A MODULAR, MACHINE LEARNING-BASED, MULTI CHEMICAL CLASS CCS PREDICTION PIPELINE

Valentin Ianchis¹, Marisa Gioioso², Joanne Ballantyne³ and Johannes PC Vissers³
¹ Waters Corporation, Braşov, Romania, ² Waters Corporation Milford, MA, ³ Waters Corporation, Wilmslow, United Kingdom

OVERVIEW

- CCS prediction informatics pipeline for small molecule applications
- Out-of-sample prediction results illustrate good performance for a diverse set of chemical classes and cases (metabolomics, drug metabolism and food additives/natural product)
- Prediction results in agreement with molecular modelling and experimental CCS data from different IM-MS geometries
- Utility of relative and absolute predicted CCS values demonstrated in identification strategies

INTRODUCTION

The application of ion mobility – mass spectrometry (IM-MS) is gaining traction in numerous application areas, most noticeably in metabolomics, drug metabolism, protein conformation analysis, antibody drug conjugates and small molecule studies, as well as in environmental and food research. The gas-phase collision cross section (CCS) values obtained from IM-MS measurements are typically used to understand structure or to reduce the number of false positive identifications. Predictive CCS methods using machine learning and neural network approaches have been developed to complement empirical data. However, their implementation and use are not common, as expert informatics knowledge is typically required to create models and conduct predictions. A user-friendly modular tool named CCSondemand is presented that can be implemented in informatics workflows or run stand-alone.

Figure 1. Collisional cross section (CCS) data acquisition and processing workflow.
METHODS

IM-MS data acquisition and collection

$^{15}$C successively acquired on Vion IMS-Q-oaToF and SYNAAPT Q-IMS-oaToF geometries, as shown in Figure 1, with at least three technical replicates for 3,775 non-redundant compounds, representing 17 different chemical super classes in total as illustrated in Figure 2. Processing of the positive and negative ESI data, including the detection of common adducts, was conducted with UNIFI Scientific Information System or Progenesis QI.

Machine Learning

The Gradient Boosting (GB) algorithm was used to train a predictive model with relevant molecular descriptors. In the current application, this was achieved through the use of the XGBoost [1] and RDKit [2] python libraries, respectively.

A nested cross-validation (CV) strategy was used that included folds for hyperparameter optimization, using autoML, inside folds for training and testing the model, for which the scikit-learn library [3] was used, which performs Monte Carlo sampling over the hyperparameters. A description of the principles are provided in the results section.

Application pipeline

The core of the application pipeline uses node.js technology. Moreover, it includes both authentication/authorization, has a web-enabled front-end that accepts structures in various common formats, processes the incoming structures, and reports prediction results in a tabular format. The basic container based set-up of the pipeline is shown in Figure 3.

Figure 2: Chemical class molecular weight and adduct distributions training data set.
RESULTS

CCS prediction model

The GB algorithm is an ensemble of decision trees, each one developed as a successive approximation to training set outcomes, the predicted \( ^{15} \text{CCS}_{N_2} \) values. Ensemble methods are models that are made up of many smaller models, in this case decision trees, referred to as weak learners. Each tree will be small (and so highly regularized) and can be trained on only a randomly selected sample of points in the training set. These two features help the model not to overfit to the data.

The first tree’s objective is to predict the outcome, \( ^{15} \text{CCS}_{N_2} \) values, given molecular descriptors and by minimizing the square error of the prediction with the predicted \( ^{15} \text{CCS}_{N_2} \) value. Since predictions are not perfect, there will likely be a non-zero error:

\[
h_1(x) = y - f_1(x)
\]

The next tree will be formulated to predict the residual, \( h(x) \), from the previous, and added to the prediction from the first tree to suggest a better approximation to the outcome. This successive approximation repeats for every subsequent tree in the ensemble, such that the final model is an aggregate prediction of all individual trees:

\[
\tilde{y} = \sum f_m(x)
\]

where each weak learner, \( f_m(x) \), has been created to solve the objective, as a function of the previous set of weak learners:

\[
\text{argmin} \frac{1}{2} \left[ y(x) - (y_{m-1} + f_m(x)) \right]^2
\]

In addition to creating each tree in this manner, the trees are constructed to minimize the number of branches created, i.e. their complexity, through a regularization term added on to the objective function. A graphical overview of the training and prediction processes and a representation of the first decision tree are shown in Figure 4.

