Prediction and Uncertainty Quantification of Small-Molecule Retention Times

Priyanka Raghavan, Jenna Fromer, Nathan Stover

10.C51 Project - Spring 2022

Date of Submission: 5/9/22

Faculty Supervisor: Connor W. Coley

Contents

Abstract

In this project, we aimed to predict high-performance liquid chromatography (HPLC) retention times using the METLIN SMRT dataset [1]. Since previous work has shown high-accuracy predictions for retention times, our additional goal was to analyze uncertainty in model predictions on these data. We implemented several models based on fingerprint and graph representations of molecules and evaluated performance on both random and structure-based splits. We found that a feed-forward neural network and multi-feature-embedded MPNN (Chemprop) performed comparably to top literature results. We observed that uncertainty values derived from ensembling on the random split were not necessarily meaningful estimates of the true uncertainty. Additionally, we carried out model ensembling and Tanimoto distance comparison on the vanilla MPNN with a scaffold split to quantify and analyze the uncertainty of our predictions. Results of this uncertainty quantification indicate that prediction error for this task has a slight dependence on a molecule's similarity to the training set.

Introduction and Prior Work

High performance liquid chromatography (HPLC) is a commonly used purification method in small molecule synthesis to separate a desired product from a reaction mixture. The retention times of mixture components ultimately determine the separability of the components and the purity of the desired product.

When performing reactions with little literature precedent, it is often necessary to have an a priori sense of the retention times of the starting material(s), intermediate(s), and product(s), as overlap of the chemical species in the liquid chromatography column can lead to poor purification and necessitate extra purification steps. A machine learning model that makes use of the myriad of HPLC data to accurately predict retention times could reduce the cost, time, and resources necessary to synthesize small molecules.

In this project, we attempt to build a quantitative structure-property relationship (QSPR) model to (1) predict the retention time of small molecules in HPLC based on molecular representations and (2) predict the uncertainty in our model's retention time prediction. The latter contribution can allow either a chemist or an automated synthesis planner to estimate the probability that a molecule will be easily purified, which can have effects on synthesis prioritization and scheduling.

Past work in this area has focused on predicting small molecule retention times using deep learning methods as well as graph neural networks [2–4]. The METLIN small molecule retention time (SMRT) dataset (see **Data** section) was first introduced by Domingo-Almenara et al in 2019 [1]. The authors of this publication also trained a neural network using molecular fingerprints and compared its performance to that of other architectures. Ultimately, the neural network model trained with fingerprints outperformed other approaches [1]. Yang et al. later attempted to also predict retention times using this database and a graph neural network (GNN) operating on molecular graphs [2]. A mean relative error under 5% was achieved with the GNN. The utilized GNN randomly sampled subgraphs of each molecule instead of using message-passing on the entire molecular graph [2]. An additional attempt by Kensert et al. demonstrated the effectiveness of a graph convolutional network in RT prediction [4], achieving relative errors below 5%, comparable to those observed by Yang et al.

In this project, we implement a feed forward neural network (FFNN), message passing neural network (MPNN) with atomic number-based atom features, and a multi-feature-embedded MPNN (Chemprop [5]) to predict retention times. For the first two architectures, we additionally train an ensemble of models to extract variance of the model and uncertainty information about our predictions.

Data

The SMRT (small-molecule retention time) dataset consists of 80,038 small molecules and their associated retention times. The data is experimentally acquired from high-performance liquid chromatography-mass spectrometry (HPLC-MS) trials, and corresponds to molecules from the METLIN library, spanning natural products, metabolites, and drug-like molecules. Each molecule is reported by its PubChem ID and InChI key, enabling calculation of SMILES strings and molecular fingerprints. Retention times range from 0.3 to 1471.7 s, and molecular weights range from 113.08 to 738.87 g/mol. Fig. 1 shows a distribution plot of the retention times and molecular weights.

