Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Manuscript Category

Original Research

PeakBot: Machine learning based chromatographic peak picking

Christoph Bueschl[1,2,*], Maria Doppler[2,3], Elisabeth Varga[4], Bernhard Seidl[2], Mira Flasch[4], and Juergen Zanghellini[1,*]

[1]Department of Analytical Chemistry, University of Vienna, Waehringer Str. 40, A-1090 Vienna, [2]IFA-Tulln, University of Natural Resources and Life Sciences Vienna, Konrad-Lorenz-Str. 20, A-3430 Tulln, [3]University of Natural Resources and Life Sciences, Core Facility Bioactive Molecules: Screening and Analysis, Konrad-Lorenz-Str. 20, A-3430 Tulln, [4]Department of Food Chemistry and Toxicology, University of Vienna, Waehringer Str. 38-40, A-1090 Vienna

*To whom correspondence should be addressed.

Abstract

Motivation
Chromatographic peak picking is among the first steps in software pipelines for processing LC-HRMS datasets in untargeted metabolomics applications. Its performance is crucial for the holistic detection of all metabolic features as well as their relative quantification for statistical analysis and metabolite identification. Unfortunately, random noise, non-baseline separated compounds and unspecific background signals complicate this task.
Results
A machine-learning framework entitled PeakBot was developed for detecting chromatographic peaks in LC-HRMS profile-mode data. It first detects all local signal maxima in a chromatogram, which are then extracted as super-sampled standardized areas (retention time vs. *m/z*). These are subsequently inspected by a custom-trained convolutional neural network that forms the basis of PeakBot's architecture. The model reports if the respective local maximum is the apex of a chromatographic peak or not as well as its peak center and bounding box.
In independent training and validation datasets used for development, PeakBot achieved a high performance with respect to discriminating between chromatographic peaks and background signals (F1 score of 0.99). A comparison of different sets of reference features showed that at least 100 reference features (including isotopologs) should be provided to achieve high-quality results for detecting new chromatographic peaks.
PeakBot is implemented in Python (3.8) and uses the TensorFlow (2.4.1) package for machine-learning related tasks. It has been tested on Linux and Windows OSs.
Availability
The framework is available free of charge for non-commercial use (CC BY-NC-SA). It is available at https://github.com/christophuv/PeakBot.
Contact: christoph.bueschl@univie.ac.at
Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Untargeted metabolomics approaches have gained much popularity in recent years. They aim at holistically detecting all compounds present in samples regardless of their chemical identity. Subsequently, statistical analysis is carried out to select differently abundant compounds between treatment and control conditions for further biological interpretation (Fiehn 2002).

The most commonly used analytical technologies for detecting metabolites in aqueous samples are liquid or gas chromatography (LC or

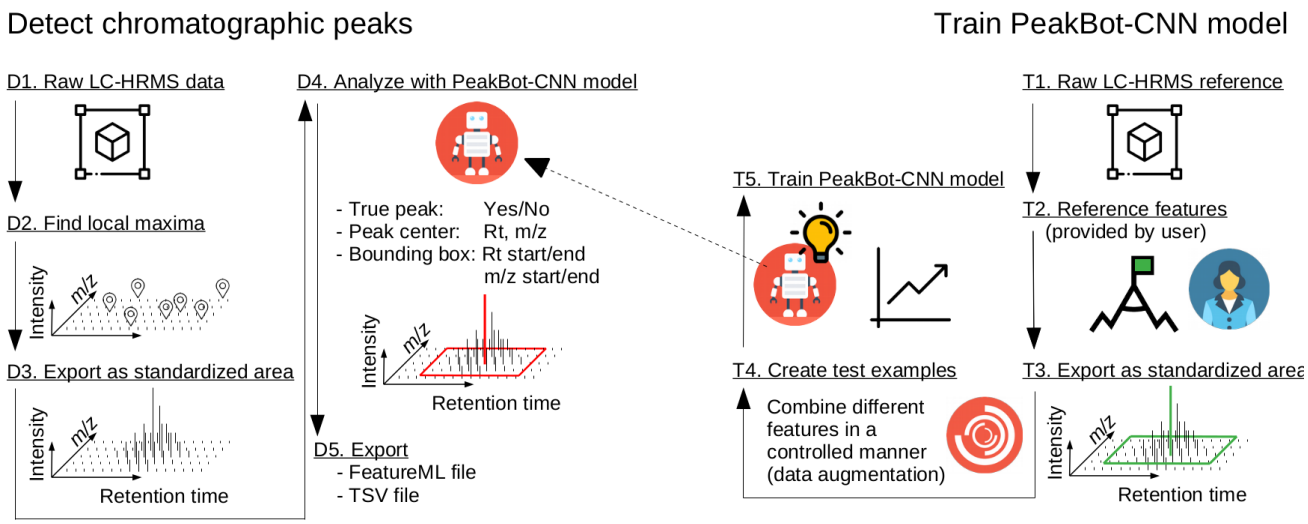


Figure 1 | Overview of the detection and training steps of PeakBot. The left panel shows the steps for detecting chromatographic peaks with an already trained PeakBot model, while the right panel shows the steps of training a new PeakBot model from reference LC-HRMS data.

GC) coupled with high resolution mass spectrometry (HRMS) and nuclear magnetic resonance (NMR) spectroscopy. In this respect, LC-HRMS offers the most versatility since it can be customized towards the samples themselves or certain constituents of interest (e.g., polar or apolar compounds, lipids, or secondary metabolites). GC-MS approaches are used to study volatile compounds and NMR approaches offer the possibility for absolute substance quantification as the underlying physical principle does not suffer from matrix effects (Segers et al. 2019).

