analytical chemistry

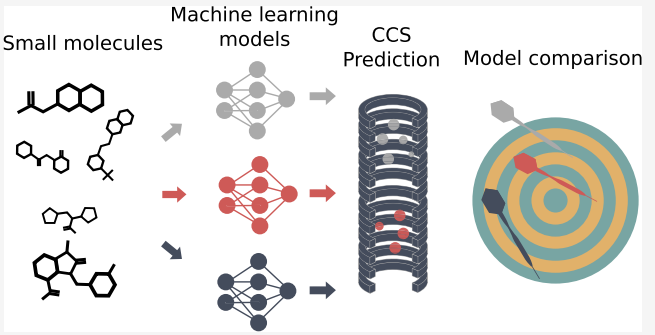
pubs.acs.org/ac
Article

Predicting the Predicted: A Comparison of Machine Learning-Based Collision Cross-Section Prediction Models for Small Molecules

Sara M. de Cripan, Trisha Arora, Adrià Olomí, Núria Canela, Gary Siuzdak, and Xavier Domingo-Almenara*

Cite This: *Anal. Chem.* 2024, 96, 9088−9096
Read Online

ABSTRACT: The application of machine learning (ML) to -omics research is growing at an exponential rate owing to the increasing availability of large amounts of data for model training. Specifically, in metabolomics, ML has enabled the prediction of tandem mass spectrometry and retention time data. More recently, due to the advent of ion mobility, new ML models have been introduced for collision cross-section (CCS) prediction, but those have been trained with different and relatively small data sets covering a few thousands of small molecules, which hampers their systematic comparison. Here, we compared four existing ML-based CCS prediction models and their capacity to predict CCS values using the recently introduced METLIN-CCS data set. We also compared them with simple linear models and with ML models that used fingerprints as regressors. We analyzed the role of structural diversity of the data on which the ML models are trained with and explored the practical application of these models for metabolite annotation using CCS values. Results showed a limited capability of the existing models to achieve the necessary accuracy to be adopted for routine metabolomics analysis. We showed that for a particular molecule, this accuracy could only be improved when models were trained with a large number of structurally similar counterparts. Therefore, we suggest that current annotation capabilities will only be significantly altered with models trained with heterogeneous data sets composed of large homogeneous hubs of structurally similar molecules to those being predicted.



INTRODUCTION

Ion mobility-mass spectrometry (IM-MS) is being increasingly adopted into conventional workflows to facilitate the analysis of lipids, metabolites, and other small molecules. Since the unambiguous annotation of small molecules continues to pose a challenge, IM-MS-generated collision cross-section (CCS), together with tandem MS (MS/MS) and/or retention time (RT) information, can be used as an additional approach to annotate or identify molecules in biological samples. CCS values are relatively robust under the same experimental conditions, providing a complementary, semi-orthogonal measure for small-molecule annotation, addressing some of the drawbacks of chromatographic columns such as poor isomer separation and high temporal and cross platform variance in RTs.[1,2] To identify molecules with IM-MS, the experimental CCS value for an observed molecule has to be compared with a reference value. Existing libraries and curated data sets containing reference CCS values continue to grow. However, construction of such libraries is a time-consuming and expensive process since they are often built from the analysis of commercially available standards. They are also likely to exclude a large number of molecules that could be observed in samples due to the unavailability of commercial standards.[3,4]

The advent of machine learning (ML) in recent years has sparked a broad interest in using this technology to bypass the analysis of commercially available standards and generate in-silico reference data, like MS/MS or RT, for metabolite annotation.[5] Different ML-based CCS prediction models and strategies have been recently introduced to address this gap. In-silico prediction of CCS values commonly relies on chemical first-principles[6] or data-driven approaches such as ML.[7] ML approaches tend to be faster and since ML-based predictions are heavily dependent on the quality and the structural diversity of the input data, there is a concerted effort to build CCS databases that can eventually be used for the robust prediction of CCS values.[8] Accordingly, different ML frameworks and tools have been implemented for the

Received: February 1, 2024
Revised: May 9, 2024
Accepted: May 10, 2024
Published: May 24, 2024

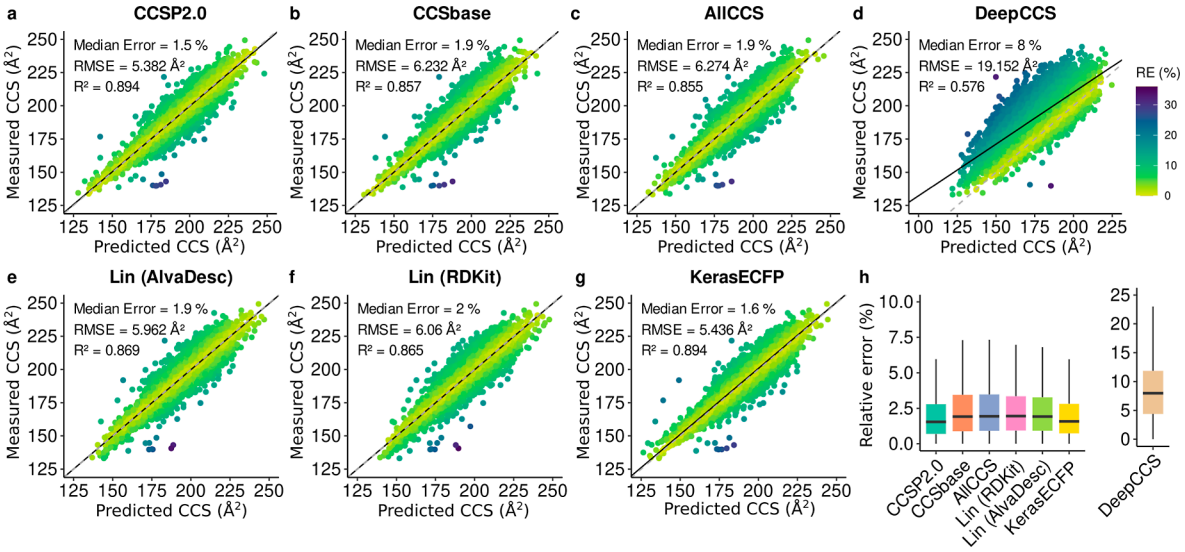


Figure 1. Scatterplots of predicted vs experimental CCS values of the models in the literature: (a) CCSP2.0, (b) CCSbase, (c) AllCCS, and (d) DeepCCS; linear (Lin) models using fingerprints generated with (e) alvaDesc and (f) RDKit; and (g) artificial neural network model generated with Keras using alvaDesc's fingerprints. (h) Relative prediction errors for each model.