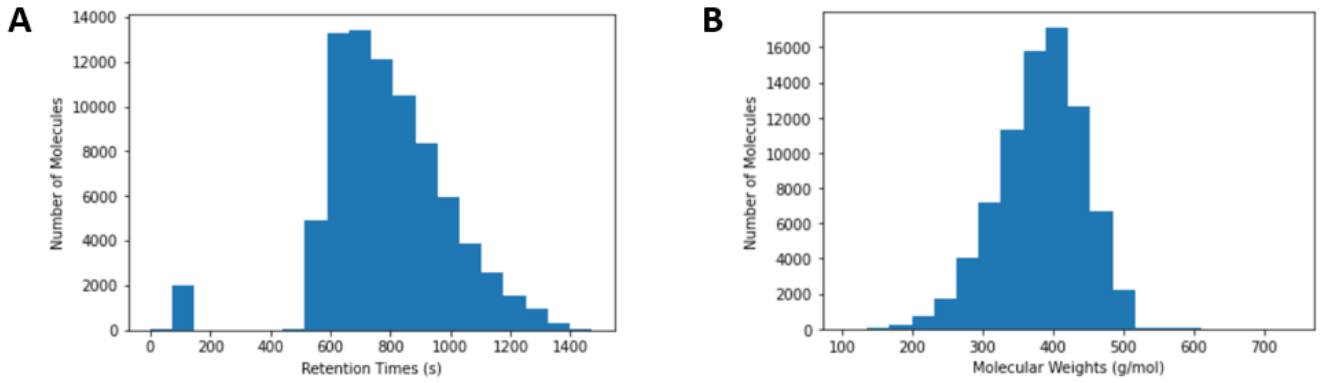

Figure 1: Histograms of (A) retention times and (B) molecular weights. The majority of molecules have retention times between 400 and 1400 seconds, and about 2000 molecules are non-retained, in that they elute very quickly. Note that (B) does not include 81 compounds whose SMILES strings could not be resolved in RDKit (see Methods - Data Processing for more info).

This dataset contains retention time measurements for one specific type of column: a standard, reverse-phase column with water, acetonitrile, and formic acid solvents. Therefore, while this data will not be directly applicable to other columns in practice, this still represents a very standard laboratory column and our work can potentially thus be useful in many experimental applications. Additionally, the data was collected through experimental HPLC analysis, which is itself subject to some degree of error. Pure standards of the compounds in the dataset were run twice through the column and a mean retention time variance of 36 seconds was observed. Due to the large size of the dataset, we would expect errors of a similar magnitude to the experimental retention time variance.

Methods

a. Data Processing and Featurization

The dataset was downloaded from the Figshare provided by the original METLIN paper [1]. We first confirmed that there were no duplicate molecules and generated SMILES strings from the provided InChI keys using RDKit. 81 molecules whose SMILES strings were not able to be rendered by RDKit were excluded. 2048-bit Morgan fingerprints were also generated with RDKit. We additionally scale the output variable, retention time, from seconds to minutes to avoid large gradients and unstable model training. A batch size of 512 was kept constant for all experiments. We did consider removing non-retained molecules from the dataset, but decided not to (Appendix A).

For graph-based approaches, the molecules' SMILES strings are used to generate a feature set on the nodes (atoms) using atomic numbers, and a list of edge connection information for each molecule. Other choices of featurization could have included various molecule-level descriptors, but since past work has

shown good performance with fingerprint and graph-based representations [1, 2], we chose to use these.

b. Data Splitting

As in the prior literature, random splitting was primarily used in this work, with 10 percent of all data held out as a final testing set (80/10/10 train/val/test). All models trained with this split were evaluated with respect to mean and median average error (MAE and MedAE), mean relative error (MRE), and R^2, in accordance with literature results.

In addition to random splitting, splitting was also performed with the goal of clustering similar molecules in the dataset. Two different clustering methods were used to separate data, to test the model's ability to generalize between areas of chemical space. The first method used principal component analysis to compress 2048-bit fingerprint representations of molecules into 10 dimensions. We used k-means clustering (where k=5) to identify discrete, non-overlapping zones in chemical space (Appendix C). These clusters were used either as testing or training data. The second strategy for data clustering used the scaffolds of molecules, which are formed by removing all functional groups. Using the same parameters as the first method, we performed k-means clustering on the PCA-compressed fingerprints of the molecular scaffolds. Since this method ignored functional groups, it was expected to focus more on clustering molecules of similar sizes and shapes. Conversely, the first method was expected to form clusters of molecules that shared similar functional groups. Splitting the data in this way not only allowed us to determine how well the model generalized, but also to determine how splitting data using different conceptions of chemical space affected model performance. See Appendix C for details on this clustering.

c. Baseline Approaches

As literature precedents have mostly focused on random splits, we first focused on a few baseline models using the generated random split detailed in section (b). We implemented linear and k-nearest neighbors (kNN) regression baselines, optimizing the kNN model with respect to the number of nearest neighbors using grid search with 5-fold cross validation in `scikit-learn`. These models were then evaluated on the held-out test set.

