Retention time prediction to facilitate molecular structural identification with tandem mass spectrometry

Patrik Friedlos

DAS ETH Data Science Capstone Project at SDSC

Supervised by Eliza Harris and Lilian Gasser

September 4, 2022

Abstract

Comparing measured and predicted retention time can improve molecular identification in tandem mass spectrometry. We assess a range of different machine learning methods to predict hydrophobicity, a molecular property that could be used as a proxy for retention time. The performance of the models is evaluated on the benchmark Martel and SAMPL7 datasets. We find that more powerful models perform better in-sample but not necessarily out-of-sample. We also find evidence that ensemble methods can outperform specific models. Additionally, a multitask learning model shows promise for improving the generalization of graph neural networks for hydrophobicity prediction. Finally, we discuss how the generalization of graph neural networks for molecular property prediction could be improved further.

1   Introduction

Identification of molecules in complex mixtures with liquid chromatography followed by tandem mass spectrometry has many applications, such as use in clinical laboratories or analysis of pharmaceutical residues [1, 2]. It has been described how retention time (RT) during liquid chromatography can be used to improve the result of tandem mass spectrometry [3]. For example, one could consider molecules identified by mass spectrometry false positives if their measured retention time varies greatly from the predicted retention time [4].

When it comes to estimating RT, one can distinguish between look-up, index-based, machine learning and modeling-based methods [5]. Look-up-based methods involve keeping a database of measured RT. Index-based methods estimate the RT of peptides based on the contained amino acids. Modeling-based approaches estimate RT from a physical model of chromatographic separation. Finally, machine learning methods predict RT

1

directly from the chemical structure of the substance.

A wide range of machine learning algorithms can be considered for RT prediction. Bouwmeester et al. [6] compared the performance of linear models such as Bayesian ridge regression, LASSO regression and linear support vector regression as well as non-linear models such as artificial neural networks, adaptive boosting, gradient boosting, random forests and non-linear support vector regression. They conclude that gradient boosting performs the best. However they suggest considering different algorithms as well as blended approaches. More recently, graph neural networks (GNNs) have been found outperforming Bayesian ridge regression, convolutional neural networks and random forests at predicting RT [4].

While mass spectra are agnostic to the setup and conditions of the experiment, RT can vary depending on the environment. Various machine learning methods have been developed to account for changing experimental conditions. Some methods perform well even with a small setup-specific training set [6]. Other methods use transfer learning to make them applicable in varying setups [4]. Furthermore, one can predict a retention index relative to reference substances or retention order which is more stable across setups than RT [7].

The octanol-water partition coefficient (P or logP) measuring the hydrophobicity of a substance is strongly related to RT [6]. One could hence consider using hydrophobicity as a proxy for predicting RT. Furthermore, hydrophobicity plays an important role in fields such as drug design [8].

Prediction of logP with machine learning methods is a well established field and many different approaches have been developed. Substructure-based methods cut molecules into fragments and estimate their contribution towards logP. Property-based methods predict logP based on other global properties of molecules [9]. Graph neural networks, which were developed more recently, do not require explicit feature selection but implicitly select features from the graph structure of the molecule alone.

Multiple techniques have been successfully applied to logP prediction. With multitask learning, helper tasks are added to the model, such as related properties or predictions of other models, which can lead to increased performance and generalization [10, 11]. Transfer learning approaches leverage easily available data from a related domain to improve model results. For example, one could train a model on a large dataset of predictions from other models and then tilt the model towards a specific molecular space with a smaller dataset [12]. Data augmentation can be used to improve logP prediction, for example by considering all tautomeric forms of a chemical [13]. Improvements were also made by leveraging predictions of multiple methods and aggregating them into a new model [14]. Recent advances in Graph Neural Networks have been applied to logP prediction as well [15].

In this report, we will consider different machine learning approaches for logP prediction. After presenting the models, they will be trained and their performance evaluated on two independent datasets. Finally, we discuss the results and present some ideas for improvements.

2 Methods

We assume for now that estimates of hydrophobicity can be used to estimate RT and improve the performance of tandem mass spectrometry. We consider a range of different machine learning methods to predict logP, namely multiple linear regression (MLR), random forest (RF), recurrent neural net (RNN) and graph neural network (GNN). Each model is trained five times with the OPERA dataset. We assess the performance with two benchmarks, the Martel and the SAMPL7 datasets.