prediction of CCS values, which are trained on curated experimental data sets. Two major components that differ in their implementation are the manner of representation of the molecular structure and the ML algorithm deployed. The molecular structures are commonly depicted in the form of molecular descriptors, fingerprints, or, more recently, features derived from graph network algorithms. Some models like DeepCCS[9] directly encode the Simplified Molecular Input Line Entry System (SMILES) as binary matrices for molecular representation. A combination or a selection of these features are then used as an input to train different ML paradigms, which have included support vector regression,[10] random forest,[7] and artificial neural networks.[9,11]

Some of these models have been published alongside small or relatively large CCS libraries, usually covering hundreds or a few thousands of molecules, with AllCCS being one of the largest libraries available so far.[12,13] The lack of large-scale libraries hampers our ability to critically assess the accuracy and the generalization capability of these models. In that regard, there are no independent studies that assess their performance with an independent and large-scale CCS data set. Recently, the METLIN CCS library has been introduced,[14] covering over 61,000 averaged CCS values for 27,633 standards for $[M + H]^+$, $[M - H]^-$, and $[M + Na]^+$ adducts and covering small molecules ranging from 140 to 662 Da with their corresponding CCS values between 121.48 and 277.46 Å^2.

Herein, we evaluated the accuracy and differences between a representative group of ML-based models for CCS prediction. We retrained four established CCS prediction models, namely, CCSP2.0,[15] CCSBase,[16] AllCCS,[13] and DeepCCS[9] with the METLIN-CCS database by mimicking their native architecture and input methods to gauge their performance and adaptability to new data sets. We further tested the use of linear modeling and molecular fingerprints for CCS prediction. We analyzed how the training and test set molecular similarity impacts the model accuracy to assess whether this similarity can be used as a metric for the reliability of the prediction. Eventually, a neural network architecture is compared with existing models, and the efficacy of different ML architectures for practical metabolite annotation is considered.

EXPERIMENTAL SECTION

CCS Prediction Model in the Literature. AllCCS, CCSbase, DeepCCS, and CCSP2.0 training models were trained with the METLIN-CCS database. After removing a total of 3,822 CCS values corresponding to molecular dimers in METLIN-CCS, the remaining 58,041 CCS values were split into a training and a validation set containing 80 and 20% of the data, respectively, except in the case of DeepCCS, wherein molecules with unseen SMILES representation that could not be predicted using the original model were excluded. Therefore a subset of 28,952 and 7,196 molecules as the training and validation set was used for DeepCCS. The number of molecules used for making the individual models is detailed in Table S1. To showcase their generalization performance, their hyperparameters were not reoptimized for the METLIN-CCS database, except for the case of CCSP2.0, as the model was designed to automatically find the best hyperparameters in the training phase. Specific details of each model's architecture and implementation for this study are described in Supporting Information.

Linear and Deep Learning Model Optimization. Extended-connectivity fingerprints (ECFPs) with radius = 2 and 1024 bits were generated for molecules using the Python package RDKit[17] and alvaDesc[18] software from their SMILES representation.

A deep learning model, KerasECFP, was made using alvaDesc FPs, m/z, and one hot encoded adduct information as input. The model was constructed using Keras Sequential API in Python (version 3.11.3), which consisted of three fully connected dense layers, consisting of 1028 (equivalent to the number of input columns), 512, and 32 units, respectively, with ReLU activation. The output layer consisted of one unit with linear activation. The Adam optimizer was used with an adjusted learning rate of 0.0001. The batch size was optimized using 10-fold cross validation considering only 32, 64, and 128 size numbers. The final model was trained for 75 epochs with a