Next we attempted to replicate the results from Domingo et al [1] using a feed forward neural network (FFNN) with four hidden layers, trained on Morgan Fingerprints. After similar results were obtained, hyperparameter optimization was performed using `Optuna` [6]. A median pruner was implemented with patience equal to 3 to speed the hyperparameter optimization (see the beginning Appendix A for optimization results). A total of 200 trials were performed.

d. Message-Passing Neural Network Models

We first attempted to replicate the current best literature precedent for this task, the MPNN implemented by Yang et al [2]. We chose to use the same model architecture and hyperparameters as reported in the paper (see Appendix A for hyperparameters), and used a "Vanilla" MPNN with node features represented by atomic numbers, and no edge features. The model was trained using gradient descent until convergence was reached, for both random and structure-based splits described in section (b).

Chemprop, a more complex MPNN that has shown success in several QSPR applications [7–9], uses directed edge messages and a variety of atom features including atomic number, mass, and bond/charge information [5]. Treating Chemprop as a black-box model, we trained it as-is on our randomly split dataset.

e. Uncertainty Quantification

We quantify the epistemic uncertainty in our predictions using the ensemble method [10]. Here, 5 different models are trained on the same data with different randomly initialized parameters. Each molecule in the test set correspondingly has 5 predicted retention times. The prediction is estimated as the mean of the 5 predictions, and the uncertainty in the prediction is the standard deviation of these predictions. Note that the width of the HPLC retention peak for the collected data ranges from 20 to 30s [1]. If we assume

that the peak width corresponds to the uncertainty, we would expect epistemic uncertainty values at this order of magnitude. This method was evaluated on both the FFNN and Vanilla MPNN. In this work, we used the mean base-2 log Tanimoto distances of the 8 nearest neighbors to find the relative distance between a new molecule and the training set [10]. These distance mt

Results

a. Random Split Results

Results of hyperparameter optimization on the FFNN are shown in Appendix A. Using a random split on our resulting model, this model achieves a MedAE of 32 seconds, a MAE of 47 seconds, and a MRE of 31%, all of which outperform the results of Domingo et al [1].

Hyperparameter optimization was not performed on Chemprop due to the computational expense of such a calculation. However, using 150 epochs and otherwise default options in Chemprop, the model achieves a MedAE of 26 seconds, a MAE of 46 seconds, and a MRE of 39%. The performance of all models is summarized in Table 1, and parity plots are given in Appendix A.

Model	MAE (min)	MedAE (min)	MRE	R^2
Linear	1.27	0.89	0.54	0.72
KNN	1.37	0.89	0.47	0.65
FFNN	0.788	0.532	0.31	0.87
Vanilla MPNN	1.18	0.79	0.58	0.74
Chemprop	0.764	0.434	0.386	0.85
GNN-RT [2]	0.67	0.42	0.05	0.85

Table 1: Evaluation metrics for all models. We compare our results to the best-performing model found previously. Note that the GNN-RT [2] omitted non-retained molecules before training, which likely resulted in the observed improvement in metrics. See Appendix A for a more direct comparison to literature results.

For both models, we propose that the median absolute error is the most indicative of performance, as the error of the non-retained molecules can strongly skew the mean results. While the Chemprop model does outperform the FFNN in median absolute error, the computational time required must also be factored in to the usability of such models.

b. Non-Random Split Results

Next we evaluated the effect of 2 different non-random split types (see "Splitting" section in Methods) on the Vanilla MPNN (Fig. 2).

The train and test performance is much more similar in the PCA-generated clusters. This is likely because molecules in the training set may have also had features similar to molecules in the test set, whereas in the scaffold-split case, there was likely to be much more difference. This implies that the extrapolation capability of these models is somewhat limited.

c. Uncertainty Analysis

An ensemble of 5 different FFNN models was generated using the same random split for all models. A histogram of the computed standard deviations is shown in Figure 3. The same was performed for the vanilla MPNN. We observe that the performance of models in each ensemble are not heavily dependent on the initialization, shown by the minimal deviation of the MedAE in Figure 3.

