From raw ion mobility measurements to disease classification: a comparison of analysis processes

Salome Horsch[1], Dominik Kopczynski[2], Jörg Ingo Baumbach[3], Jörg Rahnenführer[1,*], and Sven Rahmann[2,4,*]

[1]**Department of Statistics, TU Dortmund, Dortmund, Germany**
[2]**Computer Science XI, TU Dortmund, Dortmund, Germany**
[3]**Faculty of Applied Chemistry, Reutlingen University, Reutlingen, Germany**
[4]**Genome Informatics, Institute of Human Genetics, University Hospital Essen, University of Duisburg-Essen, Essen, Germany**
[*]These authors contributed equally to this work

ABSTRACT

Ion mobility spectrometry (IMS) is a technology for the detection of volatile compounds in the air of exhaled breath that is increasingly used in medical applications. One major goal is to classify patients into disease groups, for example diseased versus healthy, from simple breath samples. Raw IMS measurements are data matrices in which peak regions representing the compounds have to be identified and quantified. A typical analysis process consists of pre-processing and peak detection in single experiments, peak clustering to obtain consensus peaks across several experiments, and classification of samples based on the resulting multivariate peak intensities. Recently several automated algorithms for peak detection and peak clustering have been introduced, in order to overcome the current need for human-based analysis that is slow, subjective and sometimes not reproducible. We present an unbiased comparison of a multitude of combinations of peak processing and multivariate classification algorithms on a disease dataset. The specific combination of the algorithms for the different analysis steps determines the classification accuracy, with the encouraging result that certain fully-automated combinations perform even better than current manual approaches.

Keywords: Ion mobility spectrometry, peak detection, clustering, classification

INTRODUCTION

Ion mobility (IM) spectrometry (IMS), coupled with multi-capillary columns (MCCs) allows to detect the presence and measure the concentration of volatile organic compounds (VOCs), e.g., certain small metabolites, in the air or in exhaled breath. As MCC/IMS works at ambient pressure and temperature and thus requires no vacuum pump system, commercial devices are less expensive to obtain and to operate than, for example, mass spectrometers coupled to a gas chromatograph. Consequently, MCC/IMS technology is increasingly used in medical and biotechnological applications with the goal of detecting marker metabolites that occur under specific conditions, such as lung diseases. An overview of advances in breath research is presented by Fink et al. (2014).

In an MCC/IMS experiment, a mixture of VOCs is physically separated in two dimensions: first by retention time r in the MCC (the time required for a particular compound to pass through the column) and then by drift time d through the IM spectrometer. Retention times are measured in seconds, whereas drift times are measured in milliseconds. Instead of the drift time itself, a quantity normalized for pressure, temperature, electric field strength and drift tube length, called the *inverse reduced mobility* (IRM) t, is used. Its units are $\mathrm{Vs}/\mathrm{cm}^2$, and there is a proportionality factor f_{ims} that depends on the above conditions such that $t = f_{\mathrm{ims}} \cdot d$. For further information on the technology see Cumeras et al. (2015a) and Cumeras et al. (2015b).

Let R be the set of (equidistant) retention time points and let T be the set of (equidistant) IRMs where a measurement is made. The data obtained from a single experiment is an $|R| \times |T|$ matrix $S = (S_{r,t})$ of measured ion intensities, which we call an *IM spectrum-chromatogram* (IMSC). The matrix can be visualized as a heat map (Figure 1). A row of S is a *spectrum*, while a column of S is a *chromatogram*. Areas of high intensity in S are called *peaks*. A peak can be described by a few descriptors: its location (r,t), its shape (by appropriate parameters) and its intensity. Comparing