While in targeted approaches the substances under investigation are carefully defined by the experimentalists and available as authentic reference standards, in untargeted approaches especially novel compounds are of interest. Depending on the metabolic capabilities of the organism under investigation, a high number of metabolites can be expected (Peisl, Schymanski, and Wilmes 2018; Alseekh and Fernie 2018). This high number of metabolic features together with multiple experimental conditions and replicates makes it difficult to manually investigate the raw-data and integrate each peak. Thus, automated software tools, which are capable of a) detecting chromatographic peaks in the dataset and b) integrate them reliably, are of utmost importance. These two steps are typically carried out simultaneously yielding an average mass-to-charge-ratio (*m/z*) and retention time as well as a relative abundance (peak area and/or intensity of most abundant signal contributing to the peak).

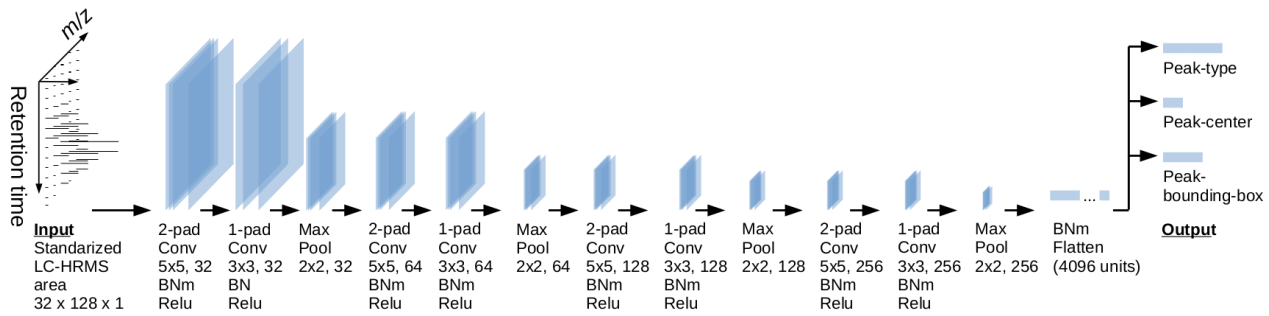
The metabolomics community has recognized the need for automatization early and started developing holistic and reliable (open source) software approaches. Arguably the best known open-source approaches free of charge are XCMS (Tautenhahn, Böttcher, and Neumann 2008) and XCMS-Online (Tautenhahn et al. 2012), MzMine2 (Pluskal et al. 2010), MSDial (Tsugawa et al. 2020), MetAlign (Lommen and Kools 2012), OpenMS (Röst et al. 2016), Lipid Data Analyzer (Hartler et al. 2017), among others (the interested reader is referred to (O'Shea and Misra 2020)). These tools scale well to different chromatographic conditions and MS instruments making them reliable and indispensable. However, as LC-HRMS methods can be customized (e.g., different chromatography times, mass ranges, resolution, separation power, etc.), one software or parameter set do not fit all experiments. Furthermore, optimizing a data processing software for a particular analytical technique can be a challenging process that requires both an in-depth knowledge of the analytics as well as the data processing algorithm in order to assess the performance of different parameter sets.

Machine-learning has recently gained attention for the task of peak-picking. For example, NeatMS (Gloaguen, Kirwan, and Beule 2020) is a machine-learning tool and a very interesting approach. It does not perform peak-picking itself, but classifies peaks detected with other tools and judges the quality of the detected peaks (high quality, acceptable, noise). Thus, it can easily be integrated into existing workflows. Another machine-learning approach is peakonly (Melnikov, Tsentalovich, and Yanshole 2020), which performs both classification and peak border estimation with the help of deep-learning. Both approaches operate in the centroid data mode.

Here we seek to explore the possibility of using machine-learning convolutional neural networks (CNN) models for the task of detecting chromatographic peaks in single profile-mode chromatograms. Other than most algorithmic approaches, the presented method uses a series of user-provided references (i.e., chromatographic peaks and different background types) to custom-train a machine-learning model that can recognize such chromatographic peaks and calculate their respective analytical properties directly in the raw profile-mode data.

2 Approach

The outcome of our research is a python-based framework entitled PeakBot that imports LC-HRMS datasets, pre-processes them and returns a list of detected chromatographic peaks. An overview of the processing pipeline of PeakBot is illustrated in Figure 1. Other than most other available tools for chromatographic peak picking, the PeakBot framework uses profile mode data instead of centroided data. Moreover, the presented approach also does not have a region-of-interest (ROI) generation step as for example XCMS, but rather enumerates all local maxima (or a reduced subset), exports areas around this local maxima as standardized, super-sampled areas of the chromatogram and tests these areas for the presence of a chromatographic peak in its center. If the CNN model (illustrated in Figure 2) detects a chromatographic peak, PeakBot also calculates the peak's center and bounding box (retention-time and *m/z* dimension). Moreover, PeakBot's CNN model can also be trained for other categories of local maxima such as various background types. For instance, a typically observed background type are "walls" that span rather large parts of the chromatogram with only minimal random changes in intensity. An example of such a wall is shown in Supplementary Figure 3.



y-pad: y-element padding with zero on each of the four matrix sides
Conv: Convolution layer; BNm: Batch-normalization: Relu: Rectified linear activation

Figure 2 | Overview of PeakBot's CNN model. In its standard configuration PeakBot starts from the standardized LC-HRMS area and applies a series of convolution and max-pooling steps to the previous layer. Finally, the output of these convolutions is flattened and the type of the local maxima as well as the peak's center and bounding box are determined.