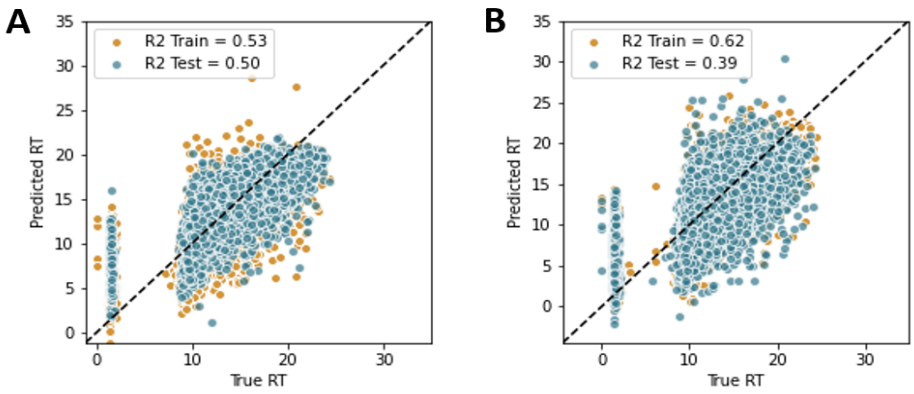

Figure 2: Parity Plots of Vanilla MPNN performance on clusters of molecules generated through (A) PCA on Morgan Fingerprints (model trained on molecules with triple bonds and carbonyl groups, and tested on molecules with sulfate groups), and (B) Kmeans of RDKit-generated scaffold splits (model trained on conjoined and 5-membered rings, and tested on 7-membered rings. Training curves are in Appendix B.

The uncertainties do range around 0.3 minutes, or about 20 seconds, which is consistent with our expectations for an uncertainty order of magnitude. Fig. 3C in particular seems to suggest higher uncertainties in the middle of the dataset (i.e. for molecules with retention times between 12-14 mins). However, this sheds little light on how accurate the uncertainty estimations are for each individual molecule.

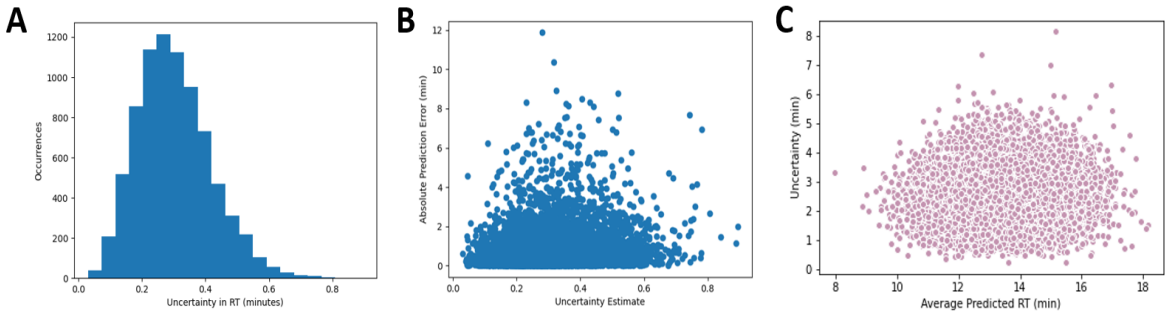

Figure 3: Uncertainty results for FFNN and vanilla MPNN. a) Histogram of Uncertainties using FFNN Ensemble. b) Prediction Error vs. Estimated Uncertainty from an ensemble of FFNNs. c) Uncertainty of Prediction vs. Mean Prediction for each point in the randomly split test set, using Vanilla MPNN ensemble.

We would expect that molecules with higher uncertainties contribute more strongly to the test error. A plot of the prediction error (difference between mean prediction and true retention time) vs. uncertainty estimate (Figure 3) does not demonstrate any such relationship. It appears that the uncertainty estimate and prediction error have no correlation.

For this reason, we cannot confirm the validity of our uncertainty estimates. We propose that a most valid uncertainty quantification would account for how different a molecule is from the training set. Distance-based methods are common for these tasks [10]. For example, the Tanimoto distance can be used to quantify how similar a molecule is to the set of training molecules, and the uncertainty shall be a function of the distance. However, in contrast to ensembling uncertainty quantification, such distance measurements do not give a simple deviation from the prediction and are instead relative values.

With this in mind, predicting the uncertainty associated with a sample is more challenging when a sample falls outside of the chemical space of the training set. In this case, relative prediction uncertainty can be determined by evaluating the chemical distance between a model's training set and a new set of molecules.

The Vanilla MPNN used in this work was trained on two different clustering strategies intended to separate the dataset in chemical space, as described in Methods, section (b). Figure 5 shows how the absolute prediction error on the test set changes with chemical Tanimoto distance for each splitting method used. The positive slope of the best-fit line indicates that prediction error has a small dependence on chemical distance. This suggests that chemical distances could be useful for uncertainty estimation for future work in this field, and provides a calibration to convert chemical distance to property uncertainty.