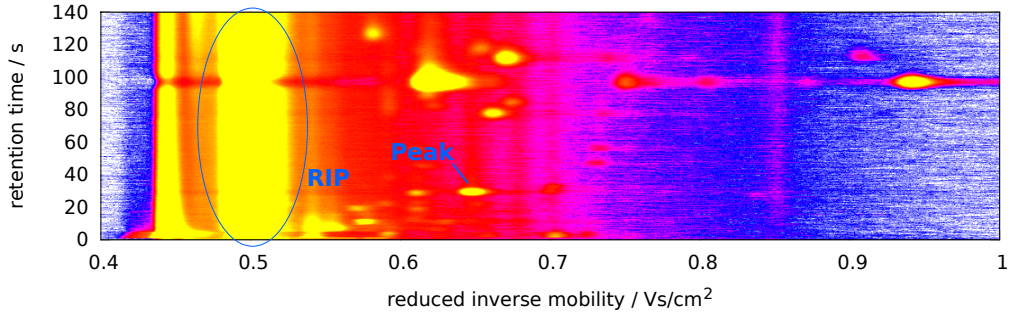

Figure 1. Visualization of a raw measurement (IMSC) as a heat map. Intensity color: white (lowest) < blue < purple < red < yellow (highest). The reactant ion peak (RIP) near 0.5 Vs/cm^2 and a VOC peak are annotated. Image reproduced from Kopczynski and Rahmann (2015).

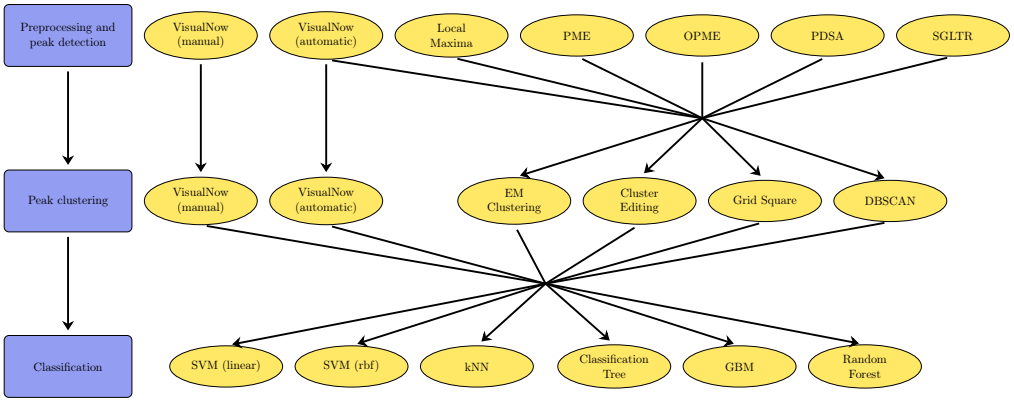

Figure 2. Possible combinations for the three steps in the analysis process from raw data to classification results. Not all peak detection and peak clustering methods can be combined. All results from peak detection are input for all classification methods.

peak coordinates with reference databases may reveal the identity of the corresponding compound. Complications in MCC/IMS analysis arise from noise in the data and from the constant presence of the so-called reactant ion peak (RIP) in each spectrum at an IRM of approximately 0.48 to 0.5 Vs/cm^2.

In this article, we examine the data analysis process from raw IMS measurement data via peak detection to classification results. This process consists of three major steps:

1. pre-processing and peak detection in each single experiment,
2. peak clustering to obtain consensus peaks across several experiments,
3. classification of samples and detection of class-specific peaks.

For each of these steps, several methods have recently been described in the literature, especially for peak detection and peak modeling (Bödeker et al., 2008; Jünger et al., 2010; D'Addario et al., 2014; Kopczynski and Rahmann, 2015).

So far, these automated methods are not widely used, and the state of the art in MCC/IMS analysis is semi-automated manual annotation of peaks and consensus peaks.

Additionally, the effect of *interactions* between different algorithms for all three steps of the process on the final classification results has not been studied in detail. The work by Hauschild et al. (2013) does study a similar question, but naturally does not cover recently developed peak detection methods and also only a smaller selection of classifiers.

Thus the main purpose of this article is to provide an extensive comparison between manual and automated analysis methods of MCC/IMS datasets, not only focusing on single methods but additionally on their interactions when combining them into pipelines.

In the following, we first give a short description of the used dataset. The next three sections summarize each method evaluated on each of the three levels of the data analysis process. Our evaluation takes all meaningful combinations of methods into account (see Figure 2 for an overview). We report the performance of each method combination and discuss observable trends.

DATA

We evaluate the methods on a two-class dataset, with infected and healthy patients. Exhaled breath was measured in 30 patients whose airways are either infected or colonized by *Pseudomonas aeruginosa* and in 37 healthy non-smoker controls. All patients were recruited from the Department of Pulmonology, Ruhrlandklinik, University Hospital of Essen, Germany, with no evidence of acute exacerbation for at least 4 weeks prior to enrollment. Diagnosis of Pseudomonas was established according to up-to-date guidelines. The healthy controls were employees of the hospital. The study was approved by the ethic committee of the University of Essen, with informed consent of all subjects. On an only slightly different dataset, Rabis et al. (2011) identified single peaks with differential intensities between the two groups.