CCS prediction model performance

Shown in Figures 5 to 7 are some typical use cases demonstrating relative performance of the GB algorithm in comparison with other CCS prediction algorithms, and application examples using out-of-sample data sets. Note that the various CCS prediction models were trained on different data sets from different IM-MS geometries.

Figure 3. Basic container set-up for the \( ^{15} \text{CCS}_{N_2} \) prediction application pipeline.

Figure 4. (A) Workflow principle for training and evaluating the CCS prediction model and (B) a representation of the first decision tree, with predictors (e.g. \( f_43 \)) represented by their ordering in the prediction matrix (the final trained model contained 709 decision trees).

Figure 5. (A) MAE, RMSD and RMSE % prediction errors (predicted vs. experimental) for a SVR (light grey), neural network (dark grey), deep learning (orange) and CCSondemand (blue) for xenobiotic environmental contaminants, and (B) correlation coefficients for the same models (blue = [M-H]; orange = [M+H]+; grey = [M+Na]+).

Figure 6. Relative (%) \( ^{15} \text{CCS}_{N_2} \) prediction error (predicted vs. experimental) for a typical drug-metabolism application, illustrating an RMSD value of 1.2% for Tienilic acid and its main metabolites. \( X = \text{OH or glucuronide} \).
Application examples

In the absence of experimental or library data, CCS predictions can be used as part of an identification strategy to reduce analysis time. A metabolomics example from a controlled drug-dosed study/rat urine sample is shown in Figure 9, where TWCCS_{N2} prediction was used to reduce the number of possible identification candidates.

Solely based on accurate mass and fragmentation score, 5 out of 6 isomeric candidate structures were identified. Including predicted TWCCS_{N2} as a search parameter, 3 structures can be eliminated from consideration. Of the 2 remaining structures, trigonelline is more likely, since it represents a primary metabolite.

A second application example is shown in Figure 10, illustrating the relative and absolute CCS values of positional (regional) isomers using different in silico, molecular modeling and machine learning based, and analytical, drift tube and traveling wave ion guide (TWIG) IM-MS, techniques. A typical use case could be the structural identification of metabolites of small molecule drugs in cases where the exact position of a biotransformation could not be identified by conventional MS/MS.

The (A) panel shows the normalized, relative values of the in silico approaches, demonstrating good correlation between the two techniques, whereas the (B) panel illustrates the absolute CCS values for one of the isomers obtained with all methods. Compared to drift tube (DTCCS_{N2}) measurement [4], errors were observed of –1.7% (average molecular modeling [4]), -0.4% (TWCCS_{N2}), and 1.2% (GB predicted TWCCS_{N2}).

Figure 7. Relative (%){\textsuperscript{TW}}CCS_{N2} prediction error (predicted vs. experimental) for pyrrolizidine alkaloids, compounds of toxicological concern measured in food and present or isolated from plants, illustrating an RMSD value of 1.2%. Blue = common substructure.

Figure 8. Web-based interface example of the use of the API of the TWCCS_{N2} prediction pipeline.

Figure 9. Compound identification search result against HMDB using 10 ppm precursor, 20 ppm fragment ion and 5% TWCCS_{N2} search tolerances. Candidate identifications were rejected on fragmentation score (green), CCS accuracy (orange) or biological relevance (yellow).

Figure 10. In-silico and experimental CCS values methylenedianiline (MDA) regioisomers; 4,4’-MDA (1 and 4 = NH_2), 2,2’-MDA (2 and 3 = NH_2) and 2,4’-MDA (1 and 3 = NH_2). (A) Normalized molecular modeling CCS vs. normalized GB predicted TWCCS_{N2} and (B) Molecular modeling CCS (orange), DTCCS_{N2} (yellow), TWCCS_{N2} (grey) and GB predicted TWCCS_{N2} (blue) for 4,4’-MDA.
CONCLUSION

- A modular informatics prediction pipeline is described and its performance and application demonstrated.
- The prediction model showed good performance for a wide and diverse set of chemical classes and cases.
- Prediction errors are in most instances of the same order of magnitude as experimental intra-instrument CCS measurement variation.
- Improved metabolomics identification results were obtained since the addition of predicted CCS as a search parameter reduced the number of potential identification candidates and thus analysis time.
- Absolute and predicted CCS values were found to be in good agreement with drift tube and molecular modelling results.

References

Acknowledgement
Scientific Operations and Biomedical Research Marketing of Waters Corporation are kindly acknowledged for providing experimental and figure data.