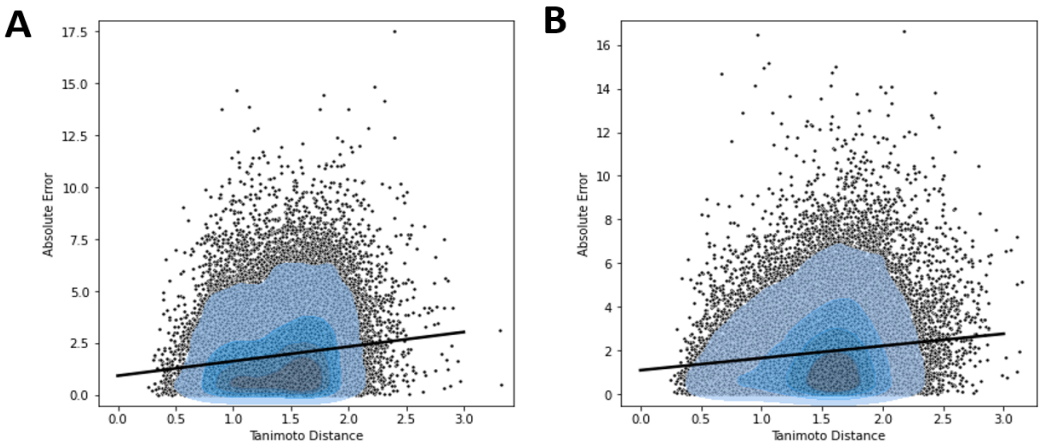

Figure 4: Absolute prediction error on test set over the mean log Tanimoto distance between test set molecules and the training set. Plot is shown for data splits of (A) PCA of Morgan Fingerprints, and (B) K-means of scaffold split. A linear best-fit line is shown.

Conclusion

In this project, we attempted to predict retention times using the SMRT METLIN database and corresponding uncertainties. A feed forward neural network and a complex feature-embedded MPNN (Chemprop) yielded the lowest median absolute error on the test set. The ensemble method, when applied to a FFNN trained on a random split, did not appear to yield meaningful uncertainty results. In contrast, the uncertainty estimates derived from ensembling a vanilla MPNN trained on scaffold split did appear to be dependent on the distance of a test molecule to the dataset. However, we are not sure whether this relationship is significant.

We only evaluated a small subset of model and uncertainty quantification pairings. Future studies may evaluate ensembles of other model architectures and other uncertainty quantification methods, such as union-based methods or mean variance estimated. This analysis could reveal whether different uncertainty quantification methods deem similar molecules more uncertain than others. Additionally, the percent of molecules in the training set whose true retention time is within the one deviation of the prediction may be a more useful metric to measure the effectiveness of an uncertainty quantification method. Ultimately, further analysis is necessary to identify an accurate uncertainty quantification method for this data set.

Contributions

Priyanka: Data cleanup and processing, featurization, and generation of random splits. Coding, optimization, training, and evaluation of linear and kNN baselines and Vanilla MPNN model. Evaluating Vanilla MPNN model on non-random splits and uncertainty analysis for this model using ensembling.

Jenna: FFNN coding, hyperparameter optimization, and ensembling. Uncertainty analysis on FFNN ensemble. Chemprop implementation. Attempted chemprop ensembling.

Nathan: Data analysis and clustering, generation of structure-based splits, uncertainty analysis with Tanimoto distance.

Code

Our code is publicly available at `https://github.com/priyanka-rag/Retention-Time-Prediction`, along with a short description of the code organization.

References

(1) Domingo-Almenara, X. et al. *Nature Communications* **2019**, *10*, 5811.
(2) Yang, Q. et al. *Analytical Chemistry* **2021**, *93*, 2200–2206.
(3) Fedorova, E. S. et al. *Journal of Chromatography A* **2022**, *1664*, 462792.
(4) Kensert, A. et al. *Analytical Chemistry* **2021**, *93*, 15633–15641.
(5) Soleimany, A. et al. Evidential Deep Learning for Guided Molecular Property Prediction and Discovery.
(6) Akiba, T. et al. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
(7) Heid, E. et al. *Journal of Chemical Information and Modeling* **2022**, *62*, 2101–2110.
(8) Meng, J. et al. *Scientific Data* **2022**, *9*, 71.
(9) Lenselink, E. B. et al. *Journal of Computer-Aided Molecular Design* **2021**, *35*, 901–909.
(10) Hirschfeld, L. et al. *Journal of Chemical Information and Modeling* **2020**, *60*, 3770–3780.

Appendix

a. Random Split Results

Hyperparameter optimization on the kNN regression model yielded 5 as the best number of neighbors.

The hyperparameter optimization on the feed-forward neural network yielded the following hyperparameters: Hidden dimensions: [1064, 437, 287, 143]; Learning Rate: 1.2e-4; Weight decay: 0.008; Number of Epochs: 15. While the number of epochs appears low, this serves as no surprise due to rapid overfitting to the training data.

Hyperparameters for the Vanilla MPNN (taken from the paper by Yang et al. [2]) are as follows: Number of Convolutions: 6; Embedding Dimension Size: 48; Learning Rate: 1.0e-4.