2.1 Datasets

2.1.1 OPERA

The OPERA dataset contains 13'828 small molecules with measured logP values. The dataset was compiled for the training of the OPERA models. It is based on the PHYSPROP logP dataset, which was further curated by the authors. In particular, errors and mismatches in chemical structure formats and identifiers as well as structure validation issues were corrected. The authors then removed low-quality datapoints, outliers and duplicates. The details of this procedure can be found in their paper [16].

The OPERA dataset contains molecules of masses between 26 and 1'203 Daltons. The logP values range from -5.08 to 11.29. The distribution of the two metrics is shown in Figure 1. The high quality of the dataset as well as the large range of molecules it covers, at least regarding mass and logP, make it ideal as a training set.

For training, we removed molecules with a logP above 8. We further removed molecules that are also contained in the Martel and SAMPL7 datasets in order to make them viable as test sets. A total of 263 molecules were removed, leaving 13'565 in the dataset for training.

2.1.2 Martel

The Martel dataset contains logP measurements of chemically diverse molecules for benchmarking studies [17]. The authors constructed a set of diverse molecules, purchased the compounds and experimentally measured the logP values.

The Martel dataset contains 707 molecules of masses between 160 and 599 Daltons. The logP values range from 0.3 to 6.69. The distribution of the two metrics is shown in Figure 1. The Martel dataset is an ideal test set as it is chemically diverse and commonly used as a benchmark.
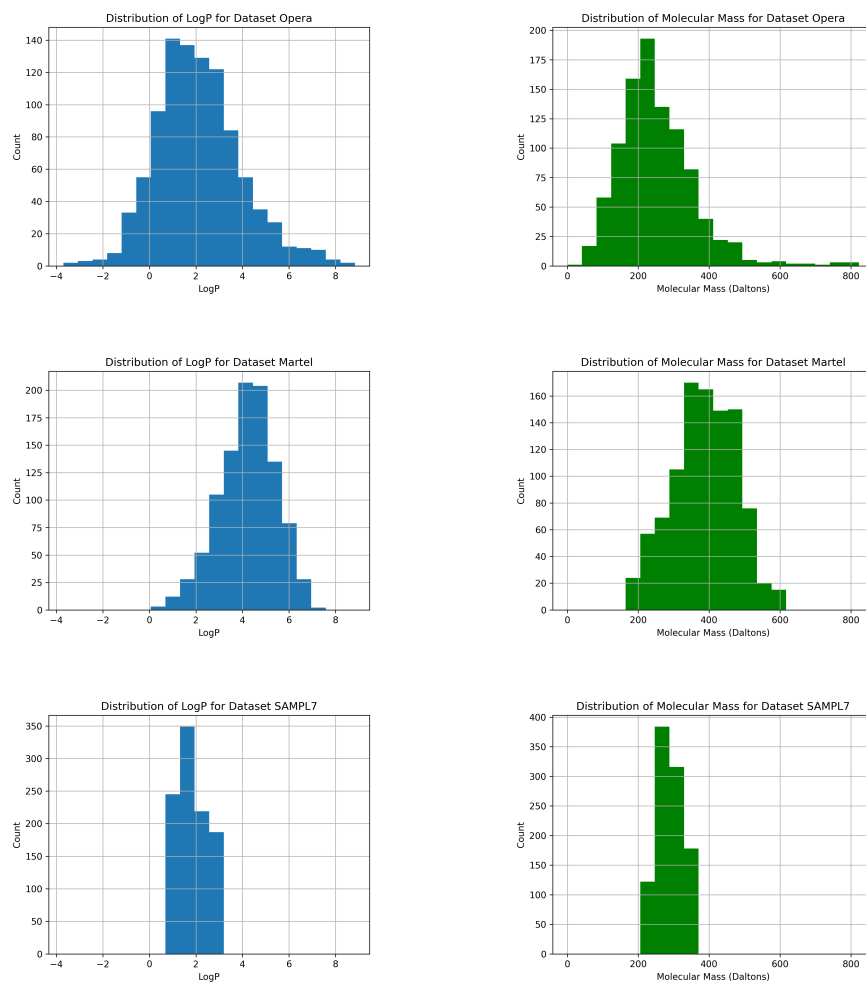
Figure 1: The plots show the distribution of logP as well as the distribution of molecular mass for the OPERA, Martel and SAMPL7 datasets.

2.1.3 SAMPL7

SAMPL7 was a blind challenge in physical property prediction [18]. One part of the challenge consists of predicting logP for 22 molecules measured by the authors.

The SAMPL7 dataset contains molecules of masses between 227 and 365 Daltons. The logP values range from 0.76 to 2.96. The distribution of the two metrics is shown in Figure 1. The SAMPL7 is a valuable test set as we can assess the performance of our general-purpose models in a specialized challenge. For example, some models participating in the challenge were specifically tuned for the molecular space of the challenge [10].