2.1 Data processing

The main steps of detecting chromatographic peaks with PeakBot are illustrated in Figure 1 in the left panel. First, the profile-mode LC-HRMS data is loaded from the mzXML/mzML format. Then, all local maxima (i.e., individual *m/z* signals that are more abundant than their 8-neighboring *m/z* signals, Moore neighborhood) are detected. This detection is carried out regardless of whether the apex signals represent true chromatographic peaks or not.

All such detected local maxima are subsequently exported as standardized LC-HRMS areas consisting of a pre-set number of HRMS scans (default 32) and a pre-set number of *m/z* signals (default 128). As there are slight deviations in the *m/z* signals within even neighboring MS-scans, these standardized areas are super-sampled, meaning that an exact, equally spaced number of reference *m/z* values relative to the respective local-maximum is generated with the local maxima being in the center of this area. Finally, the standardized area is scaled to a maximum abundance of 1.

However, as the number of local maxima in an LC-HRMS sample can be quite high, an optional preprocessing step of PeakBot is a basic gradient-descend based peak detection approach. This preprocessing starts from each detected local maximum. Next, adaptive EICs are calculated for these local maximum (further information about this step is presented in Supplementary Figure 1). A fixed *m/z* value is not used but rather the EIC adapts from the previous *m/z* value of the previous scan thereby accounting for small shifts between the profile-mode signals. This EIC is then smoothed with a Savitzky-Golay filter. Starting from the local maxima the pre-processing algorithm moves left and right on the retention time axis until the infliction points are reached (strictly monotony decreasing EIC). Then the EIC is further traveled until either a scan has an increase in the intensity relative to its predecessor or until the intensity relative to the peaks' apex (the local maximum) undercuts 1%. If this preprocessing step is used, then only such local maxima with a minimum peak width are further used and exported, while the other local maxima are prematurely discarded. However, this pre-detection step can be omitted.

Each standardized area is then used as the input for the CNN model, which tests it for the presence of a chromatographic peak in its center. PeakBot reports detected chromatographic peaks either as single peaks or such having left and/or right isomeric neighboring peaks (partly overlapping chromatographic peaks). Moreover, if the local maximum and thus the standardized area do not contain a chromatographic peak, but for example background walls, then no peak will be reported but rather the category wall or background. For local maxima designated to contain chromatographic peaks, PeakBot will also calculate the peaks' center (which will mostly be the center of the standardized area) as well as a bounding-box in both the retention-time and *m/z* dimension.

Finally, all such local maxima that represent a chromatographic peaks according to PeakBot are exported to a tsv-file and as a featureML file that can be used to visualize the results with e.g. TOPPView (Sturm and Kohlbacher 2009).

2.2 Training of PeakBot

The main steps of training a new PeakBot-CNN model are illustrated in Figure 1 in the right panel. To train a new PeakBot model from scratch, reference LC-HRMS chromatograms and a list of reference features have to be provided by the user. The reference chromatograms do not need to be special in any case, however the list of reference features provided by the user must only consist of features that represent single, isolated ions with as little background as possible. Partly co-eluting isobaric or isomeric compounds are not supported yet and thus cannot be used as reference features. In a typical untargeted metabolomics experiments compiling such a list should be easily possible as either several compounds are measured as references, QC samples or several unknown metabolites can be recognized by trained analytical chemists. Moreover, this list of reference features can also be extended easily by using different isotopologs of the already available compounds determined by the user. For each reference feature the user must specify the peak center (retention-time and *m/z* value) as well as the bounding box of that feature in the retention-time and *m/z* dimension. Furthermore, the user must specify a couple of background-regions that contain backgrounds.

Once specified, PeakBot will first recognize the features in the different samples and calculate their properties directly from the data (weighted-average *m/z* and apex retention-time) thereby accounting for small shifts and drifts. For this refinement step the basic gradient-descend preprocessing algorithm is utilized.

From this user-curated input, many training instances that are used as the input for training a new PeakBot-CNN model are automatically generated, thereby supporting the specific characteristics of the LC-HRMS dataset and especially those of the references provided by the user (narrower/broader chromatographic peaks, deviation of related *m/z* signals, saturation effects, etc.). Each such training instance has a ground-truth center representing the local-maximum. This ground-truth can be either a feature or a background and the respective area in the

chromatogram is super-sampled identical to the super-sampling carried out during the detection phase of PeakBot. Then, in order to avoid over-fitting and to automatically generate many training instances (data augmentation), this ground-truth and super-sampled area is iteratively combined with randomly selected other features/background signals (decoys) from the user's input thereby simulating a diverse neighborhood of the local maximum. These other features/background signals are randomly placed into the standardized area of the ground-truth. Some degree of overlap is allowed. Additionally, all signals are randomly varied to a certain degree (multiplied with a random value, default within the range of 0.9 to 1.1) to simulate intensity-deviations caused by the instrument.

Each such generated and augmented training instance is then used by PeakBot to optimize the CNN model with the aim of correctly detecting the training instance, classify it and calculate its properties.

2.3 PeakBot-CNN model

PeakBot is a convolutional neural network (CNN) that has a standardized LC-HRMS area as input. The architecture is illustrated in Figure 2. Then, several convolutional steps and pooling layers follow. Two convolutional layers are followed by a max-pooling layer and a ReLu activation. This cascade is repeated four times each reducing the intermediate size but increasing the number of computed information gained from the layers. The number and computed information can easily be extended if required by the user. Finally, after the last convolutional layer the convoluted area is flattened, from which the outputs (feature-type, boundaries and center) are calculated. These outputs are A) peak type (one-hot-encoded peak category consisting of 6 types in its standard configuration), B) peak-center (indices in the retention-time and *m/z* dimension in the standardized area), and C) peak-bounding-box (start/stop indices in the retention-time and *m/z* dimension in the standardized area).