PEAK DETECTION AND PRE-PROCESSING

Pre-processing is integrated into most peak detection methods and based on two basic ideas (Bader et al., 2008), which we mention first. The raw data is smoothed and de-noised by applying a threshold, a low-pass filter or a smoothing kernel such as the Savitzky-Golay filter (Savitzky and Golay, 1964). To compensate for the RIP, one estimates its shape (either from a spectrum without VOC peaks or by fitting model functions) and subtracts it from the measured spectra.

We now describe all evaluated automated peak detection methods. The manual detection method combines the two steps peak detection and peak clustering within one step. Hence this detection is described in the section on "Peak Clustering Methods".

The following six peak detection methods can be divided into two categories. The *automated detection in VisualNow* as well as *local maxima* (LM) and *peak model estimation* (PME) are offline methods, where the whole data matrix of an IMSC is available at any time during the entire detection process. All remaining methods, i.e. *peak detection by slope analysis* (PDSA), *Savitzky-Golay Laplace-operator filtering thresholding regions* (SGLTR) and *online peak model estimation* (OPME), are online methods, where the single (or a small constant amount of) IM spectra are processed right after capturing and are then directly discarded. Storing the whole matrix then becomes obsolete; this is a desirable property especially for resource-constrained embedded devices.

In addition, the methods PME and OPME provide not only the position and signal value of the peaks mode but several parameters describing a whole peak shape with a statistical model.

Automated Detection in VisualNow VisualNow is a commercial program for the analysis of IMSC data based on the developments of Bödeker et al. (2008). It provides an automated peak extraction method which was introduced by Bader et al. (2005). Having an IMSC data matrix, the first step is performed by a k-means algorithm. Each cell of the ISMC matrix is labeled as "peak" or "non-peak" based on k-means clustering of the intensity values into high and low. In the second step the matrix is first processed row-wise. Neighboring cells having the label "peak" are considered as one row unit. Having determined all runs in each row, the matrix is processed column-wise. Adjacent runs of two neighboring rows are merged. In a last step all centroids of the emerged regions are computed.

Local Maxima (LM) This approach from the PEAX framework (D'Addario et al., 2014) marks each value that exceeds a given noise threshold and all of its eight neighbor values as a peak candidate. In the second step peak candidates that are too close to each other and thus cannot be explained by the underlying physical process of the detector are merged by a Weighted Cluster Editing algorithm (see "Cluster Editing" below). Detected clusters of peak candidates are merged into final peaks. When merging two or more candidates, both the position and the signal value of the highest candidate is assigned to the final peak.

Peak Model Estimation (PME) Similar to the LM approach, PME finds peak candidates by estimating (smoothed) derivatives in retention time and IRM direction and reporting local maxima. Candidates with too low signal value are discarded. For the remaining candidates, an augmented version of the EM algorithm (for more details consider Section "EM Clustering") is executed. Finally also an EM algorithm estimates parameters for a statistical model describing the shape of each final peak with seven parameters. This peak model was introduced by Kopczynski et al. (2012).

Peak Detection by Slope Analysis (PDSA) PDSA is an online algorithm and was first described by Egorov et al. (2013). A sliding window with a fixed width is used to examine each single spectrum for a specific pattern called a "run". A run begins when the sum of all values within the window exceeds a threshold computed based on an area that contains only noise. The run continues as long as the sum for the previous window is smaller than the sum for the current one. The difference between

two sums (i.e., an estimate of the derivative) is being tracked. A run ends when the derivative is decreasing and the intensity sum drops below the noise threshold. Runs with more than one change of the sign of the derivative are discarded. Runs are merged into peaks over different retention time when certain distance criteria stated by Hauschild et al. (2013) are fulfilled.

Savitzky-Golay Laplace-operator filtering thresholding Regions (SGLTR) The second online detection method described by Egorov et al. (2013) considers a small set of consecutive spectra at once. The Laplace operator yields the sum of both second partial derivatives in retention time IRM dimension, i.e., a measure for the curvature of the intensity function at each coordinate (r,t). Both Laplace operator and first derivatives may be estimated by a Savitzky-Golay filter. Peaks can be inferred where the intensities exceed the noise level, the first derivatives are close to zero and the curvature is strongly negative.