2.2 Models

2.2.1 Multiple linear regression

Despite its simplicity, MLR can show good performance in predicting logP. For example, a linear model was among the best performing in the SAMPL7 blind challenge [19]. Similarly, Mannhold et al. proposed a simple linear model based on just the count of carbon atoms and the count of heteroatoms in the molecule. It outperformed many more complex models in their comparison [9].

For the linear model, we have to perform a feature selection. Firstly, we use the count of each atom type contained in the molecule. Secondly, we use all fragment counts available in the RDKit Python package (rdkit.Chem.Fragments). The regression is then performed using the statsmodels package.

2.2.2 Random forest

We trained a RF model that creates an ensemble of decision trees. We use the standard implementation of sklearn (RandomForestRegressor from sklearn.ensemble) with default settings. We select the same features as for the MLR model.

2.2.3 Recurrent neural net

While predicting molecular properties with RNNs appears to be less common, they have been used for molecule generation in contexts such as drug discovery [20]. In this model, we pass atom representation iteratively through the RNN. As there is no natural ordering of atoms in a molecule, we use the default order of RDKit. Finally, there is a fully connected layer predicting logP from the learned representation of the molecule.

The RNN does not require explicit feature selection, however we have to find a suitable representation for each atom within the molecule. We represent each atom as the atom type itself and its neighboring atoms and

bond types. We find that there are around 500 different combinations of atoms with their neighboring atoms in our datasets, provided we do not consider hydrogen atoms. We then use an embedding layer to reduce the dimensionality of the representation.

2.2.4 Graph neural networks

The use of GNNs for logP prediction has been examined by many authors [15]. We use the same node representation as in the RNN model. After feeding the node representation through 2 linear layers with relu activation, 6 graph convolutional layers as suggested by Kipf and Welling are applied to the graph [21]. The sum is used as the global pooling function. Finally, two more linear layers are used to predict logP. The implementation is done with PyTorch Geometric.

In a second model, we want to evaluate the benefit of recent advances in neighborhood aggregation for molecular property prediction. In particular, we use Principal Neighbourhood Aggregation (PNA) as suggested by Corso et al. [22]. It uses sum, max and var as well as degree-scaled versions thereof in the neighborhood aggregation. We use the Pytorch Geometric implementation of the PNA layer.

In a third model, we want to evaluate the findings of Ryu et al. who report benefits of using gated skip-connection and attention for molecular property prediction [23]. To this end, we use a general GNN layer of Pytorch Geometric based on the work of You et al. which features both attention and skip-connections [24].

Finally, we consider a simple GNN with multitask learning. Multitask learning has been successfully applied to molecular property prediction and logP prediction in particular. Furthermore, it can improve model generalization, which is of major importance should the model be applied to novel molecules. Apart from logP, the model predicts 15 molecular features deemed most relevant by the RF model. This includes counts of atoms but also functional groups, such as the count of benzene rings.

2.3 Training

In order to objectively asses model performances on the Martel and SAMPL7 test sets, we removed these molecules from the OPERA training set. A random 80/20 split of the remaining OPERA dataset is used for training and assessing in-sample performance. Hyperparameters were optimized, if at all, by creating an additional validation set within the training set.

Every model is trained five times with different random initialization, such that variations in training outcomes can be estimated.

2.4 Results

2.4.1 Performance on OPERA

First, we consider the in-sample performance of the models on the OPERA dataset. The root-mean-square error (RMSE) of each model type as well as the standard deviation of the 5 trained models is shown in Table 1. As expected, the more complex models show better performance on both the training and test set. Most models show signs of overfitting. The MLR does not overfit, which can probably be attributed to model simplicity. Interestingly, the multitask model does not overfit at all. This could be explained by competition for model bandwidth among the different tasks.

Model	RMSE Train	RMSE Test
MLR	0.86 ± 0.00	0.88 ± 0.00
RF	0.25 ± 0.00	0.61 ± 0.00
RNN	0.27 ± 0.00	0.45 ± 0.00
GNN	0.27 ± 0.01	0.45 ± 0.01
GNN + PNA	0.20 ± 0.01	0.51 ± 0.01
GNN + Attention	0.23 ± 0.01	0.45 ± 0.01
GNN + Multitask	0.47 ± 0.01	0.46 ± 0.00

Table 1: Comparison of in-sample performance of the models as measured by RMSE for both the training and test sets.