Model	MAE (min)	MedAE (min)	MRE	R^2
Linear	1.17	0.87	0.09	0.7
KNN	1.26	0.87	0.1	0.62
FFNN	0.73	0.52	0.056	0.86
Vanilla MPNN	1.1	0.77	0.08	0.7
Chemprop	0.69	0.43	0.054	0.84
GNN-RT [2]	0.67	0.42	0.05	0.85

Table 2: Metrics for all models using the random split evaluated on retained molecules only. Note that in comparison to Table 1, this includes models trained on the full training dataset, but only evaluated on the retained molecules in the test set. The GNN-RT (last row) is the current best performing model in the literature.

Note that our results in this table represent models trained on the full training dataset, but only evaluated on the retained molecules in the test set, as opposed to Table 1, which showed results evaluated on the full test set. The GNN-RT (last row) is the current best-performing model in the literature, but was only trained on the retained molecules, so it is not a direct comparison.

We did consider removing the non-retained molecules from the dataset before training and evaluation. However, we were unable to distinguish retained from non-retained molecules using principle component analysis. Therefore, we deem that arbitrarily removing non-retained molecules, which are not easily distinguishable from retained molecules, would be biasing the data.

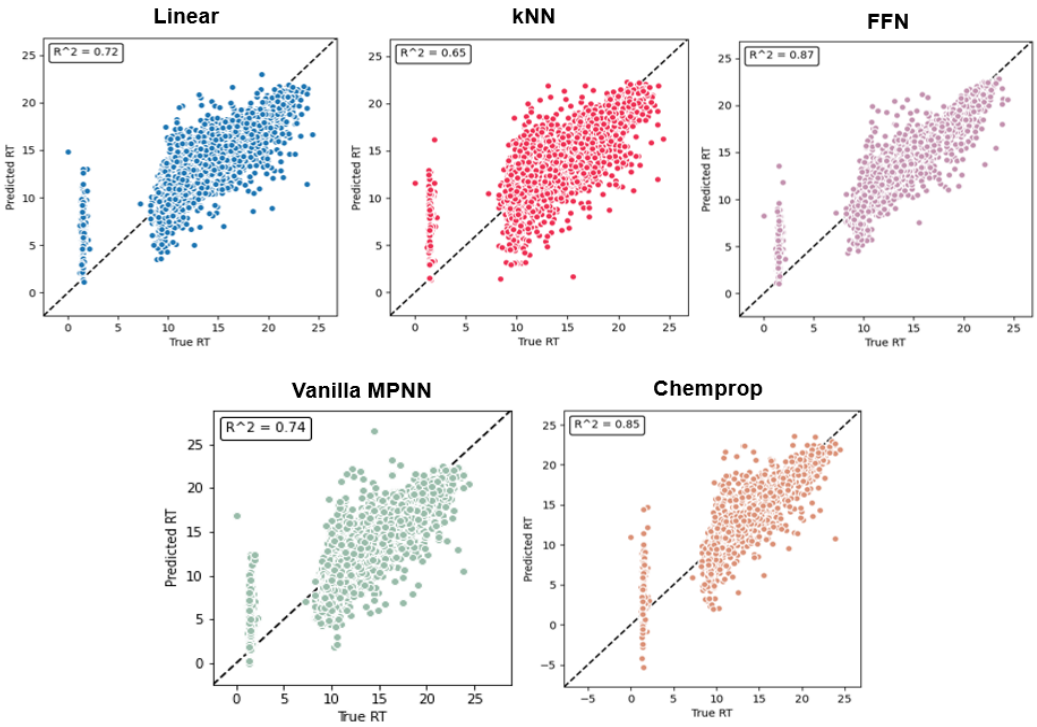

Figure 5: Parity Plots of all models' performance on the held-out test set generated from the random split. For more detailed metrics for each model, see Table 1.

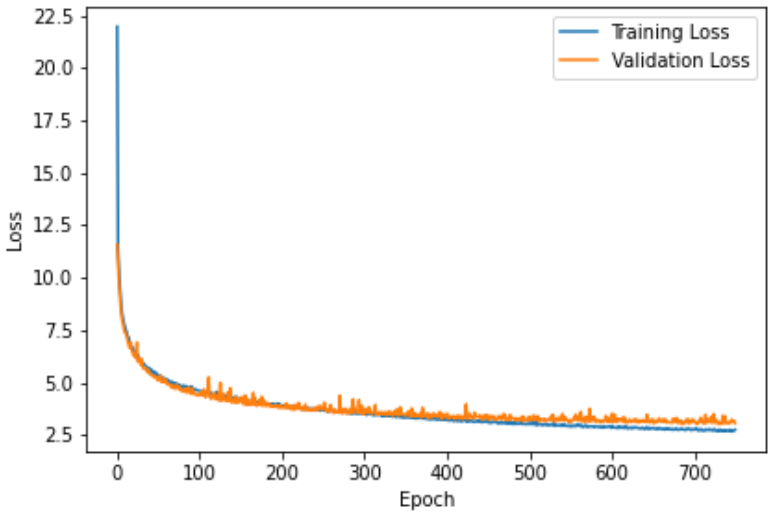

Figure 6: Training loss curves for the Vanilla MPNN (random split).