https://doi.org/10.1021/acs.analchem.4c00630
Anal. Chem. 2024, 96, 9088−9096

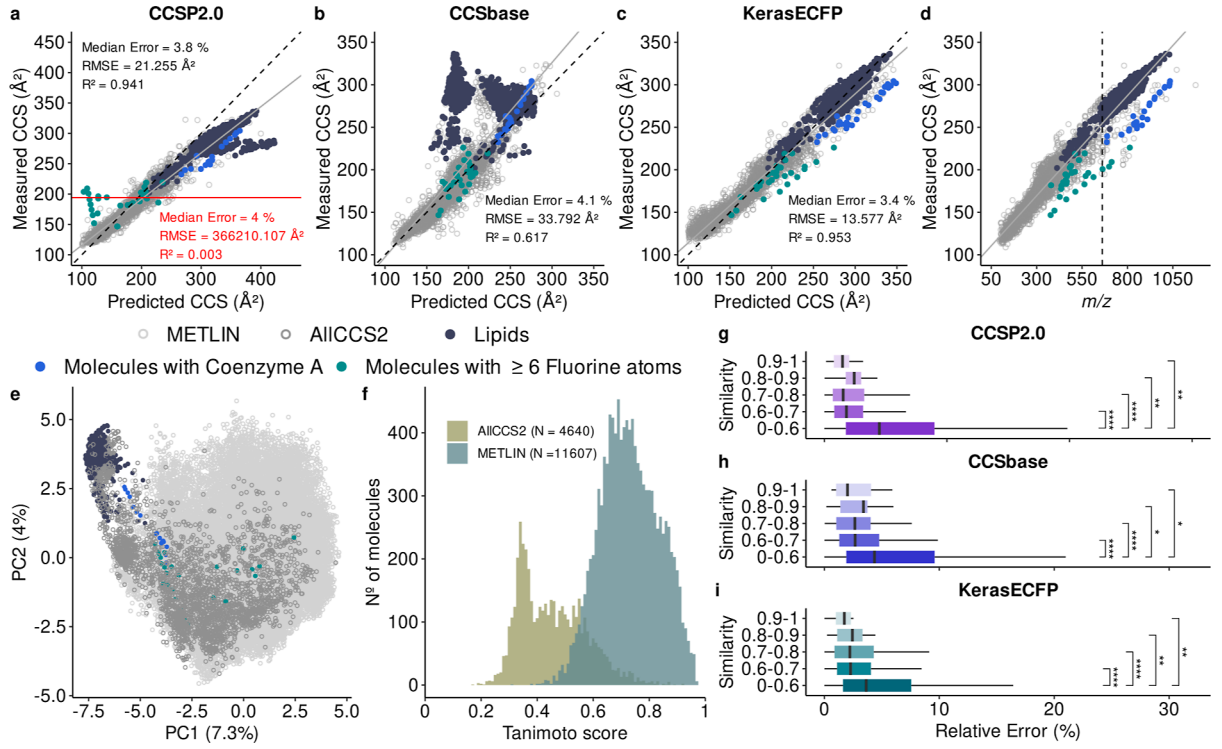


Figure 2. Scatterplots of predicted vs reference CCS values from the AllCCS2 library using (a) CCSP2.0, (b) CCSBase, or (c) KerasECFP models trained with METLIN-CCS data. Outliers (relative error >50%) are not shown. The dashed black line is the identity function, the gray line is the regression line calculated with predicted CCS values with relative errors below 50%, whereas the red line is the regression line calculated with the complete set of predicted CCS values. Specific lipid classes, coenzyme-A motifs, and multi-fluorine (>6 fluorine atoms) within the AllCCS2 database are shown in color. (d) m/z vs CCS values in the AllCCS2 library. The vertical dashed line indicates the value of the maximum m/z in METLIN-CCS. (e) PCA projecting all METLIN-CCS (light gray) and AllCCS2 (dark gray) molecules' fingerprints into a two-dimensional space. (f) Frequency distributions of the Tanimoto similarity between AllCCS2 and METLIN-CCS training set (N = 4,640), and between METLIN-CCS validation and training set (N = 11,607). (g–i) Boxplots representing the relative errors according to similarity ranges for AllCCS2-predicted molecules with (g) CCSP2.0, (h) CCSbase, and (i) KerasECFP trained with METLIN-CCS, with significances calculated with a Wilcoxon test.

batch size of 128 and using the mean squared error as a loss function.

RESULTS AND DISCUSSION

Comparison of Established ML-Based Models for CCS Prediction. We trained four established ML-based models for CCS prediction: CCSP2.0, CCSBase, AllCCS, and DeepCCS, using the recently published METLIN-CCS library. Prediction results for each model are summarized in Figure 1. Results show similar prediction performance for CCSP2.0, CCSbase, and AllCCS, while DeepCCS showed a significantly higher relative error compared to the other models in all pairwise comparisons (Wilcoxon test, p-value < 0.0001, Figure 1a–d and Table S5). CCSbase and AllCCS showed very similar performances with a 1.9% median relative error with no statistically significant differences between them. CCSP2.0 prediction performance was statistically different compared to that of the rest of the models (p-value < 0.0001), with a median relative error of 1.5%. This can be attributed to the fact that CCSP2.0 was designed to automatically find the best hyperparameters in the training phase. The low performance of DeepCCS can be attributed to the low sample size that was used to develop the model and the fact that DeepCCS encodes molecules using a one-hot vector encoding of the SMILES representation, as opposed to molecular descriptors used by CCSP2.0, CCSbase, and AllCCS. All four methods showed adduct-specific differences in performance: for CCSP2.0, CCSbase, and AllCCS predictions, $[M + H]^+$adducts showed higher prediction accuracy compared to the predicted CCS values for $[M - H]^-$ and $[M + Na]^+$ adducts (p-value < 0.0001). CCSbase and AllCCS also showed statistically significant prediction errors between $[M + Na]^+$ (p-value < 0.001) and $[M - H]^-$ (p-value < 0.01) adducts. This can potentially be attributed to the higher amount of data for $[M + H]^+$ adducts (43%) in METLIN-CCS relative to $[M + Na]^+$ (26%) and $[M - H]^-$ (31%) adducts.

As the aim of this study is to test the original models, the hyperparameters of CCSBase, AllCCS, and DeepCCS models were not specifically optimized for METLIN-CCS, as opposed to CCSP2.0, which was designed to automatically find the best hyperparameters in the training phase. However, further hyperparameter optimization along with molecular descriptor selection was performed for CCSBase and AllCCS to assess the changes in prediction performance. Table S1 and Figure S2 show the prediction results after hyperparameter optimization. Results show that, generally, a specific optimization did not lead to a significant improvement in accuracy, and in some cases, the accuracy decreased. In addition, the optimization was highly demanding in terms of computational resources. These observations suggest that the hyperparameter optimization with a larger data set (METLIN-CCS) does not improve the AllCCS and CCSBase capacity to learn from new latent variables that were not initially represented in or learned from a smaller but already rich data set (AllCCS).

https://doi.org/10.1021/acs.analchem.4c00630
Anal. Chem. 2024, 96, 9088−9096

Comparison of Established CCS Prediction Methods with Linear Models and the Use of Fingerprints. We examined the utility of linear models for CCS prediction and as baselines for comparison, as opposed to the use of advanced ML algorithms used in the existing literature. We additionally explored the use of fingerprints, a widely used alternative in chemical ML-based applications, which encode the molecular structure as binary vectors[19] as input features for the linear models. In contrast, CCSP2.0, AllCCS, and CCSBase examined in the previous section use molecular descriptors, while DeepCCS encodes the SMILES into binary matrices to be used as input features. We trained two linear models using two types of fingerprints as regressors, alvaDesc[18] and RDKit[17] extended-connectivity fingerprints (ECFP, radius = 2, 1024 bit-strings),[20] in addition to m/z values, and one-hot encoded adduct information (see Supporting Information for details). Both linear models showed the median relative errors of 1.92% (LinECFP—AlvaDesc) and 1.95% (LinECFP—RDKit), which, surprisingly, implies that their performance was at par with that of AllCCS (2.08%) and CCSBase (1.91%) (Figures 1e,f,h and S1), both of which used advanced ML algorithms. Importantly, linear models do not need to be optimized, which is also a clear advantage over advanced ML algorithms and provides a benchmark for optimizing subsequent regression models. Given that among the linear models tested, alvaDesc fingerprints showed the lowest error, they were further selected for optimization and performance improvement with artificial neural networks using Keras. The KerasECFP model in turn showed a median relative error of 1.58%, which was comparable to 1.54% of CCSP2.0, with no significant differences between them. Keras ECFP showed significantly better performance (Wilcoxon test, p-value < 0.0001) in pairwise comparisons with AllCCS, CCSBase, and DeepCCS (Figure 1g,h and Table S5).