2.4.2 Performance on Martel

Next, we compare the performance of the models with other methods on the Martel benchmark dataset. Table 2 shows the RMSE of our and various other models on the Martel dataset.

Model	RMSE
COSMO-RS [1]	0.93
Lui at el. [2]	1.03
JPlogP-Coeff	1.04
JPlogP—library	1.05
LogP4Average	1.12
XlogP3-AA	1.16
AAM	1.18
MLR	1.20 ± 0.00
GNN + Multitask	1.20 ± 0.04
RNN	1.21 ± 0.02
Molinspiration	1.21
SlogP	1.24
ALogP (Vega)	1.24
Biobyte CLogP	1.24
GNN + Attention	1.27 ± 0.05
XLogP2	1.28
AlogPS logP	1.29
GNN	1.30 ± 0.04
GNN + PNA	1.39 ± 0.05
ACD	1.39
KowWIN	1.42
RF	1.44 ± 0.01
Meylan (Vega)	1.66
Mannhold LogP	1.71
MLogP (Vega)	1.95
AlogP (CDK)	3.72

Table 2: The table shows the RMSE of different models for predictions on the Martel dataset. Our models are highlighted in bold. Unless stated otherwise, results are taken from Plante et al. [14].

The binned absolute errors of the models can be considered to get a better understanding of the distribution of errors. Table 3 shows the binned absolute errors for our models on the Martel dataset. The values of XlogP3-AA are added for reference and are taken from Plante et al. [14].

	0-0.5	0.5-1	1-1.5	1.5-2	> 2
XlogP3-AA	33.0%	27.9%	20.7%	10.5%	8.1%
MLR	30.8%	29.8%	18.7%	11.9%	8.8%
RF	25.1%	24.4%	19.7%	14.9%	15.8%
RNN	32.8%	28.1%	18.3%	12.0%	8.8%
GNN	27.9%	26.4%	21.4%	12.7%	11.6%
GNN + PNA	26.5%	25.2%	19.6%	14.1%	14.5%
GNN + A	29.3%	26.8%	20.5%	12.6%	10.8%
GNN + M	32.8%	29.2%	18.2%	10.1%	9.6%

---

[1]COSMO-RS predicts logP with a physical model [13]

[2]This model uses stochastic gradient descent-optimized multilinear regression with 1438 physicochemical descriptors [25]

Table 3: The table shows the distribution of absolute errors of the different models for predictions on the Martel dataset.

We can plot the prediction error as a function of measured logP to assess if the models perform better for some logP ranges. Figure **??** shows the prediction errors of the MLR model against measured logP of the Martel dataset. At least visually, the distribution of prediction errors appears to be homoscedastic. However, the model systematically underestimates logP values for the martel dataset. The pattern becomes more pronounced for higher measured logP values. This finding is consistent across different models and can probably be attributed to the differences between the OPERA and the Martel datasets. Whereas the peak of the logP distribution is around a value of 2 for Opera, it is above 4 for the Martel dataset. The models are hence biased towards the logP distribution of the training data.
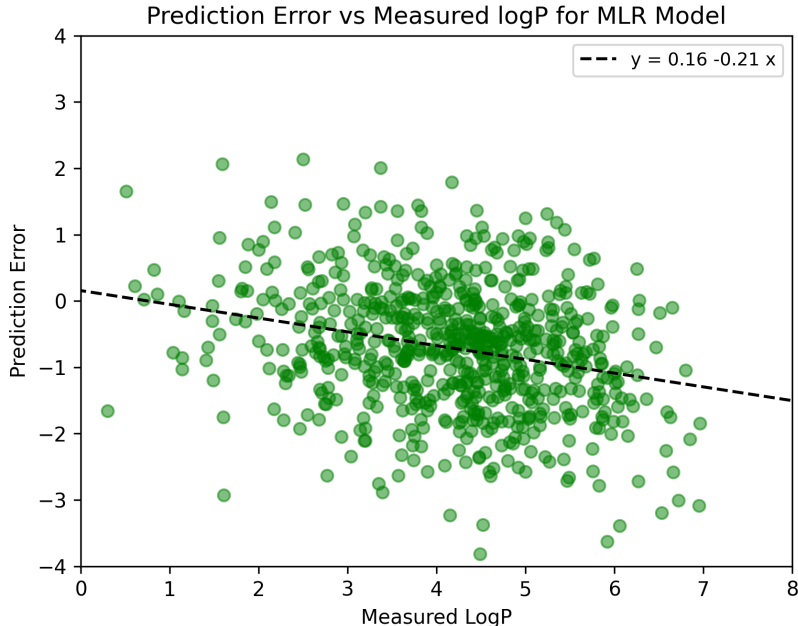

Figure 2: The plot shows the prediction error against the measured logP values for each molecule in the Martel dataset. Although the errors appear homoscedastic, the model systematically underestimates logP values as indicated with the linear regression line.

