) different experimental protocol with different associated errors. The combined dataset is therefore rather heterogeneous. Just by chance, it could be that the first subset with 4'304 molecules has a sizeable fraction of measurements that were done with similar experimental protocols as the SAMPL6 molecules. An effective way to handle the heterogeneity in the data would be to stratify the entire set into subsets clustered by experimental protocol and train separate ML models on them. However, there is no automated approach to deduce the similarity between experimental protocols in the literature, i.e. this would have to be done manually. As such a stratification is not available, we decided to model the noise as random by generating an ensemble of 100 GTR models (entry 8), each trained on a random subset of 5'000 data points from the dataset (the cross-validation results from each draw are close to those for entry 7, with small fluctuations due to the random sampling). For the prediction of an unseen molecule, the outcomes of the 100 models are then averaged. This

approach resulted in a MAE = 0.35 and RMSE = 0.44 on the SAMPL6 molecules, which would have corresponded to rank 4 in the competition. Compared to entries 1–4, the absolute error distribution for the SAMPL6 molecules is shifted substantially to lower values, with SM15 as outlier. This molecule is consistently predicted poorly by all our models. In addition, all of the top 20 entries to the SAMPL6 challenge underestimate the $\log P_{\text{oct/wat}}$ value of SM15. SM15 is the only molecule where the median error of the top 20 entries exceeds 0.5 log units. It is currently unclear where this deviation comes from. Figure 5 shows the predicted versus experimental $\log P_{\text{oct/wat}}$ values for the 11 SAMPL6 molecules for entries 1–4 and 8.

Table 5: Predicted and experimental $\log P_{\text{oct/wat}}$ value and standard error for the 11 SAMPL6 molecules. The mean and standard deviation for all entries were obtained from 100 repeated trainings of the ML models with different random number seeds. The experimental measurements were obtained from potentiometric method with at least three replicates. The mean and the standard deviation averaged over the replicas are reported.

Molecule Index	Entry 5	Entry 6	Entry 7	Entry 8	Exp.
SM02	3.36 ± 0.21	3.11 ± 0.58	3.47 ± 0.15	3.55 ± 0.28	4.09 ± 0.03
SM04	3.43 ± 0.28	3.70 ± 0.47	3.77 ± 0.16	3.67 ± 0.29	3.98 ± 0.03
SM07	2.51 ± 0.18	3.00 ± 0.46	3.16 ± 0.18	3.16 ± 0.30	3.21 ± 0.04
SM08	2.57 ± 0.32	3.23 ± 0.44	2.86 ± 0.15	2.76 ± 0.33	3.10 ± 0.03
SM09	2.43 ± 0.20	2.82 ± 0.43	2.59 ± 0.12	2.79 ± 0.31	3.03 ± 0.07
SM11	2.20 ± 0.17	1.73 ± 0.38	1.91 ± 0.15	2.02 ± 0.30	2.10 ± 0.04
SM12	3.03 ± 0.32	3.33 ± 0.47	3.44 ± 0.23	3.30 ± 0.34	3.83 ± 0.03
SM13	2.75 ± 0.19	2.96 ± 0.44	2.78 ± 0.37	2.86 ± 0.31	2.92 ± 0.04
SM14	2.05 ± 0.21	2.28 ± 0.38	2.09 ± 0.08	2.23 ± 0.22	1.95 ± 0.03
SM15	1.89 ± 0.21	2.13 ± 0.34	1.80 ± 0.12	2.01 ± 0.22	3.07 ± 0.03
SM16	3.24 ± 0.24	3.58 ± 0.39	2.81 ± 0.07	2.92 ± 0.33	2.62 ± 0.01
MAE	0.59	0.73	0.38	0.35	
RMSE	0.69	0.91	0.49	0.44	

4 Conclusion

The MDFP-ML approach was applied to predict $\log P_{\text{oct/wat}}$ values of 11 unseen molecules in the SAMPL6 blind challenge. The ML models can thereby be trained either against the underlying solvation free energies or the partition coefficients directly. In this competition, we found that the models trained against $\log P_{\text{oct/wat}}$ data performed better, mainly due to the much higher amount of experimental data publicly available. Models trained against ΔG_{solv} have the advantage to be more robust and flexible, which is of advantage for partition coefficients between other pairs of solvents, for which less data is available (potentially in SAMPL7).

Of the four entries submitted to the competition, entry 3 performed best on the SAMPL6 molecules, resulting in rank 10 in terms of MAE. In this approach, an additional meta-learner was employed, which exploits the different sources of errors in different ML methods. However, it requires a large amount of memory and scales poorly with the number of data points. Improvements after the competition yielded entry 8, which considers the heterogeneity in the training data due to different experimental protocols as random noise. The performance of entry 8 on the 11 SAMPL6 molecules would have resulted in rank 4 in the competition (in terms of MAE and RMSE).

The combination of MD simulations and ML via the MDFPs is a promising approach, as these fingerprints describe the molecules in a different (orthogonal) way to classical 2D topological descriptors. Through the open-source MDFPtools package that enables automated workflows, we make the MDFP-ML approach easily available to the community.