Assessment of the Generalization Performance. The aim of prediction models is to predict values for molecules previously unseen by the model (i.e., lacking reference data in libraries). Known as generalization capability, the ability of a model to extrapolate to new and different data dictates the utility of ML models for real-world applications like metabolite annotation. As the training and test sets are derived from the same data set and are composed of a similar population, they tend to share similarities (e.g., groups of similar molecules), which might introduce a bias that precludes getting a good generalization assessment. To further assess the generalization capability of the models, we compared the prediction accuracy of CCSP2.0, CCSBase, and KerasECFP models trained with the METLIN-CCS database when predicting the molecules of the AllCCS2 library.[12] AllCCS and DeepCCS models were not compared as AllCCS showed a similar performance to CCSBase and DeepCCS showed a poorer performance compared to the rest of the models.

Prediction results are shown in Figure 2a−c. From the figure, three interesting observations can be made. First, both CCSBase and AllCCS models yielded a median relative error of around 4%, while KerasECFP showed an error of 3.4%. These errors are significantly greater than when the models were trained and tested with the same data set (METLIN-CCS). A comparable analysis was made elsewhere,[12] where the original AllCCS and CCSBase models were used to predict a subset of the AllCCS2 library, reporting median relative errors of 2.09% (AllCCS) and 2.03% (CCSBase). Second, CCSP2.0 and CCSbase showed a poor performance at predicting molecules having an m/z outside the range of the m/z of molecules in the training set (Figure 2d). In fact, even when molecules with an m/z out of the METLIN-CCS's m/z range were removed from the validation set, the yielded median relative errors were 3.05% (CCSP2.0) and 3.43% (CCSBase), which are still greater (around twice the error) than when the models were trained and tested with the same data set (Figure 2a−c). The third observation that can be made is that the models showed distinctive differences when predicting lipid-like molecules, molecules with coenzyme A motif (CoA), and molecules with more than six fluorine atoms (F) (Figures 2a−c and S1), even in cases when these molecules fell within the m/z range of the training set (METLIN-CCS) and where CCSP2.0 yielded prediction errors above 50% in some cases (Figure S1). In contrast, the KerasECFP model showed a much better generalization capability at predicting these molecules, which might suggest that the use of fingerprints as regressors is a better alternative to the use of molecular descriptors. A more detailed discussion about the model's differences is included in Supporting Information. This low performance for lipid-like molecules can be partly explained by the low coverage of lipids in METLIN-CCS. Globally, the differences in the chemical space spanned by METLIN-CCS and AllCCS2 can be observed from a principal component analysis (PCA) projection of the fingerprints (Figure 2e). In the figure, we can also observe that lipids in the AllCCS2 data set that yielded large errors when predicted using the model trained with METLIN-CCS are grouped in a cluster that segregates from the main METLIN-CCS molecular space (Figure 2e). To further assess the similarities between data sets, we calculated the structural similarity among molecules using the Tanimoto similarity index,[21] which measures how similar the two-dimensional structures of two molecules are, and it ranges from 0 (no similarity) to 1 (identical molecules). Figure 2f shows the similarity distributions between AllCCS2 molecules and their most similar counterpart in METLIN-CCS compared to the similarity distribution between the METLIN-CCS validation set and the most similar counterpart in the METLIN-CCS training set. Some overlap is observed in the range of 0.6−0.7 in the otherwise distinctly different distributions.

Taken together, these observations suggest that existing models suffer from a significant lack of generalization capability and the chemical and structural diversity of the training data dictates the subsequent performance accuracy.

Impact of Structural Similarity for Prediction Accuracy. It has been shown that the prediction performance depends on the structural similarity[5,12,13,22] and that when the predicted molecule is similar to those used for training (previously seen by the model), the error is lower compared to that when the model has not been trained with any similar counterpart. For the previous case (prediction of AllCCS2 data using the METLIN-trained model), this fact can be seen in Figure 2g−i, where prediction errors of CCSP2.0, CCSBase, and KerasECFP for molecules having at least one highly similar counterpart in the training set are significantly lower compared to the errors for molecules lacking a similar counterpart in the training set.

Efforts have been made to create a score to determine whether the prediction of a molecule will fall within a specific error range by considering the structural similarity between the predicted molecule with the molecules in the training set.[13,22,23] For instance, Zhou et al.[13] suggested that a

https://doi.org/10.1021/acs.analchem.4c00630
Anal. Chem. 2024, 96, 9088−9096

prediction score for a given molecule can be determined by averaging the similarity of the top 5 most similar molecules in the training set to the predicted molecule. In this respect, if this top-5 average similarity is high, the prediction error is likely to be lower compared to lower top-5 similarities. However, although these scores can discriminate between prediction error ranges, they cannot statistically ascertain that the prediction error will fall within a practical error range that allows these scores to be adopted for routine analysis.

In light of these observations, we sought to determine the minimum number of molecules above a specific similarity in the training set needed to statistically determine the likelihood of achieving a low prediction error for a molecule using a false discovery rate (FDR) approach. We focused on the results of the CCSP2.0, CCSBase, and KerasECFP as they constituted the top-performing models. We determined the FDR as follows: first, we filtered the molecules in the test set of each model by retaining only those with more than a specific number (from 1 to 15) of similar counterparts in the training set above a given similarity threshold. Next, each resulting molecule was classified as true positive (TP) or false positive (FP) if its prediction error was below or above a specific threshold, respectively. We calculated the FDR using 0.6, 0.7, and 0.8 similarity thresholds and using specific error thresholds corresponding to the median and third quartile (Q3) of the relative prediction errors in each model's case. FDR was calculated only if the number of TPs in addition to the number of FPs was at least 10. The generated mean FDR across all methods is shown in Figure 3. As observed from the figure, there is a high likelihood (FDR < 0.05) that the actual prediction error is lower than 3% (mean of the Q3 error across models) if the predicted molecule has at least eight similar counterparts in the training set above a 0.8 similarity. These