This training is performed with the TensorFlow framework. For optimizing the output of the model, different loss functions are applied for the local-maximum types (loss: categorical crossentropy), peak-center (loss: Huber), peak-bounding-box (loss: Huber), peak mask (loss: binary crossentropy). Moreover, also the peak accuracy (loss: categorical accuracy) and the peak-bounding-box intersect over union (loss: custom) are reported to the user but these two are not used to optimize the CNN model.

2.4 Grouping of results from different chromatograms

To group the detected chromatographic peaks from different LC-HRMS chromatograms, PeakBot implements a k-nearest-neighbor approach. Here all detected chromatographic peaks from all samples of the experiment are first loaded. Then for each feature the nearest features in both the retention-time and *m/z* dimension are searched for in all other chromatograms (within a user-defined window). Then, all such features are updated with the mean retention-time and *m/z* values of all its neighbors thereby accounting for small retention-time and *m/z* variations across the chromatograms. Furthermore, this step can be repeated multiple times with wider and narrower parameters thereby iteratively aligning the chromatograms to each other. If experimental conditions with vastly different metabolic constituents are analyzed, the different experimental groups can be aligned independently first to create a virtual consensus sample and then the results of different experimental conditions can be merged.

2.5 Implementation details

PeakBot is implemented in the python programming language (at least 3.8.1). The machine-learning CNN model is implemented with the tensorflow package (at least 2.4.1, https://www.tensorflow.org/), which allows the user to execute the functionality easily on CUDA-enabled graphics cards.

Furthermore, PeakBot utilizes the numba package (http://numba.pydata.org/) for just-in-time compilation to detect the local maxima in the LC-HRMS data and export them to the standardized areas. The generation of the augmented training instances also benefits from the numba package as this functionality can also be executed con CUDA-enabled graphics cards.

PeakBot supports the mzXML/mzML file format for reading raw LC-HRMS data and is available at https://github.com/christophuv/PeakBot.

3 Methods

3.1 Training and validation datasets

To train a new CNN model and monitor PeakBot's performance during the training step, different datasets were generated from the user-provided reference feature list. Briefly, these were derived from untargeted metabolomics experiments of wheat ears (named "WheatEar", further information is provided in Supplementary Information 1), porcine hepatic microsome incubation samples (named "PHM", further information is provided in Supplementary Information 2) and 3 other datasets obtained from the MetaboLights repository (https://www.ebi.ac.uk/metabolights; MTBLS1358, MTBLS868, MTBLS797, further information about these datasets is provided in Supplementary Information 3). Separate PeakBot CNN models were trained for each dataset to evaluate the performance as presented in section 2.2. 5-fold cross validation with randomly selected peaks for training and validation was also carried out.

For each such LC-HRMS dataset, 4 reference-set were generated for training and verifying the CNN models' performance. These sets are:

- T: The training set obtained from the training chromatograms with a list of at least 100 reference features.
- V: A validation set obtained from the training chromatograms (same as for T) but different reference features (no overlap with T).
- iT: A validation set obtained from different validation chromatograms (no overlap with T), but the reference features used for training the model (same as in T).
- iV: A validation set obtained from different validation chromatograms (no overlap with T) and also different reference features (no overlap with T). As this set neither shares the chromatograms nor the reference features with the training set T it can be considered an independent validation set of the model.

During training, the agreement of the ground truth (i.e., the generated training instances) and the predictions of the model are summarized in several loss/metric values detailed in Table 1. All validation results were critically evaluated with respect to the metrics reported by the training process. Moreover, a subset of all results was also manually verified by the person training the CNN model.

4 Results and Discussion

Humans are extraordinarily good at recognizing structure in unstructured data (e.g., object detection and separation). Moreover, they

Table 1 | Overview of loss/metric values.

Loss/metric	Description
Box_iou	Intersection of union (IOU) metric of the predicted and ground-truth bounding box of a true chromatographic peak. Higher values indicate a better overlap between the ground-truth and the prediction (Rahman and Wang 2016).
Box_loss	Mean squared error (MSE) of the predicted and ground-truth bounding box (retention time start and end, *m/z* start and end) of a true chromatographic peak. Lower values indicate a better overlap.
Center_loss	MSE of the predicted and ground-truth peak center (retention-time and *m/z*) of a true chromatographic peak. Lower values indicate a better overlap.
PeakType_ categorical_ accuracy	Accuracy of the 6 peak-types (one-hot encoded) reported by PeakBot (single peak, peak with left/right/both-sided isomers, wall, background). Higher values indicate a better overlap.
peakType_loss	Categorical crossentropy of the 6 peak-types reported by PeakBot for a putative chromatographic peak. Lower values indicate a better overlap.
peakType_pF1	F1-score for the peakType reduced to either a chromatographic peak regardless of isomeric forms or not. Higher values indicate a better overlap
peakType_pFPR	False-positive rate of reporting a true chromatographic peak. Lower values indicate less incorrectly detected chromatographic peaks.
peakType_pTRP	True-positive rate of reporting a true chromatographic peak. Higher values indicate more correctly detected chromatographic peaks.

also are also capable of recognizing chromatographic peaks with a high accuracy and are able to separate overlapping peaks from each other. However, it is not possible for them to pick many peaks manually as unfortunately humans are also terribly slow in that respect. Thus, reliable, automated peak-picking is important.

Here PeakBot is presented. It is a CNN-machine learning model for the detection of chromatographic peaks in LC-HRMS datasets. As the analytical parameters of different datasets (e.g., chromatographic peak width and/or *m/z* deviation) can be quite different, the PeakBot CNN models can be custom trained for each dataset or analytical method.