We can consider the correlation of errors of the different models in order to evaluate to what extent they have learned different information. In case correlations between models are low, an ensemble model might be able to outperform the individual models. Table 4 shows the error correlations of the different models on the Martel dataset.

	MLR	RF	RNN	GNN	GNN + PNA	GNN + A	GNN + M
MLR	1.00	0.68	0.62	0.60	0.56	0.60	0.71
RF	0.68	1.00	0.55	0.53	0.49	0.51	0.56
RNN	0.62	0.55	1.00	0.81	0.72	0.78	0.80
GNN	0.60	0.53	0.81	1.00	0.84	0.90	0.87
GNN + PNA	0.56	0.49	0.72	0.84	1.00	0.85	0.80
GNN + A	0.60	0.51	0.78	0.90	0.85	1.00	0.84
GNN + M	0.71	0.56	0.80	0.87	0.80	0.84	1.00

Table 4: The table shows the correlation of errors of the different models for predictions on the Martel dataset. The mean error of five models of each type is considered for the comparison.

We can see that model errors on the Martel dataset are correlated. The correlation is particularly high amongst the GNN models, which might be expected. Interestingly, the errors of the multitask model not only correlate with the other GNNs but also with the MLR and the RF. This might suggest that the multitask model indeed relies on the features learned by the helper tasks to some extent. However, there is no perfect correlation between models and we might expect an ensemble approach to perform even better. If we take the mean of all model predictions, the RMSE on Martel turns out to be 1.14, better than any individual model.

2.4.3   Performance on SAMPL7

Finally, we evaluate the performance of the models on the highly specific SAMPL7 dataset. We take all ranked submissions from the SAMPL7 challenge as the benchmark [26]. Table 5 shows the performance of the different approaches on the SAMPL7 dataset as measured by RMSE.

Model	RMSE
Chemprop	0.49
GROVER	0.62
ffsampled_deeplearning_cl1	0.63
TFE_Attentive_FP	0.64
COSMO-RS	0.66
GNN	0.71 ± 0.10
TFE MLR	0.71
ClassicalGSG DB3	0.74
GNN + Multitask	0.80 ± 0.08
GNN + Attention	0.80 ± 0.08
RF	0.87 ± 0.05
GNN + PNA	0.99 ± 0.10
MLR	1.03 ± 0.00
TFE IEFPCM MST	1.04
RNN	1.07 ± 0.10
NES-1 (GAFF2/OPC3) B	1.21
MD (CGenFF/TIP3P)	1.47
EC_RISM_wet	1.65
TFE-NHLBI-TZVP-QM	1.70
MD-EE-MCC (GAFF-TIP4P-Ew)	1.92
TFE b3lypd3	2.32
TFE-SMD-solvent-opt	2.54
Ensemble Martel	3.40
QSPR_Mordred2D_TPOT_AutoML	3.76

Table 5: The table shows the performance as measured by RMSE for all ranked submissions to the SAMPL7 challenge compared to our models.

2.5 Discussion

2.5.1 Model performance

The results generally show that in-sample performance increases with more powerful methods. As expected, these methods were also more prone to overfitting. The performance of the RNN is remarkable as it performs similarly to the GNNs without any explicit notion of the graph structure of the molecules. For GNNs we found little benefit in using PNA or attention layers as opposed to a GCN.

When it comes to out-of-sample performance, we find that more powerful models offer little benefit a priori. In fact, the MLR performs the best on the Martel dataset despite being the simplest model. Furthermore, there seems to be little correlation between the generalization of the models to Martel and SAMPL7. Maybe this could be attributed to the different chemical spaces and logP ranges of the two datasets. We also have to consider that SAMPL7 is a small dataset of specific molecules and is not representative for broad model generalization.

The discrepancy between in-sample and out-of-sample performance can in part be explained by model bias towards the logP value distribution of the in-sample dataset. We have seen that a model trained on the Opera dataset systematically underestimates logP on the Martel dataset. If we train a MLR model with training data from Opera but sampled according to a uniform logP distribution using bins on (-2, 8), RMSE on Martel is reduced from 1.20 to 1.16. If we were to sample according to the logP distribution of Martel, RMSE drops to just 0.96. Of course, the logP distribution of the target dataset can not be known in advance. One could however consider an approach where models are iteratively trained with data sampled according to the logP distribution of the predictions of the previous model.