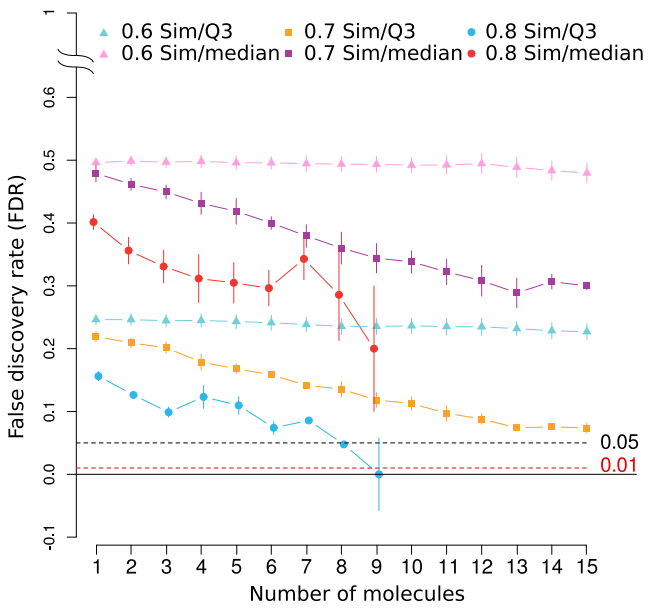


Figure 3. FDR represents the likelihood to achieve a prediction error lower than the value corresponding to the median relative error or the third quartile (Q3) of the CCSP2.0, CCSBase, or KerasECFP models as a function of the number of molecules in the training set above a specific similarity threshold to the predicted molecules: 0.6, 0.7, and 0.8 similarity (Sim) thresholds. FDR values correspond to the average values among the three models, and vertical lines represent the standard deviation among models. Only $N > 10$ cases are shown.

results also suggest that more than 15 similar counterparts above 0.7 similarity are needed to achieve at least an FDR of 0.05 for the same error, but the lack of data hampered the determination of the exact minimum number. The lack of data also prevented the determination of the minimum number of molecules to achieve an error below 1.68% (mean of the median relative errors across models). Overall, these results reiterate the impact of the structural similarity in the prediction performance and suggest that models need to be trained with a relatively large number of structurally similar counterparts to achieve low prediction errors.

Annotation Performance Based on Predicted CCS Values. Multiple putative metabolite identities can match to an observed protonated/deprotonated ion peak in MS^1 data via accurate mass search, and CCS values can be used to narrow down the list of potential identities. We aimed at comparing the ability to annotate metabolites based on predicted CCS values. To leverage the large-scale information provided by the METLIN-CCS database, we simulated the situation where multiple putative candidates match to the m/z of an observed unknown ion peak by considering each metabolite in the METLIN-CCS (test set) as an unknown metabolite but with a known m/z and CCS value, i.e., METLIN-CCS's reference values were used as a proxy of experimental data in real cases.

To that end, prediction results from CCSBase, CCSP2.0, AllCCS, DeepCCS, LinECFP (AlvaDesc), and KerasECFP were used, in addition to a simple linear regression of m/z against reference CCS values, which is equivalent to a random classifier where the correct identity among all possible identities is selected randomly. For each metabolite in the test set, we searched for those metabolites with the same m/z (within a 10 ppm error) also in the same test set, considering each protonated, deprotonated, and sodiated species separately. Next, we determined their predicted-reference CCS errors for all the methods using the reference CCS value in the METLIN-CCS library. We evaluated the performance of the different models at ranking putative identities based on their predicted-reference CCS error, compared to the ranking using reference values. We only considered these cases where there were at least four metabolite candidates that matched to the original metabolite, which resulted in 743 ($[M + H]^+$), 366 ($[M - H]^-$), and 386 ($[M + Na]^+$) cases.

Ranking results are shown in Figure 4a. The chart shows the percentage of cases in which the correct identity was ranked as the first, second, or third hit or not within the top-3 hits (>4), based on the predicted-reference CCS error. For comparative purposes, the figure also includes the results of the linear m/z model, which randomly classified the ranking position of each candidate in each case. As expected from our previous observations, CCSP2.0 and KerasECFP exhibited the best ranking capability, whereas CCSBase, AllCCS, and LinECFP showed similar trends. CCSP2.0 and KerasECFP showed the best top-1 ranking capability; CCSBase, AllCCS, and LinECFP models showed a similar top-1 ranking, but the LinECFP model showed a slightly superior top-2 performance, and overall, both CCSP2.0 and KerasECFP showed the best performance at ranking the correct identity among the top-3 candidates. To correctly interpret these results, it is important to compare the performance with that of the random classifier. The random classifier yielded a top-3 performance of approximately 75%, equivalent to the performance of DeepCCS but not very different from that of the other