Online Peak Model Estimation (OPME) This online method was introduced by Kopczynski and Rahmann (2015). It provides a parameter set for every peak describing the peak shape as mentioned, similarly to the (offline) Peak Model Estimation (PME) method. For every IM spectrum within a moving window of fixed width, a second order polynomial is fitted by least squares regression. Having found an appropriate position, parameters for a parametric function describing one-dimensional peaks in IRM are computed using the polynomial. To connect the one-dimensional models of two consecutive spectra, a modified version of a global alignment is utilized. Finally, parameters for a parametric function describing one-dimensional peaks in retention time are computed using the set of n one-dimensional models for every peak, also by fitting a second order polynomial.

PEAK CLUSTERING METHODS

Assuming that not a single IMSC is being analyzed but a series of several entities measured under the same condition, e.g. patients suffering from the same disease, it is important to determine if two peaks from different measurements at similar positions should be considered as the "same" peak (i.e., may result from the same analyte). In the following, a *peak* refers to a detected peak in a single measurement, whereas a *consensus peak* refers to a set (cluster) of peaks in different measurements at approximately the same position that an algorithm has decided to belong together.

This section describes the used peak clustering methods that produce consensus peaks from n peak lists from n measurements. An individual peak is given by its measurement number i, its coordinates (r,t) and its intensity y. The output is an intensity matrix (often simply called *data table*) $D = (D_{ij})$, such that D_{ij} is the intensity of consensus peak j in measurement i.

In the following we describe the manual annotation of peaks, as well as four clustering methods, namely *Grid Square*, *Density-Based Spatial Clustering of Applications with Noise*, *Cluster Editing*, and *EM Clustering* providing a data table. All chosen clustering methods do not require a predefined number of clusters, but determine the number dynamically.

Manual Peak Detection and Clustering Manual peak detection and clustering with the VisualNow software is a single step. Having a set of IMSC measurements, quadratic regions can be drawn interactively over a visualization of an arbitrary data matrix within the set. One can view the resulting subsections of the matrix in each measurement and interactively move the regions and add or remove consensus peaks. The final regions must not overlap, and the highest value within a region is considered as center of a cluster.

Automated Clustering in VisualNow VisualNow can automatically cluster a set of measurements with a method based on k-means clustering (Bödeker et al., 2008).

Grid Square This fast and simple approach partitions the coordinate system of consensus peaks into disjoint rectangular regions. Each region corresponds to a potential consensus peak and has fixed width and linearly increasing height depending on the retention time, as stated by Hauschild et al. (2013). For every peak, its associated region can be computed in constant time. Empty regions or regions with too few peaks are discarded. For every remaining region, the average position of all contained peaks is computed and assigned to a consensus peak.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) Introduced by Ester et al. (1996), DBSCAN is a widely used clustering algorithm in data mining. Starting with a random point p, the algorithm considers all neighbors of p within a fixed distance ε. When the number of neighbors exceeds a threshold `minPts`, p is put into a new cluster, otherwise it is labeled as noise. Continuing with the neighbors, their respective neighbors are determined and put into the same cluster, when `minPts` is exceeded. When no remaining unvisited point can be reached, the cluster is

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1294v1 | CC-BY 4.0 Open Access | rec: 11 Aug 2015, publ: 11 Aug 2015

closed and the algorithm starts again with an unvisited point. DBSCAN makes no assumptions about possible underlying distribution of the points building a cluster. Thus, no features like centroids are provided in the original approach. We additionally add centroids and compute the average position in both retention time and IRM and consider them as consensus peaks.

Cluster Editing The weighted cluster editing approach (Rahmann et al., 2007; Böcker et al., 2011) finds a set of disjoint cliques (consensus peaks) from (noisy) input data points (peaks). The relationship between points is given by a similarity weight function that can be positive (resulting in a similarity edge between the points) or negative. For peaks, the similarity weight function is based on the (r,t) distance between peaks. Cluster editing allows to add or remove edges from the so constructed input graph, and the cost of each such operation is proportional to the absolute weight. The goal is to find the minimum cost partition into disjoint cliques. The problem is NP hard, but fixed parameter tractable, and practical implementations exist. We utilized the software yoshiko[1].