The multitask model might offer a good compromise between in-sample and out-of-sample performance. The loss of performance due to the helper tasks on the in-sample test set is negligible. However, generalization seems to be better, at least to the Martel dataset.

One could argue that the models perform reasonably well on the SAMPL7 dataset, considering that it contains specific types of molecules. In particular, some models competing in the challenge were specifically tuned for the molecular space.

The variance of model performance is relatively high in particular for the GNNs. This suggests that a significant part of the performance has to be attributed to the initialization. An ensemble of models could be used to improve this issue.

The computational cost of running the models is difficult to estimate, as it depends on the implementation as well as the available hardware. One also has to distinguish between the cost of training the models and using the models. The former is typically high for neural nets. Generally an MLR model should be orders of magnitudes faster than a GNN model or a physical models such as COSMO-RS. Predictions can be parallelized however, lowering the runtime of batch predictions if required.

2.5.2   Ideas for model improvement

The implmentation of the models was kept relatively straightforward. In particular, we did not perform systematic hyperparameter optimization. For the models with explicit feature selection, the MLR and the RF, only atom counts and default fragment counts of RDKit were used. More careful tuning of the models would probably increase performance in some cases.

An interesting paradigm for logP prediction is making it relative to reference molecules with known logP. Such an approach is used by XLOGP3, amongst others [27]. Although this comes with additional challenges, such

as having to determine the appropriate reference molecules, it might also improve model performance. In particular because such models might not acquire the systematic bias towards logP values found in the training set.

Further improvements could be made by using more datasets. An aggregation of multiple datasets could cover a broader range of molecules and yield increased performance. Different sampling and splitting techniques, such as scaffold splitting, could also be considered.

For GNNs specifically, a wide range of different architectures can be considered. This involves choosing from a wide range of convolutional layers, neighborhood aggregations and global pooling layers. Furthermore, there are many options for the initial representation of the atoms. We constructed a representation that includes both the neighboring bonds and atoms. However, there are also setups which handle edge features separately. One could even consider a related graph where every bond in the molecule is a node and the atoms being connected are encoded as node features.

2.5.3   Improving generalization

It seems apparent that a good understanding of the chemical space of both the training set and the target molecules is crucial. If they are similar, the use of powerful methods such as GNNs may well be justified. If they are different, simpler models may generalize a lot better. One could imagine that the latter is more often the case in most practical settings.

One solution to achieving good performance on a set of target molecules is the careful construction of a training set. For example, Lenselink et al. created a tailored training set for their participation in the SAMPL7 challenge [10].

Many well-established general logP prediction methods, such as XLOGP3, are essentially linear models [27]. One has to wonder if GNNs could outperform such methods in broad molecular spaces. The following ideas could be considered to construct a model with broad generalization capabilities:

- Multitask learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias [28]. It has already been successfully applied to logP prediction [10]. We have also seen the benefit of this approach. As additional tasks, functional groups deemed relevant for logP prediction by other models were used. One could also consider using a RT dataset such as SMRT of METLIN as RT is related to logP [29]. Finally, quantum mechanical properties related to logP could be considered as well. To this end, a dataset such as QMugs could be used [30].
- Transfer learning approaches could be used to enhance performance.

In transfer learning, more easily available data from a different but related domain is used for training [31]. For example, a GNN could learn to count functional groups of molecules relevant for logP prediction on a large diverse dataset. The model could then be further trained to predict logP. One would either train a small new part of the model or retrain with elastic weight consolidation in order to nudge the model to use the more generalizing features learned on the large dataset if possible.

- Combining multitask and transfer learning, a multitask model predicting counts of functional groups and logP could be trained. Next, the model would be used to add predicted logP to a large and diverse dataset for pseudo-rehersal. Finally, the multitask model could be retrained on the large dataset, hoping the helper tasks, on which logP prediction now relies on to a degree, are learned in a more generalizing way.

- It is generally recommended to use a time or scaffold split instead of a random split on the dataset as this provides a more realistic assessment of model performance [10]. One could also tune model parameters with respect to the worst performing of many random splits. Additionally, an adversarial model could be used where an additional model constructs the training set. The resulting model would hence be the best performing on the most difficult split and would arguably generalize better.

- Maybe model generalization to a diverse dataset could be judged a priori to some extent. For example, one could hypothesize that similar distributions of neuron activations between the training and a diverse molecular dataset indicate better generalization. Similarly, a pruning method could be considered in which nodes are removed from the model if its loss reduction on the training set significantly outweighs the loss reduction on the predictions of a large diverse dataset.