https://doi.org/10.1021/acs.analchem.4c00630
Anal. Chem. 2024, 96, 9088−9096

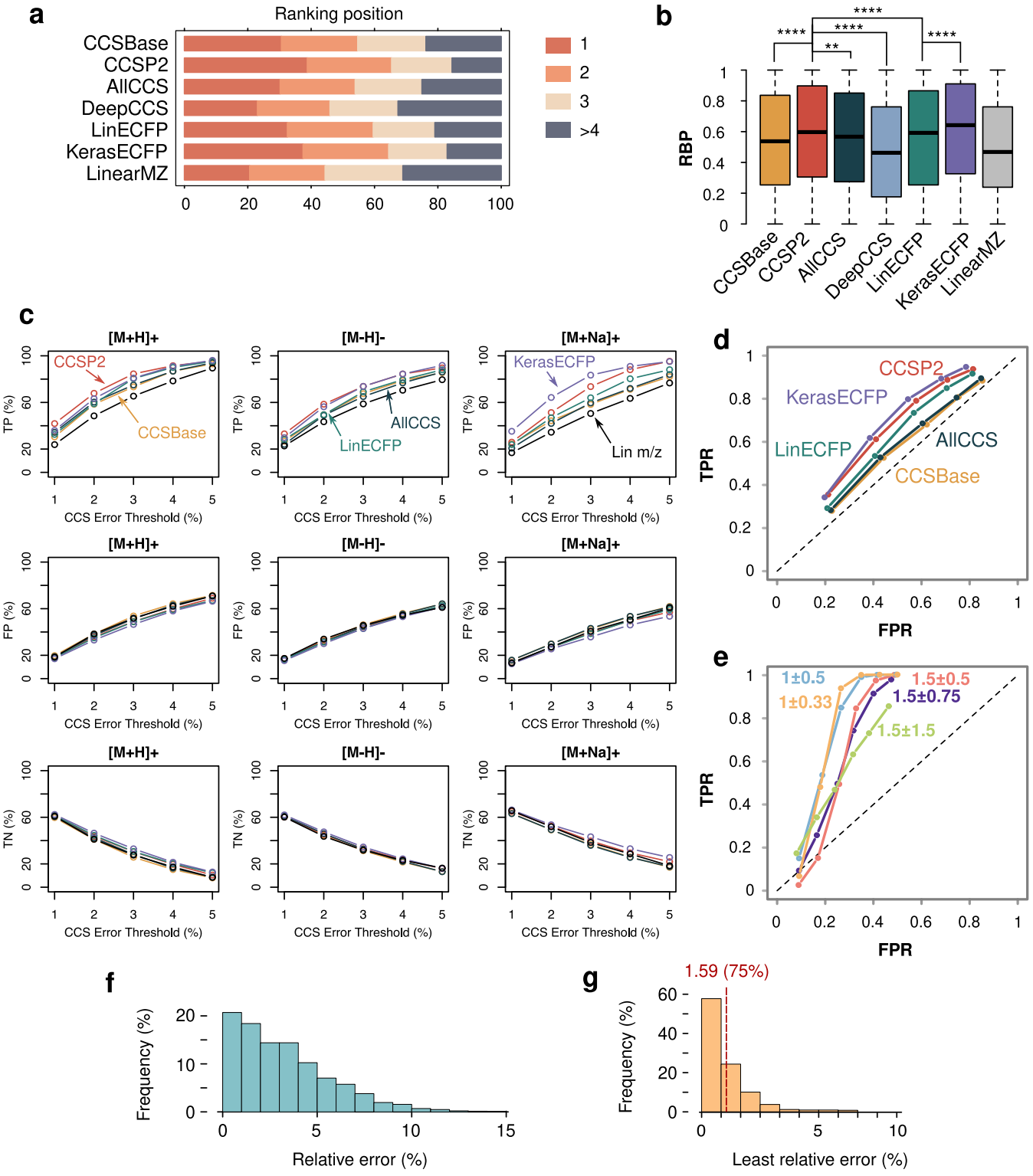


Figure 4. (a) Capability (percentage) at ranking the true metabolite candidate in the first, second, third, or > fourth position for all methods; (b) RBP-based ranking values, where statistical significances (Wilcoxon test) are shown for a set of representative comparisons; (c) TP, FP, and true negative (TN) rates (percentage) for all methods and adduct types; (d) receiver operating characteristic (ROC) curves for all methods, where dots represent the 1−5% thresholds and the dashed line represents the performance of a random classifier; (e) ROC curves for simulated data, where dots represent thresholds ranging from 0.5 to 3% with 0.5% steps and the text indicates the mean and standard deviation of the simulated data; (f) distribution of the relative CCS error among molecules with the same m/z (10 ppm); and (g) distribution of the least relative CCS error between any pair of molecules with the same m/z (10 ppm).

methods. This similarity can be attributed to the fact that in most of the cases, the number of candidates per m/z are 4 (the minimum number allowed in our comparison). Therefore, the random classifier classifies the first hit as first, second, third, and fourth in approximately 25% of the cases each. This means that in cases where the number of candidates per m/z are 4, a

https://doi.org/10.1021/acs.analchem.4c00630
Anal. Chem. 2024, 96, 9088−9096

random classifier will have a top-3 performance of 75%. Size limitations of our data preclude the comparison of the methods in a larger number of cases where the number of hits per m/z is above 4.

To further compare the ranking performance, we used an adaptation of the rank-biased precision (RBP) algorithm (see Supporting Information), commonly used to assess the ranking precision of ranking algorithms in web search engines. Our adapted RBP version yields a normalized score, from 0 to 1, giving a score of 1 when the ranking is perfect and 0 where the ranking is the worst possible, by weighting the importance of the hit order; i.e., the score drops more significantly if the order changes from [1,2,3,4] to [2,1,3,4] compared to from [1,2,3,4] to [1,2,4,3]. RBP-based ranking scores across prediction methods are shown in Figure 4 b, where statistically significant differences are observed between CCSBase, AllCCS, DeepCCS, and LinECFP compared to CCSP2.0 (Wilcoxon test) and also compared to KerasECFP (p-value < 0.0001, significance not shown in the Figure 4). These observations also support the previous findings, where CCSP2.0 and KerasECFP yield the best performance compared to the other methods.

Next, we evaluated the different model performance at filtering the true identity from all matches, i.e., retaining as many TP identities while filtering out as many FP identities as possible. We aimed to filter FP by removing all matches with a predicted-experimental CCS error above a 1, 2, 3, 4, and 5% threshold. When applied, this filtering will yield a TP if the true candidate is retained; true negatives (TNs) if the false candidates are removed; a false negative (FN) if the true candidate is removed; and FP if the false candidates are retained. Figure 4c shows the results of the TP, FP, and TN rates (in percentage) for the different methods and adducts, excluding the DeepCCS given that this method showed a poorer performance compared to the rest. FNs are not shown since they are equivalent to 1-TP because there is always a single true identity. From the figure, we can appreciate some differences on the TP rate, where CCSP2.0 and KerasECFP again show the best performance. Interestingly, all the methods, including the linear m/z method, show almost the same performance at filtering FPs and discarding TNs, which implies that their performance is equivalent to a random classification.