EM Clustering The original EM algorithm introduced by Dempster et al. (1977) is a statistical method to estimate parameters of parametric mixture models by estimating the membership of every data point to every model component. As a by-product a soft clustering is provided. The standard EM algorithm requires a fixed number of models. An improved version presented by Kopczynski and Rahmann (2015) adjusts the number of models dynamically. It is assumed that peaks are normally distributed around a center with known standard deviation in both retention time and IRM. At the start of the EM Clustering method as much models as peaks are initialized. Between expectation and maximization step, a new merging step is introduced. This method is the only which provides the position of its centroids innately.

CLASSIFICATION METHODS

Each data table D resulting after peak clustering contains n observations and a differing number of k consensus peaks. We apply different classification algorithms to each data table D. For an overview see Hastie et al. (2009) and for explicit implementations the references after the descriptions of the methods. The goal is to identify the combinations of peak finding, peak clustering, and classification algorithms with highest classification accuracy. We use 10-fold cross-validation (CV), that is, we split our observations in 10 equally sized groups, train our classification algorithm sequentially on 90% of the data (training set) and predict the left out 10th partition of the data (test set) with the resulting model. In case of necessary parameter tuning for the classification algorithm a nested 10-fold CV is performed. Since the results of the CV depend on the 10 random splits, we repeat each procedure of classification 50 times. We use accuracy in the first place and additionally sensitivity, specificity and AUC as performance measures. Classification was performed using the statistical programming language R, version 3.1.2 (2014-10-31) (R Core Team, 2014). If not stated otherwise, default parameter settings of the implementations were used.

Support Vector Machine (SVM) A common classification algorithm is the Support Vector Machine. The aim is to find a boundary to separate the observations of two classes in a way that the points are as far away from the boundary as possible. R-package: e1071 (Meyer et al., 2014).

Linear SVM The simplest SVM is the linear SVM which determines a hyperplane to separate the data points. Since the observations of the groups are usually not linearly separable it is necessary to penalize observations lying on the wrong side of the boundary with a cost parameter C. C is a tuning parameter and 10 values between 2^{-15} with 2^{15} were tried, with equidistant step size on the exponential scale.

Radial Basis Function (RBF) SVM Since a linear decision boundary might often not be suitable, other boundaries can be achieved by transforming the data into a higher dimension and then searching for a linear boundary in this space. To optimize this boundary the algorithm does not have to actually compute the transformations. A so called kernel function calculates the necessary dot products between two transformed observations without computing the actual values first. The used kernel function is the Radial Basis Function (RBF) with $K(x_i,x_j) = \exp\{-\gamma|x_i - x_j|^2\}$ for two points x_i and x_j. The second tuning parameter besides the cost parameter C is γ and the same 10 values between 2^{-15} and 2^{15} were evaluated in the inner CV.

[1] http://www.mi.fu-berlin.de/w/LiSA/YoshikoCharles

K-Nearest-Neighbor (KNN) A simple idea to classify a new observation is to assign it to the class most of its closest neighbors belong to. The k closest points (in terms of Euclidean distance) decide per majority vote. The number of considered neighbors k is treated as tuning parameter, taking the integers from one to ten in the inner CV. R-package: kknn (Schliep and Hechenbichler, 2014)

Classification Tree (CT) In a two-class classification problem, a Classification Tree splits the training set into two groups using a simple cut-point on a single variable. The procedure proceeds iteratively for the two resulting sets, which means that they are also split regarding one variable, until a stopping criterion is reached. A set of observations in the tree is called node. To decide which variable and which cut-point should be used at each step, the Gini index is used to measure the decrease of impurity achieved by a certain split. A node is considered pure if it contains only observations of the same class.