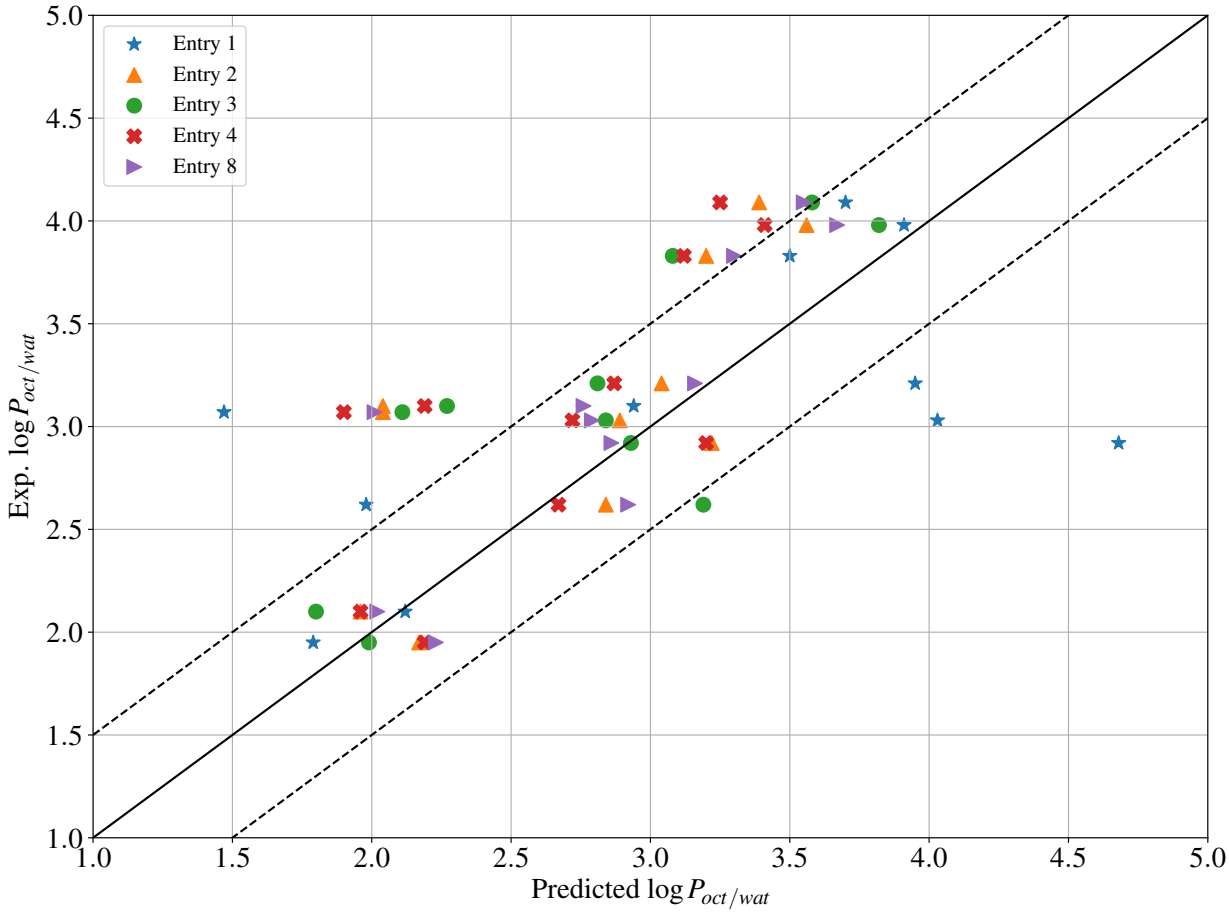

Figure 5: Experimental versus predicted $\log P_{\text{oct/wat}}$ values for the 11 SAMPL6 molecules for entries 1 (blue), 2 (orange), 3 (green), 4 (red), and 8 (purple). Solid black line shows $y = x$, with dashed lines indicating 0.5 log unit offsets on either side.

Acknowledgments

The authors gratefully acknowledge financial support by the Swiss National Science Foundation (Grant Number 200021-178762) and by ETH Zurich (ETH-34 17-2). They further acknowledge SAMPL NIH grant 1R01GM124270-01A1 for support of the experimental work.

References

[1] M. J. Waring, *Expert Opin. Drug Discov.* **2016**, *5*, 235–248.

[2] H. Meyer, *Naunyn-Schmiedeberg's Archives of Pharmacology* **1899**, *42*, 109–118.

[3] R. Mannhold, G. I. Poda, C. Ostermann, I. V. Tetko, *J. Pharma. Sci.* **2009**, *98*, 861–893.

[4] R. W. Zwanzig, *J. Chem. Phys.* **1954**, *22*, 1420–1426.

[5] A. Klamt, F. Eckert, W. Arlt, *Annu. Rev. Chem. Biomol. Eng.* **2010**, *1*, 101–122.

[6] A. Klamt, F. Eckert, J. Reinisch, K. Wichmann, *J. Comput.-Aided Drug Des.* **2016**, *30*, 959–967.