Using these values, we also determined the sensitivity or TP rate (TPR) and the specificity or false positive rate (FPR) for all the methods, excluding DeepCCS. This allowed the generation of a ROC curve that depicts the performance of the range of filtering thresholds at discriminating TP, FN, FP, and TN identities. Figure 4d shows the ROC curves for the different methods. Areas under the curve were 0.54 (CCSBase), 0.63 (CCSP2.0), 0.56 (AllCCS), 0.64 (KerasECFP), and 0.59 (LinECFP). The ROC curves also indicate that all methods have a comparable performance, although CCSP2.0 and KerasECFP exhibited a superior performance. The ROC curve also suggests that a 2% threshold is the optimal filtering threshold to retain as many TP identities as possible while filtering out as many FP identities as possible. Yet, this 2% cutoff will filter more than 30% of true candidates and will retain more than 40% of false candidates. This observed poor filtering performance is aligned with previous observations.[24] In light of these results, we sought to analyze the effect of this filtering threshold when using experimental CCS values. Figure 4f shows the histogram of the relative error among molecules with the same m/z (10 ppm), whereas Figure 4g shows the distribution of the least relative error between any pair of molecules with the same m/z. From Figure 4g, we can conclude that to filter all the FPs in 75% of cases where two or more molecules match to a specific m/z, we need to reach an overall prediction accuracy below 1.58%. This implies that almost all predicted values should have a prediction accuracy below 1.58%. Interestingly, although the median error of CCSP2.0 or KerasECFP have a median error of around 1.5%, the standard deviation of their error yields an accuracy where only around 50% of the molecules have a prediction accuracy below 1.5%. This implies that the standard error deviation of the prediction needs to be comparatively lower. To demonstrate that, we simulated a CCS prediction algorithm with random accuracies including 1 and 1.5% mean errors and standard deviations equal to the same, half, and a third of those errors and determined the ROC curves over different error thresholds ranging from 0.5 to 3% with 0.5% steps. Figure 4e shows the ROC curve for these different simulated accuracies, showing that to reach a desirable performance (around 80% of TPR and 30% of FPR), we would need at least an accuracy of 1.5% ± 0.5 (84% TPR and 33% of FPR with a 2% filtering threshold), although an accuracy of 1.5% ± 0.75 would yield a drop in TPR while the FPR would remain similar (74% of TPR and 31% of FPR with a 2% filtering threshold).

CONCLUSIONS

We performed a comparative analysis on the performance of four existing CCS prediction methods, CCSP2.0, CCSBase, AllCCS, and DeepCCS by retraining them on the recently introduced METLIN-CCS data set. Next, we compared these advanced algorithms with simpler alternatives based on linear regression and using fingerprints as opposed to chemical descriptors. While all models showed comparable performances, our results suggest that the development of tools that offer the possibility to retrain on individual data sets by an automatic optimization such as the workflow provided by CCSP2.0 have a greater adaptability and at the same time cater to part of the community unfamiliar to ML. However, we found that linear methods, which bypass the need for both model optimization and ML expertise, yielded comparable performances and can be used as robust baseline for building CCS prediction models.

We assessed the generalization performance of the models, i.e., their accuracy at predicting previously unseen data by the model. To that end, we trained CCSP2.0 and CCSBase using the METLIN-CCS data set while using the AllCCS2 data set for external validation. Results showed a poor generalization performance of the models, with significantly higher relative errors compared to when the models are validated with the same data sets. Because of the importance of structural similarity between molecules in the training set and the molecule being predicted, we observed that if we want to be certain (FDR < 0.05) that the accuracy of the predicted error of a given molecule is below 3%, we need to train the model with at least eight structurally similar counterparts above a 0.8 similarity. These observations raise an important concern as we later found that an error of 3% has a limited practical use for routine metabolite annotation. These results demonstrated the need to consolidate existing databases to further improve CCS prediction accuracies.

https://doi.org/10.1021/acs.analchem.4c00630
Anal. Chem. 2024, 96, 9088−9096

Finally, we explored the addition of predicted CCS information for filtering and scoring putative annotations and removing false positives, using only CCS and m/z information. We found that while a 2% error was the optimal filtering threshold, this 2% cutoff will filter more than 30% of TP candidates and will retain more than 40% of FP candidates. In fact, results showed that the models' performance at filtering FP metabolite identities based on predicted CCS values is equivalent to a random classification. This implies that although predicted CCS values may help in the ranking of the potential identity candidates, it does not necessarily translate to a better capacity to decrease the number of FP candidates. In the current scenario, multidimensional matching in addition to RT and MS/MS information is required to decrease the number of potential candidates.

Taken together, our results cumulatively reiterate the dependence of the prediction reliability on the structural similarity between training data and the molecule for which the CCS values are being predicted. This implies that training data sets should be heterogeneous yet have a high representation of the structures similar to the molecule being predicted. Finally, given the similarity in the performance of the existing models, we propose that new models should be compared with existing models using and optimized with the same data.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.analchem.4c00630.

Additional data comprising ML-based algorithms' performance metrics, additional descriptions about ML-based algorithm designs, specific results on the prediction of AllCCS2 molecules, and definition of rank-biased precision (RBP) calculation (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Xavier Domingo-Almenara − *Computational Metabolomics for Systems Biology Lab, Eurecat—Technology Centre of Catalonia, Barcelona 08005 Catalonia, Spain; Centre for Omics Sciences (COS), Unique Scientific and Technical Infrastructures (ICTS), Eurecat—Technology Centre of Catalonia & Rovira i Virgili University Joint Unit, Reus 43204 Catalonia, Spain; Department of Electrical, Electronic and Control Engineering (DEEEA), Universitat Rovira i Virgili, Tarragona 43007 Catalonia, Spain;* orcid.org/0000-0002-0133-6863; Email: xavier.domingoa@eurecat.org

Authors

Sara M. de Cripan − *Computational Metabolomics for Systems Biology Lab, Eurecat—Technology Centre of Catalonia, Barcelona 08005 Catalonia, Spain; Centre for Omics Sciences (COS), Unique Scientific and Technical Infrastructures (ICTS), Eurecat—Technology Centre of Catalonia & Rovira i Virgili University Joint Unit, Reus 43204 Catalonia, Spain; Department of Electrical, Electronic and Control Engineering (DEEEA), Universitat Rovira i Virgili, Tarragona 43007 Catalonia, Spain;* orcid.org/0000-0002-2565-7864

Trisha Arora − *Computational Metabolomics for Systems Biology Lab, Eurecat—Technology Centre of Catalonia, Barcelona 08005 Catalonia, Spain; Centre for Omics Sciences (COS), Unique Scientific and Technical Infrastructures (ICTS), Eurecat—Technology Centre of Catalonia & Rovira i Virgili University Joint Unit, Reus 43204 Catalonia, Spain; Department of Electrical, Electronic and Control Engineering (DEEEA), Universitat Rovira i Virgili, Tarragona 43007 Catalonia, Spain*

Adrià Olomí − *Computational Metabolomics for Systems Biology Lab, Eurecat—Technology Centre of Catalonia, Barcelona 08005 Catalonia, Spain; Centre for Omics Sciences (COS), Unique Scientific and Technical Infrastructures (ICTS), Eurecat—Technology Centre of Catalonia & Rovira i Virgili University Joint Unit, Reus 43204 Catalonia, Spain*