- The training set could be augmented to be more representative of a large and diverse dataset. For example, more weight could be added to structurally underrepresented molecules in the training set.

Using any of the above approaches is challenging as they typically come with additional hyperparameters. Not only can the line between too little and too much regularization be narrow, there are also limited independent datasets available for tuning and validation of the approach.

2.5.4 Implications for retention time prediction

It is conceivable that logP predictions can be used as a proxy for RT prediction. However, one might have to consider that other factors besides logP influence RT. An alternative approach could be a transfer learning approach from existing RT datasets to the specific setup as presented by Yang et al. [4]. The value of using RT in tandem mass spectrometry might be greatly improved by not just predicting RT but also errors. This

way, molecules with small expected errors on RT can be identified as false positives more aggressively.

3   Conclusion

We set out to evaluate different machine learning algorithms for logP prediction in order to facilitate molecular structural identification with tandem mass spectrometry. Like others before us, we found that different types of models can show good performance. Both explicit feature selection, such as the use of counts of functional groups as well implicit feature selection through a GNN are feasible. Further, different models create highly, but not perfectly correlated predictions. This indicates that ensemble methods could further improve predictions.

We conclude that a thorough understanding of the molecular space of both the training as well as the target molecules is essential for logP prediction. Models can show great in-sample performance but lack generalization, and vice versa. More powerful approaches tend to show better in-sample performance. Simpler models, such as MLR, might be more suitable for predicting logP out-of-sample.

Modern techniques such as GNNs have been used for logP prediction, often in combination with multitask learning or transfer learning and with a specific set of target molecules. Many of the models geared towards logP prediction in broad molecular spaces, such as XlogP3-AA, have been developed many years ago. We have seen that multitask models might be valuable for creating generalizing GNNs for molecular property prediction. Quite possibly, further improvements could be made with the techniques outlined in the discussion.

References

[1] J. M. van den Ouweland and I. P. Kema, "The role of liquid chromatography–tandem mass spectrometry in the clinical laboratory," *Journal of chromatography B*, vol. 883, pp. 18–32, 2012.

[2] M. Petrović, M. D. Hernando, M. S. Díaz-Cruz, and D. Barceló, "Liquid chromatography–tandem mass spectrometry for the analysis of pharmaceutical residues in environmental samples: a review," *Journal of Chromatography A*, vol. 1067, no. 1-2, pp. 1–14, 2005.

[3] E. F. Strittmatter, L. J. Kangas, K. Petritis, H. M. Mottaz, G. A. Anderson, Y. Shen, J. M. Jacobs, D. G. Camp, and R. D. Smith, "Application of peptide lc retention time information in a discriminant function for peptide identification by tandem mass spectrometry," *Journal of proteome research*, vol. 3, no. 4, pp. 760–769, 2004.

[4] Q. Yang, H. Ji, H. Lu, and Z. Zhang, "Prediction of liquid chromatographic retention time with graph neural networks to assist in small molecule identification," *Analytical Chemistry*, vol. 93, no. 4, pp. 2200–2206, 2021.

[5] L. Moruz and L. Käll, "Peptide retention time prediction," *Mass spectrometry reviews*, vol. 36, no. 5, pp. 615–623, 2017.

[6] R. Bouwmeester, L. Martens, and S. Degroeve, "Comprehensive and empirical evaluation of machine learning algorithms for small molecule lc retention time prediction," *Analytical chemistry*, vol. 91, no. 5, pp. 3694–3703, 2019.

[7] C. Qu, B. I. Schneider, A. J. Kearsley, W. Keyrouz, and T. C. Allison, "Predicting kovats retention indices using graph neural networks," *Journal of Chromatography A*, vol. 1646, p. 462100, 2021.

[8] I. V. Tetko and P. Bruneau, "Application of alogps to predict 1-octanol/water distribution coefficients, logp, and logd, of astrazeneca in-house database," *Journal of pharmaceutical sciences*, vol. 93, no. 12, pp. 3103–3110, 2004.

[9] R. Mannhold, G. I. Poda, C. Ostermann, and I. V. Tetko, "Calculation of molecular lipophilicity: State-of-the-art and comparison of log p methods on more than 96,000 compounds," *Journal of pharmaceutical sciences*, vol. 98, no. 3, pp. 861–893, 2009.

[10] E. B. Lenselink and P. F. Stouten, "Multitask machine learning models for predicting lipophilicity (logp) in the sampl7 challenge," *Journal of Computer-Aided Molecular Design*, vol. 35, no. 8, pp. 901–909, 2021.