[7] S. Riniker, *J. Chem. Inf. Model.* **2017**, *57*, 726–741.

[8] C. C. Bannan, K. H. Burley, M. Chiu, M. R. Shirts, M. K. Gilson, D. L. Mobley, *J. Comput. Aided Mol. Des.* **2016**, *30*, 927–944.

[9] M. Isik, T. D. Bergazin, T. Fox, A. Rizzi, J. D. Chodera, D. L. Mobley, *J. Comput.-Aided Mol. Des.* **2019**, submitted.

[10] C. I. Bayly, D. McKay, J. F. Truchon, An informal AMBER small molecule force field: parm@Frosst, **2011**.

[11] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, D. A. Case, *J. Comput. Chem.* **2004**, *25*, 1157–1174.

[12] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, et al., *J. Chem. Theory Comput.* **2012**, *9*, 461–469.

[13] N. Schmid, C. D. Christ, M. Christen, A. P. Eichenberger, W. F. van Gunsteren, *Comp. Phys. Comm.* **2012**, *183*, 890–903.

[14] T. Darden, D. York, L. Pedersen, *J. Chem. Phys.* **1993**, *98*, 10089.

[15] I. Tironi, R. Sperb, P. E. Smith, W. F. van Gunsteren, *J. Chem. Phys.* **1995**, *102*, 5451–5459.

[16] S. Riniker, G. A. Landrum, *J. Chem. Inf. Model.* **2015**, *55*, 2652–2574.

[17] P. C. D. Hawkins, A. G. Skillman, G. L. Warren, B. A. Ellingson, M. T. Stahl, *J. Chem. Inf. Model.* **2010**, *50*, 572–584.

[18] P. Eastman, PDBFixer, **2019**.

[19] D. L. Mobley, C. C. Bannan, A. Rizzi, C. I. Bayly, J. D. Chodera, V. T. Lim, N. M. Lim, K. A. Beauchamp, D. R. Slochower, M. R. Shirts, M. K. Gilson, P. K. Eastman, *J. Chem. Theory Comput.* **2018**, *14*, 6076–6092.

[20] A. Jakalian, B. L. Bush, D. B. Jack, C. I. Bayly, *J. Comput. Chem.* **2000**, *21*, 132–146.

[21] A. Jakalian, D. B. Jack, C. I. Bayly, *J. Comput. Chem.* **2002**, *23*, 1623–1641.

[22] D. A. Case, R. M. B. W. Botello-Smith, D. S. Cerutti, T. E. Cheatham, T. A. Darden, R. E. Duke, et al., AMBER 16, **2016**.

[23] P. Bleiziffer, K. Schaller, S. Riniker, *J. Chem. Inf. Model.* **2018**, *58*, 579–590.

[24] J. Swails, C. Hernandez, D. L. Mobley, H. Nguyen, L. P. Wang, P. Janowski, ParmEd, **2010**.

[25] H. J. C. Berendsen, D. van der Spoel, R. van Drunen, *Comput. Phys. Commun.* **1995**, *91*, 43–56.

[26] R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, V. S. Pande, *Biophys. J.* **2015**, *109*, 1528–1532.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

[28] S. Flennerhag, ML-Ensemble, **2017**.

[29] D. L. Mobley, J. P. Guthrie, *J. Comput. Aided Mol. Des.* **2014**, *28*, 711–720.

[30] G. D. R. Matos, D. Y. Kyu, H. H. Loeffler, J. D. Chodera, M. R. Shirts, D. L. Mobley, *J. Chem. Eng. Data* **2017**, *62*, 1559–1569.

[31] A. V. Marenich, C. P. Kelly, J. D. Thompson, G. D. Hawkins, C. C. Chambers, D. J. Giesen, P. Winget, C. J. Cramer, D. G. Truhlar, Minnesota Solvation Database, version 2012, Minneapolis, **2012**.

[32] I. Sushko, S. Novotarskyi, R. Körner, A. K. Pandey, M. Rupp, W. Teetz, S. Brandmaier, A. Abdelaziz, V. V. Prokopenko, V. Y. Tanchuk, R. Todeschini, A. Varnek, G. Marcou, P. Ertl, V. Potemkin, M. Grishina, J. Gasteiger, C. Schwab, I. I. Baskin, V. A. Palyulin, E. V. Radchenko, W. J. Welsh, V. Kholodovych, D. Chekmarev, A. Cherkasov, J. A. de Sousa, Q.-Y. Zhang, A. Bender, F. Nigsch, L. Patiny, A. Williams, V. Tkachenko, I. V. Tetko, *J. Comput.-Aided Mol. Des.* **2011**, *25*, 533–554.

[33] R. Tibshirani, *Journal of the Royal Statistical Society: Series B (Methodological)* **1996**, *58*, 267–288.

[34] J. H. Friedman, *Computational statistics & data analysis* **2002**, *38*, 367–378.

[35] J. S. Smith, O. Isayev, A. E. Roitberg, *Chem. Sci.* **2017**, *8*, 3192–3203.

[36] T. Chen, C. Guestrin in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, **2016**, pp. 785–794.