b. Non-Random Split Results

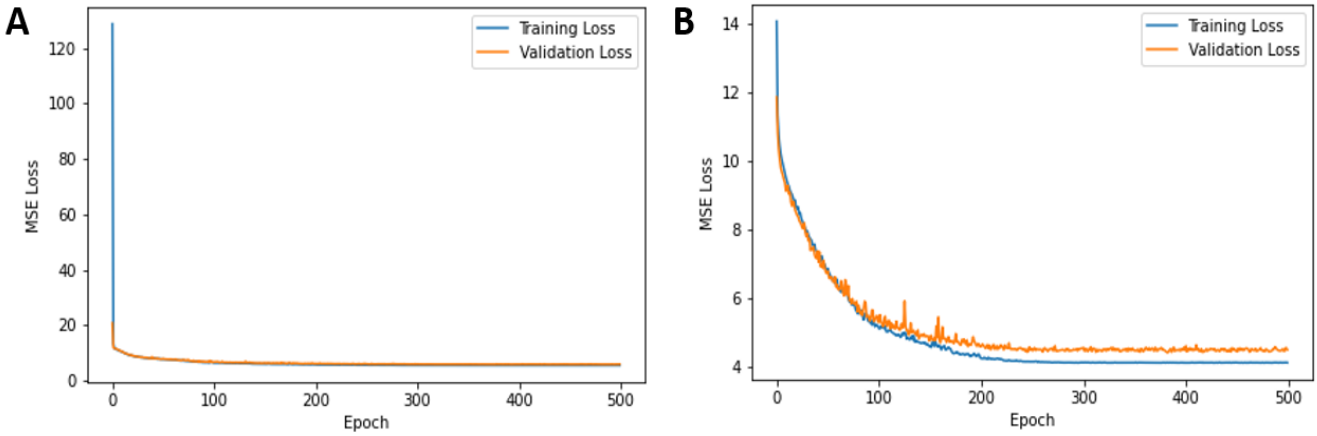

Figure 7: Training loss curves for the Vanilla MPNN for the splits mentioned in Methods, Section B: (A) PCA on Morgan Fingerprints and (B) Kmeans on Scaffold Split.

c. Uncertainty Analysis and Data Splitting

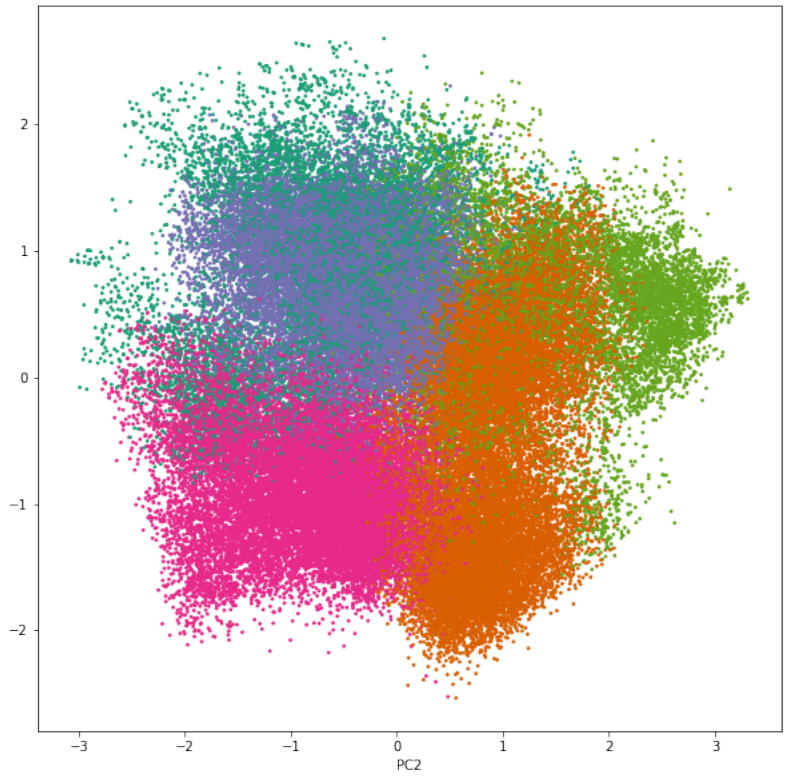

Figure 8: k-means clustering on 10-dimensional PCA of molecular fingerprints, visualized using the first two principal components. These clusters were used to generate structure-based splits for the MPNN.