Núria Canela − *Centre for Omics Sciences (COS), Unique Scientific and Technical Infrastructures (ICTS), Eurecat—Technology Centre of Catalonia & Rovira i Virgili University Joint Unit, Reus 43204 Catalonia, Spain;* orcid.org/0000-0003-0261-2396

Gary Siuzdak − *Scripps Center of Metabolomics and Mass Spectrometry, Department of Chemistry, Molecular and Computational Biology, Scripps Research Institute, La Jolla, California 92037, United States;* orcid.org/0000-0002-4749-0014

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.analchem.4c00630

Author Contributions

S.M.d.C. and T.A. contributed equally. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was partially funded by the Spanish State Research Agency (AEI/10.13039/501100011033) grant PID2019-106277RA-I00 (X.D.-A); by "la Caixa" Foundation (ID 100010434) via the Junior Leader Fellowship LCF/BQ/PR21/11840001 (X.D.-A); and by the European Commission's Horizon 2020 Research and Innovation Program via the GLOMICAVE project under grant agreement no. 952908 (X.D.-A) and via the Innovative Training Network Marie Skłodowska-Curie COL_RES project under the grant agreement no. 956279 (T.A., X.D.-A, and N.C). G.S. acknowledges support from the US National Institutes of Health grants R35 GM130385 and U01 CA235493. S.M.d.C. acknowledges the financial support of the Vicente Lopez Fellowship by Fundació Eurecat.

■ REFERENCES

(1) Pukala, T. *Rapid Commun. Mass Spectrom.* **2019**, *33* (S3), 72−82.
(2) Kartowikromo, K. Y.; Olajide, O. E.; Hamid, A. M. *J. Mass Spectrom.* **2023**, *58*, No. e4973.
(3) da Silva, K. M.; van de Lavoir, M.; Robeyns, R.; Iturrospe, E.; Verheggen, L.; Covaci, A.; van Nuijs, A. L. N. *Metabolomics* **2022**, *19*, 4.
(4) Broeckling, C. D.; Yao, L.; Isaac, G.; Gioioso, M.; Ianchis, V.; Vissers, J. P. C. *J. Am. Soc. Mass Spectrom.* **2021**, *32*, 661−669.
(5) Domingo-Almenara, X.; Guijas, C.; Billings, E.; Montenegro-Burke, J. R.; Uritboonthai, W.; Aisporna, A. E.; Chen, E.; Benton, H. P.; Siuzdak, G. *Nat. Commun.* **2019**, *10*, 5811.

(6) Das, S.; Tanemura, K. A.; Dinpazhoh, L.; Keng, M.; Schumm, C.; Leahy, L.; Asef, C. K.; Rainey, M.; Edison, A. S.; Fernández, F. M.; Merz, K. M. *J. Am. Soc. Mass Spectrom.* **2022**, *33*, 750−759.
(7) Li, X.; Wang, H.; Jiang, M.; Ding, M.; Xu, X.; Xu, B.; Zou, Y.; Yu, Y.; Yang, W. *Molecules* **2023**, *28*, 4050.
(8) Ross, D. H.; Seguin, R. P.; Krinsky, A. M.; Xu, L. *J. Am. Soc. Mass Spectrom.* **2022**, *33*, 1061−1072.
(9) Plante, P.-L.; Francovic-Fontaine, É.; May, J. C.; McLean, J. A.; Baker, E. S.; Laviolette, F.; Marchand, M.; Corbeil, J. *Anal. Chem.* **2019**, *91*, 5191−5199.
(10) Zhou, Z.; Xiong, X.; Zhu, Z.-J. *Bioinformatics* **2017**, *33*, 2235−2237.
(11) Colby, S. M.; Nuñez, J. R.; Hodas, N. O.; Corley, C. D.; Renslow, R. R. *Anal. Chem.* **2020**, *92*, 1720−1729.
(12) Zhang, H.; Luo, M.; Wang, H.; Ren, F.; Yin, Y.; Zhu, Z.-J. *Anal. Chem.* **2023**, *95*, 13913−13921.
(13) Zhou, Z.; Luo, M.; Chen, X.; Yin, Y.; Xiong, X.; Wang, R.; Zhu, Z.-J. *Nat. Commun.* **2020**, *11*, 4334.
(14) Baker, E. S.; Hoang, C.; Uritboonthai, W.; Heyman, H. M.; Pratt, B.; MacCoss, M.; MacLean, B.; Plumb, R.; Aisporna, A.; Siuzdak, G. *Nat. Methods* **2023**, *20*, 1836−1837.
(15) Rainey, M. A.; Watson, C. A.; Asef, C. K.; Foster, M. R.; Baker, E. S.; Fernández, F. M. *Anal. Chem.* **2022**, *94*, 17456−17466.
(16) Ross, D. H.; Cho, J. H.; Xu, L. *Anal. Chem.* **2020**, *92*, 4548−4557.
(17) Landrum, G. *RDKit (Open-Source Cheminformatics Software)*, 2021. https://www.rdkit.org/.
(18) Mauri, A. *alvaDesc: A Tool to Calculate and Analyze Molecular Descriptors and Fingerprints*; Roy, K., Ed.; Springer US, 2020; pp 801−820.
(19) Cereto-Massagué, A.; Ojeda, M. J.; Valls, C.; Mulero, M.; Garcia-Vallvé, S.; Pujadas, G. *Methods* **2015**, *71*, 58−63.
(20) Yang, F.; van Herwerden, D.; Preud'homme, H.; Samanipour, S. *Molecules* **2022**, *27*, 6424.
(21) Bajusz, D.; Rácz, A.; Héberger, K. *J. Cheminf.* **2015**, *7*, 20.
(22) de Cripan, S. M.; Cereto-Massagué, A.; Herrero, P.; Barcaru, A.; Canela, N.; Domingo-Almenara, X. *Biomedicines* **2022**, *10*, 879.
(23) Sheridan, R. P.; Feuston, B. P.; Maiorov, V. N.; Kearsley, S. K. *J. Chem. Inf. Model.* **2004**, *44*, 1912−1928.
(24) Asef, C. K.; Rainey, M. A.; Garcia, B. M.; Gouveia, G. J.; Shaver, A. O.; Leach, F. E.; Morse, A. M.; Edison, A. S.; McIntyre, L. M.; Fernández, F. M. *Anal. Chem.* **2023**, *95*, 1047−1056.

https://doi.org/10.1021/acs.analchem.4c00630
Anal. Chem. 2024, 96, 9088−9096