PeakBot exports local maxima as standardized areas, which can also be tailored to the LC-HRMS data (e.g. scan rate). Thus, PeakBot models can be adjusted to extract a wider area around each detected local maximum, while for UHPLC approaches with lower scan-rates the standardized area can be relatively narrow. Subsequently, a custom trained CNN model inspects these standardized areas for the presence of chromatographic peaks (like the provided reference features) or to different types of backgrounds observed in the dataset.

PeakBot has different outputs namely peak-type, -center and -bounding-box. The peak-type is a one-hot-encoded class identifier and consists of four types if a true chromatographic peak is believed to be in the center of the standardized area. These four peak types are whether the peak is isolated (no left and right isomeric neighbors) or not. Additionally, two peak categories also indicate if the local maximum of the standardized area is a background signal (either a wall or a random

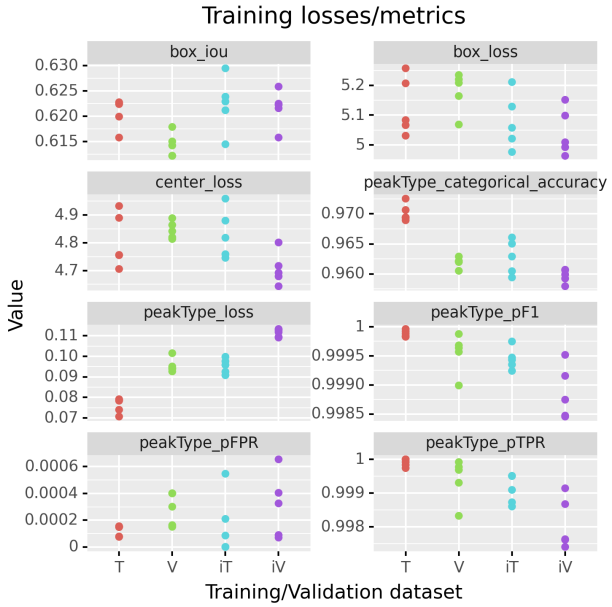


Figure 3 | Overview of training losses and metrices on the training dataset T and the additional validation datasets V, iV, iT as well as the external dataset eV obtained from a different LC-HRMS method.

signal). If PeakBot considers the local maximum to be a true chromatographic peak, it will also calculate the peak's center and bounding box, which describe the location of the peak in the two-dimensional chromatogram.

4.1 Training and validation of PeakBot

Training of machine-learning models is a difficult process with different aspects to be considered as otherwise the model might overfit. In this respect overfitting refers to a model that has a minimal error on the training dataset but does not generalize well with new data not used for the training process and thus has a poor prediction quality. It is common practice to test the performance of machine-learning models during training also with independent validation instances to check for such effects. If the model generalizes well, the validation- and training-datasets should have similar loss and metric values.

Three validation data sets are used to verify and judge the trained model's performance. For this, at least one independent dataset (i.e., same analytical method, but different features than used for training, different chromatograms than used for training) are used. The losses and metrices for the training and validation for the 5 demonstration datasets are illustrated in Figure 3 and Supplementary Figure 6. The loss and metric values were similar (less than 5% difference in the metrices) between the training set (T) and the 3 validation sets (V, iT, iV) indicating that the model did not overfit to the training data. The model was able to accurately estimate if a local maximum was indeed a chromatographic peak or not (F1 score, true-positive-rate and false-positive-rate for the peakType). In this respect, the independent set iV was most interesting as it was derived from chromatograms and reference features completely independent of the training data. The trained PeakBot models were able to recognize chromatographic peaks there with a true-positive-rate of more than 99% (peakType_pTPR metric) on average, while only few (less than 1% on average; peakType_pFPR metric) of the true chromatographic peaks were not detected as such. The peaks' centers were similar to the training and

other validation dataset, while the bounding-box agreed less (less IOU overlap).

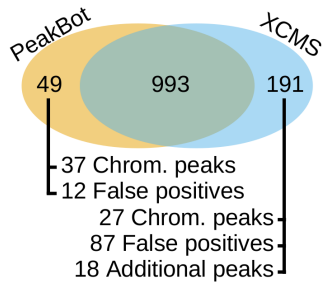


Figure 4 | Comparison of results from PeakBot and XCMS.

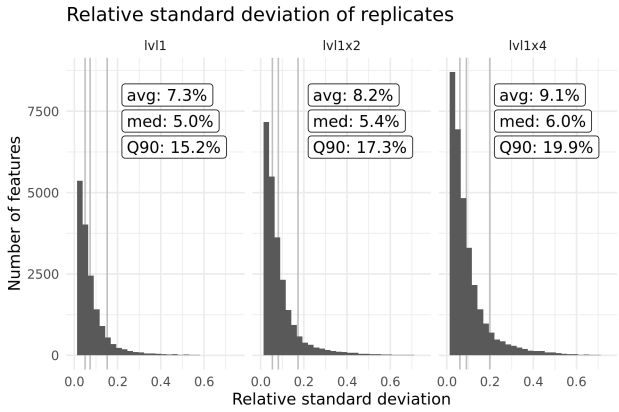


Figure 5 | RSD values of the peaks' area of the WheatEar dataset.

4.2 Size of reference feature set

PeakBot requires reference features for training of a CNN model. These reference features must be isolated chromatographic peaks curated by the user. Moreover, the different reference features should cover all expected, typical peak shapes of the particular analytical method.