[11] F. Capela, V. Nouchi, R. Van Deursen, I. V. Tetko, and G. Godin, "Multitask learning on graph neural networks applied to molecular property predictions," *arXiv preprint arXiv:1910.13124*, 2019.

[12] Y.-K. Chen, S. Shave, and M. Auer, "Mrlogp: transfer learning enables accurate logp prediction using small experimental training datasets," *Processes*, vol. 9, no. 11, p. 2029, 2021.

[13] N. Ulrich, K.-U. Goss, and A. Ebert, "Exploring the octanol–water partition coefficient dataset using deep learning techniques and data augmentation," *Communications Chemistry*, vol. 4, no. 1, pp. 1–10, 2021.

[14] J. Plante and S. Werner, "Jplogp: an improved logp predictor trained using predicted data," *Journal of cheminformatics*, vol. 10, no. 1, pp. 1–10, 2018.

[15] O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel, and T. Langer, "A compact review of molecular property prediction with graph neural networks," *Drug Discovery Today: Technologies*, vol. 37, pp. 1–12, 2020.

[16] K. Mansouri, C. M. Grulke, R. S. Judson, and A. J. Williams, "Opera models for predicting physicochemical properties and environmental fate endpoints," *Journal of cheminformatics*, vol. 10, no. 1, pp. 1–19, 2018.

[17] S. Martel, F. Gillerat, E. Carosati, D. Maiarelli, I. V. Tetko, R. Mannhold, and P.-A. Carrupt, "Large, chemically diverse dataset of log p measurements for benchmarking studies," *European Journal of Pharmaceutical Sciences*, vol. 48, no. 1-2, pp. 21–29, 2013.

[18] T. D. Bergazin, N. Tielker, Y. Zhang, J. Mao, M. R. Gunner, K. Francisco, C. Ballatore, S. M. Kast, and D. L. Mobley, "Evaluation of log p, pka, and log d predictions from the sampl7 blind challenge," *Journal of computer-aided molecular design*, vol. 35, no. 7, pp. 771–802, 2021.

[19] K. Lopez, S. Pinheiro, and W. J. Zamora, "Multiple linear regression models for predicting the n-octanol/water partition coefficients in the sampl7 blind challenge," *Journal of computer-aided molecular design*, vol. 35, no. 8, pp. 923–931, 2021.

[20] E. J. Bjerrum and R. Threlfall, "Molecular generation with recurrent neural networks (rnns)," *arXiv preprint arXiv:1705.04612*, 2017.

[21] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[22] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, "Principal neighbourhood aggregation for graph nets," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13260–13271, 2020.

[23] S. Ryu, J. Lim, S. H. Hong, and W. Y. Kim, "Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network," *arXiv preprint arXiv:1805.10988*, 2018.

[24] J. You, Z. Ying, and J. Leskovec, "Design space for graph neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17009–17021, 2020.

[25] R. Lui, D. Guan, and S. Matthews, "A comparison of molecular representations for lipophilicity quantitative structure–property relationships with results from the sampl6 logp prediction challenge," *Journal of computer-aided molecular design*, vol. 34, no. 5, pp. 523–534, 2020.

[26] T. D. Bergazin, D. L. Mobley, M. Amezcua, H. Grosjean, M. Isik, D. Slochower, J. Chodera, N. Tielker, D. Ray, S. Sasmal, and K. Murakumo, "samplchallenges/SAMPL7: Version 1.1: Update logP analysis; release PHIP2 analysis," Nov. 2021.

[27] T. Cheng, Y. Zhao, X. Li, F. Lin, Y. Xu, X. Zhang, Y. Li, R. Wang, and L. Lai, "Computation of octanol- water partition coefficients by guiding an additive model with knowledge," *Journal of chemical information and modeling*, vol. 47, no. 6, pp. 2140–2148, 2007.

[28] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[29] X. Domingo-Almenara, C. Guijas, E. Billings, J. R. Montenegro-Burke, W. Uritboonthai, A. E. Aisporna, E. Chen, H. P. Benton, and G. Siuzdak, "The metlin small molecule dataset for machine learning-based retention time prediction," *Nature communications*, vol. 10, no. 1, pp. 1–9, 2019.

[30] C. Isert, K. Atz, J. Jiménez-Luna, and G. Schneider, "Qmugs, quantum mechanical properties of drug-like molecules," *Scientific Data*, vol. 9, no. 1, pp. 1–11, 2022.

[31] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.