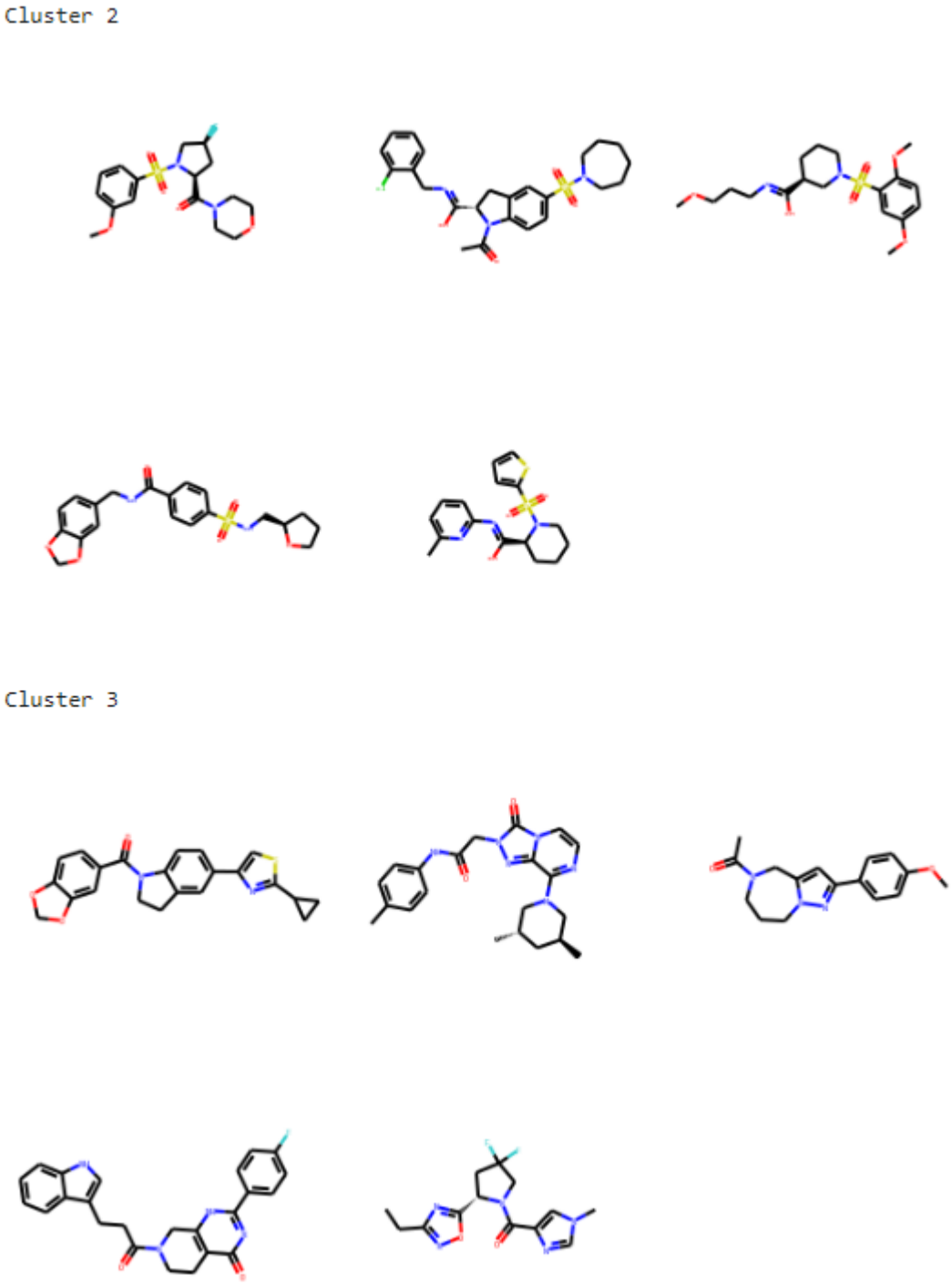

Figure 9: Example molecules showing the characteristic differences between molecules in different splits. Two example clusters are shown from the same splitting as shown in Figure 8. Cluster 2 molecules contain sulfate groups, while cluster 3 molecules all have carbonyl groups.