To investigate the effect of differently sized reference lists upon the CNN-models performance, subsets of different size were generated from the initial reference feature list consisting of 2731 reference features, which were combined to 1,048,576 training instances. Different sizes (10 to 2400) of the reference list were used to train separate PeakBot CNN models and to obtain their metric values. In summary, a slight reduction in performance could be observed when fewer reference features were used during the training phase (Supplementary Figures 4 and 5). At around 100 features on the reference list a change in the metric values could be observed. Using a lower number of features greatly reduced the performance of the model, while the model did not noticeably benefit from a higher number of reference features. Nevertheless, the different models were still able to differentiate between true chromatographic peaks or backgrounds with a high certainty. Thus, this investigation showed that at least 100 reference features (including isotopolog features) should be provided during training by the user for good performance. These reference features were then automatically combined with other reference features and background into many different training instances, thereby removing this laborious task from the users. Moreover, only the approximate retention-time and *m/z* values of the reference features need to be provided by the user as this generation will automatically adapt to the respective properties in the reference chromatograms.

4.3 Comparison with XCMS based peak-picking

XCMS and especially the centWave algorithm is a popular software package for processing LC-HRMS data in untargeted metabolomics approaches (Tautenhahn, Böttcher, and Neumann 2008). It requires the data in the centroid-mode and then first generates a list of regions of interest (ROIs). Each such ROIs' corresponding EIC is then calculated and a wavelet-based algorithm detects chromatographic peaks of different widths on it. Optionally, a Gaussian peak shape can also be fitted to the detected chromatographic peaks.

To compare PeakBot with XCMS with respect to its ability to detect chromatographic peaks in LC-HRMS datasets, a comparison of the two approaches on the PHM dataset was carried out. Both algorithms were separately optimized to the PHM dataset (i.e., parameters in XCMS, training a new CNN-model in PeakBot). Then, the detected peaks of XCMS and PeakBot were compared automatically to see how well these two methods agree. Supplementary Information 4 provides details of this automated comparison (parameters of both approaches and comparison).

Between 80% and 84% of all features were detected by both approaches, 4% - 6% of the chromatographic peaks were solely detected by PeakBot and another 11% to 15% of the features were only found with XCMS, demonstrating that both approaches detect the majority of the chromatographic peaks (Supplementary Table 1). This was of no surprise as the majority of the chromatographic peaks were baseline-separated or had only partial overlaps in the peaks' flanks.

The results obtained for one chromatogram by XCMS (1,124 features) and PeakBot (1,042 features) were then also compared manually using TOPPView (Sturm and Kohlbacher 2009). Some examples of the detected and classified features are illustrated in Supplementary Figure 3. Of the 49 features only detected by PeakBot only 12 features (24%) were not chromatographic peaks as manually categorized by us, while the remaining 37 features, which were 76% of the additional features only detected with PeakBot, were true chromatographic peaks. Interestingly, 24 of these additional peaks were of high quality and abundance but with almost baseline separated isomeric other compounds. The features only detected by XCMS were also manually inspected. Of these 131 additional features, 87 features (66%) were categorized as incorrectly detected wall-backgrounds. The remaining 44 features (33%) were true chromatographic peaks correctly detected by XCMS. It should be noted that XCMS only uses the set intensity threshold during its region-of-interest generation step but not later. Thus, it is possible that some low abundant chromatographic peaks that fell below this threshold were still detected by XCMS. Moreover, the centroiding step also slightly increased the signals' abundances and thus few chromatographic peaks detected with XCMS exceeded this threshold, while in the profile mode data used by PeakBot the most abundant profile peak signal were below the set threshold and were therefore not detected. 18 of the additional 44 features were detected by XCMS because of these two facts.

Interestingly, a difference of either +2 or -2 ppm was observed on average for those features detected with both approaches. This difference was an artefact and was based on the super-sampling for the standardized area. The local-maximum weighted-average *m/z* value agreed much better (+1 to -1 ppm deviation) with the *m/z* value reported by XCMS after the centroiding process. Consequently, PeakBot was adapted to not reply on the super-sampled area but instead calculate the mean *m/z* value for detected chromatographic peaks from the actual data. This observation is illustrated in Supplementary Figure 2.

Table 2 | Run times of PeakBot for the WheatEar dataset.

	Workstation PC	HPC server
CPU	Intel i7-4790K	AMD Epyc 7542
Main memory	16 GB	2,096 GB
GPU	Nvidia GTX 970	Nvidia Tesla v100S
GPU memory	4 GB	32 GB
OS	Windows 10	Debian 10.9
Generating training and validation sets	48 min	8.9 min
Training CNN model	33 min	13.4 min
Processing single chromatogram (31,259 local maxima)	45 sec	17 sec

4.4 Comparison with MS-Dial based peak picking

MS-Dial is another popular peak-picking method for LC-HRMS datasets (Tsugawa et al. 2020). It supports both the profile and centroid data mode as well as liquid and gas chromatography systems.

PeakBot and MS-Dial were compared on the MTBLS1358 dataset. Supplementary Information 5 provides details of this automated comparison (parameters of both approaches and comparison). The majority of chromatographic peaks in this dataset were quite narrow (some low-abundant peaks are only 3-5 scans wide) and many had rather abrupt beginnings and long tailings. Consequently, peak-picking was complicated. Both datasets were independently processed, and the results were compared. The majority of the chromatographic peaks were detected with both software tools (between 63% and 71%). A single file was then chosen, and 100 peaks only detected with PeakBot and 100 peaks only detected with MS-Dial were randomly chosen and inspected manually. From the 100 features only detected with PeakBot, 75 were considered true chromatographic peaks, another 11 were complicated (i.e., manual annotation was unsuccessful) and 14 features were false positives. On the other hand, with MS-Dial 25 features were characterized as true chromatographic peaks and 20 as broad chromatographic peaks, while another 12 were complicated (i.e., manual annotation was unsuccessful) and 43 were false-positives. More information is available in Supplementary Table 3.

In summary, with PeakBot more features were detected than with MS-Dial on the same dataset. The manual investigation also showed that a higher number of with PeakBot exclusively detected features (75%) were true chromatographic peaks, while with MS-Dial only 45% were true chromatographic peaks. However, PeakBot and MS-Dial both had 11% respectively 12% features, which were complicated for manual verification.

4.5 Relative quantification

Reliable peak detection and peak area integration for subsequent statistical comparison of different experimental conditions are the main tasks of peak picking. PeakBot can automatically derive peak borders for detected chromatographic peaks. These borders are used for peak area integration. To test how well and repeatable the trained CNN model integrated the peaks' areas, the data was a) manually sighted to confirm correctly detected peak areas and b) a comparison of the replicates from the WheatEar study using relative standard deviation (RSD) was carried out. The manual, human verification confirmed that PeakBot reliably detected the peaks' borders in the profile mode data. Moreover, the RSD values of the replicates from the WheatEar study, which were technical replicates of repeated injections from the same sample, also showed small sample-to-sample differences, including both instrumental as well as the data processing variations. On average less than 10% RSD was observed for the different features and 90% of them had RSD values lower than 20%, which was well within the typical ranges of untargeted metabolomics experiments (Bueschl et al. 2014). No systematic error was observed in that respect. The distribution of the RSD values is shown in Figure 5.

4.6 Runtime

CNN models are computationally more complex than currently often utilized algorithmic approaches that work with one-dimensional EIC data. It is thus of no surprise that the evaluation with PeakBot takes longer in comparison. However, PeakBot uses the TensorFlow framework for its machine-learning related tasks and thus it has the possibility to harness the computational power of graphical processing units (GPUs) of CUDA-enabled graphics cards. While such a setup does not allow processing several chromatograms in parallel, it allows processing several local maxima of a single chromatogram in parallel thereby reducing the overall computation time.

All run-time tests were carried out using either an Nvidia Tesla V100S or an Nvidia GTX 970 GPU. The first one is at the time one of the best performing CUDA cards available, however, it is typically installed only in super-computing centers, while the latter one is some 6-year-old mainstream graphics card. On both systems the CNN-model and the necessary pre-processing routines that are also implemented with GPU support executed within reasonable time demonstrating that PeakBot's pre-processing and CNN model execute quickly even on older, desktop-PC hardware. The systems and the runtimes of generating the training dataset, training a new PeakBot model, as well as detecting chromatographic peaks in LC-HRMS samples are summarized in Table 2.

4.7 Generation of MSMS exclusion lists

Data dependent acquisition of MSMS spectra is typically carried out in untargeted metabolomics experiments. However, large backgrounds (i.e., walls) especially in samples with low metabolite ion abundances can interfere with this acquisition strategy and thus the coverage of the sample's metabolites is reduced. The degree of the reduction depends on the background and its intensity. An example of such a chromatogram where large parts of the background have been used for data dependent acquisition is shown in Supplementary Figure 7.

In such experiments PeakBot's background annotation feature can be useful for the automated generation of MS/MS exclusion lists for subsequent measurements. Such exclusion lists can be generated e.g., from the detected walls in background samples. In the shown example the use of such an exclusion list increased the MS/MS coverage of true chromatographic peaks from 1,356 respectively 855 to 1,605 respectively 1,810 for the negative respectively positive ionization mode (Supplementary Table 4), which is an increase of 18% respectively 112%. Nevertheless, it should be noted that peaks with the same *m/z* ratio as such walls could potentially be missed with this strategy.

Summary

PeakBot is a high-performance machine-learning based approach for detecting chromatographic peaks in LC-HRMS profile-mode datasets.

The use of reference features and chromatograms makes it independent of different parameter settings and thus it can adapt to the particular LC-HRMS dataset characteristics. Furthermore, the advantage here is that PeakBot reports metrices about its training-progress thereby allowing the user to monitor its performance on an independent dataset not used for training but for validation. To train a new model, PeakBot offers functionality for the automated generation of diverse training instances thereby reducing the number of training features the user has to provide.

For prediction PeakBot requires only the raw chromatograms in profile mode data and an intensity threshold. All local maxima (or alternatively these with a gradient-descend peak shape) are used as starting points for the detection of chromatographic peaks. Each local maximum is then inspected by the reference-feature trained model for chromatographic peaks or backgrounds.

In our tests, PeakBot showed to be a reliable peak-picking method and can competed with the established XCMS software, with itself is a high-performance and reliable algorithm if tuned properly. PeakBot's performance has also been compared with MS-Dial and in the tests with a dataset consisting of many chromatographic peaks with a complicated shape it performed better than the MS-Dial software in recognizing true chromatographic peaks. Moreover, PeakBot also reliably integrates the peaks' boundaries for relative quantification thereby not introducing a too high data processing variation. Additionally, as PeakBot also classifies background signals it can be used for the efficient and reliable generation of exclusion lists for broad walls, which are often unnecessarily used for MS/MS scans thereby saving experimentalists time, resources and sample.

We are convinced that PeakBot is a viable and good alternative for peak picking. Specifying a list of reference features and letting the model learn their characteristics will allow for easier and improved peak-picking over algorithmic approaches with parameters. As a python package PeakBot also allows programmatic access to the raw LC-HRMS data and will be of interest for prototyping ideas. Furthermore, it can be integrated in other software tools and the models can easily be reused or shared. In future releases of the framework, we want to add machine-learning capabilities also to the tasks of grouping/aligning detected features from different results, re-integration, convolution and annotation of features and provide an easy-to-use graphical user interface for users without coding experience.

PeakBot and related examples are available free of charge for academic use at https://github.com/christophuv/PeakBot and https://github.com/christophuv/PeakBot_example.

Acknowledgements

Figures have been designed using resources from https://flaticon.com. We acknowledge Eszter Borsos for performing the porcine hepatic microsome assays. Furthermore, the authors thank the members of the Mass Spectrometry Center at the University of Vienna, where LC-HRMS measurements (LTQ Orbitrap Velos) were performed, for skillful technical support. This project was supported by the BOKU Core Facility Bioactive Molecules: Screening and Analysis.

Funding

This work has been supported by the University of Vienna, Faculty of Chemistry specifically the Department of Analytical Chemistry and the Department of Food Chemistry and Toxicology and included the access to the Mass Spectrometry Center. Furthermore, the Austrian Science Fund is acknowledged for financial support (project SFB-Fusarium-37) as well as the Provincial Government of Lower Austria (project MetExtend).

Conflict of Interest: none declared.

References

Alseekh, Saleh, and Alisdair R. Fernie. 2018. "Metabolomics 20 Years on: What Have We Learned and What Hurdles Remain?" *The Plant Journal* 94 (6): 933–42. https://doi.org/10.1111/tpj.13950.

Bueschl, Christoph, Bernhard Kluger, Marc Lemmens, Gerhard Adam, Gerlinde Wiesenberger, Valentina Maschietto, Adriano Marocco, et al. 2014. "A Novel Stable Isotope Labelling Assisted Workflow for Improved Untargeted LC–HRMS Based Metabolomics Research." *Metabolomics* 10 (4): 754–69. https://doi.org/10.1007/s11306-013-0611-0.

Fiehn, Oliver. 2002. "Metabolomics — the Link between Genotypes and Phenotypes." In *Functional Genomics*, edited by Chris Town, 155–71. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-010-0448-0_11.

Gloaguen, Yoann, Jennifer Kirwan, and Dieter Beule. 2020. "Deep Learning Assisted Peak Curation for Large Scale LC-MS Metabolomics." Preprint. Bioinformatics. https://doi.org/10.1101/2020.08.09.242727.

Hartler, Jürgen, Alexander Triebl, Andreas Ziegl, Martin Trötzmüller, Gerald N. Rechberger, Oana A. Zeleznik, Kathrin A. Zierler, et al. 2017. "Deciphering Lipid Structures Based on Platform-Independent Decision Rules." *Nature Methods* 14 (12): 1171–74. https://doi.org/10.1038/nmeth.4470.

Lommen, Arjen, and Harrie J. Kools. 2012. "MetAlign 3.0: Performance Enhancement by Efficient Use of Advances in Computer Hardware." *Metabolomics* 8 (4): 719–26. https://doi.org/10.1007/s11306-011-0369-1.

Melnikov, Arsenty D., Yuri P. Tsentalovich, and Vadim V. Yanshole. 2020. "Deep Learning for the Precise Peak Detection in High-Resolution LC–MS Data." *Analytical Chemistry* 92 (1): 588–92. https://doi.org/10.1021/acs.analchem.9b04811.

O'Shea, Keiron, and Biswapriya B. Misra. 2020. "Software Tools, Databases and Resources in Metabolomics: Updates from 2018 to 2019." *Metabolomics* 16 (3): 36. https://doi.org/10.1007/s11306-020-01657-3.

Peisl, B.Y. Loulou, Emma L. Schymanski, and Paul Wilmes. 2018. "Dark Matter in Host-Microbiome Metabolomics: Tackling the Unknowns–A Review." *Analytica Chimica Acta* 1037 (December): 13–27. https://doi.org/10.1016/j.aca.2017.12.034.

Pluskal, Tomáš, Sandra Castillo, Alejandro Villar-Briones, and Matej Orešič. 2010. "MZmine 2: Modular Framework for Processing, Visualizing, and Analyzing Mass Spectrometry-Based Molecular Profile Data." *BMC Bioinformatics* 11 (1): 395. https://doi.org/10.1186/1471-2105-11-395.

Röst, Hannes L, Timo Sachsenberg, Stephan Aiche, Chris Bielow, Hendrik Weisser, Fabian Aicheler, Sandro Andreotti, et al. 2016. "OpenMS: A Flexible Open-Source Software Platform for Mass Spectrometry Data Analysis." *Nature Methods* 13 (9): 741–48. https://doi.org/10.1038/nmeth.3959.

Segers, Karen, Sven Declerck, Debby Mangelings, Yvan Vander Heyden, and Ann Van Eeckhaut. 2019. "Analytical Techniques for Metabolomic Studies: A Review." *Bioanalysis* 11 (24): 2297–2318. https://doi.org/10.4155/bio-2019-0014.

Sturm, Marc, and Oliver Kohlbacher. 2009. "TOPPView: An Open-Source Viewer for Mass Spectrometry Data." *Journal of Proteome Research* 8 (7): 3760–63. https://doi.org/10.1021/pr900171m.

Tautenhahn, Ralf, Christoph Böttcher, and Steffen Neumann. 2008. "Highly Sensitive Feature Detection for High Resolution LC/MS." *BMC Bioinformatics* 9 (1): 504. https://doi.org/10.1186/1471-2105-9-504.

Tautenhahn, Ralf, Gary J. Patti, Duane Rinehart, and Gary Siuzdak. 2012. "XCMS Online: A Web-Based Platform to Process Untargeted Metabolomic Data." *Analytical Chemistry* 84 (11): 5035–39. https://doi.org/10.1021/ac300698c.

Tsugawa, Hiroshi, Kazutaka Ikeda, Mikiko Takahashi, Aya Satoh, Yoshifumi Mori, Haruki Uchino, Nobuyuki Okahashi, et al. 2020. "A Lipidome Atlas in MS-DIAL 4." *Nature Biotechnology* 38 (10): 1159–63. https://doi.org/10.1038/s41587-020-0531-2.