If the node impurity does not improve by a factor of 0.01, the node is not split any more. Since large trees with lots of splits are prone to overfitting, the size of a terminal node that is not split any further can be restricted. The variable *minbucket* specifies the minimum number of observations in a terminal node. It is considered as a tuning parameter and all integer values between 1 and 5 were tried. R-package: rpart (Therneau et al., 2015)

Generalized Boosted Models (GBM) Boosting is a strategy to combine many weak learners into a strong one. Here, 100 simple trees with just few nodes (interaction depth was considered as tuning parameter with values from one to three) are iteratively added. The next tree is chosen such that it minimizes a loss criterion (here the deviance, the negative log-likelihood of the Bernoulli model) based on the current model. To determine the next tree, just 50% of the training data is used. The tuning parameter *shrinkage* (values between 2^{-15} and 2^{15}) controls the step size, the parameter *n.minobsinnode* (values from 1-5) how many observations have to be in a terminal node. R-package: gbm (Ridgeway and with contributions from others, 2013)

Random Forest (RF) A Random Forest is a collection of many (here 500) trees, each of which is based only on a random selection of $\sqrt{k}$ of the available variables and a bootstrap sample of size n. Each tree is fully grown, that means that there are no restrictions of minimum node size. To classify an observation it is passed trough all 500 trees and the class is assigned by majority vote. R-package: randomForest (Liaw and Wiener, 2002)

RESULTS

We now evaluate all meaningful combinations of peak detection, peak clustering, and classification algorithms on the Pseudomonas dataset; Figure 2 illustrates the resulting 156 analysis processes. Due to this large number, we split the evaluation into different parts. First we look at each analysis step separately, irrespective of the other steps. Then we identify and discuss the best specific combinations. Remember that our focus is on classification accuracy based on the identified peak intensities and not the quality of the peak detection itself.

Separate evaluation of each step After peak detection and peak clustering we have 26 datasets of peak intensities that are input for the remaining classification step. The number of consensus peaks and thus variables varies over the datasets. Whereas the VN variants find many consensus peaks, 224 for VN (manual) and 236 for VN (auto) combined with its own peak clustering method, the other methods identify much less, from 11 for OPME combined with Grid Square up to 67 for SGLTR combined with EM Clustering. This may indicate that the automated versions often miss or merge existing peaks and thus corresponding metabolites in the exhaled air. Still we will see that classification accuracy does not break down.

Table 1 shows common measures of location and dispersion, namely the minimum, maximum, median, mean and standard deviation of all accuracies that are achieved. Here one of the three steps is fixed and the measures are calculated across all combinations of the other steps with the specific method and across the 50 CV replications. One should not over-interpret these results, especially the extreme values, since all combinations are included, also the worst ones with the two other steps. As we will see later on, certain combinations are much better. The statistics serve as a summary of the single steps, combined with "medium" methods from the remaining steps.

Most peak detection methods could be combined with four peak clustering methods and their measures are therefore based on 1200 ($4 \cdot 6 \cdot 50$) accuracies. VN (automatic) was also combined with its own peak clustering method leading to 1500 values, whereas VN (manual) is fixed in steps 1 and 2 and thus generates only 300 values. The upper table shows that the best median accuracies above 0.8 are achieved by LM and VN (manual) with 0.817 and VN (auto) with 0.806. The small standard

Table 1. Summaries of accuracies for all methods of the three analysis process steps (top: peak detection, middle: peak clustering, bottom: classification), aggregated across all possible combinations in which they are involved.

	LM	PME	PDSA	SGLTR	OPME	VN (auto)	VN (manual)
Minimum	0.612	0.507	0.701	0.627	0.433	0.657	0.731
Maximum	0.970	0.970	0.866	0.955	0.866	0.955	0.881
Median	0.821	0.761	0.791	0.791	0.761	0.806	0.821
Mean	0.806	0.769	0.788	0.808	0.739	0.816	0.817
Stand. dev.	0.081	0.107	0.025	0.068	0.074	0.050	0.038

	Grid Square	DBSCAN	Cluster Editing	EM Clustering	VN (auto)	VN (manual)
Minimum	0.433	0.567	0.507	0.612	0.731	0.731
Maximum	0.970	0.955	0.955	0.955	0.925	0.881
Median	0.761	0.791	0.791	0.791	0.851	0.821
Mean	0.762	0.799	0.783	0.800	0.850	0.817
Stand. dev.	0.098	0.065	0.072	0.058	0.040	0.038

	SVM (linear)	SVM (rbf)	kNN	CT	RF	GBM
Minimum	0.552	0.507	0.537	0.478	0.627	0.433
Maximum	0.910	0.896	0.910	0.940	0.970	0.970
Median	0.776	0.761	0.761	0.806	0.836	0.836
Mean	0.759	0.751	0.753	0.803	0.838	0.835
Stand. dev.	0.056	0.064	0.074	0.065	0.056	0.075

deviation for VN (manual) is due to the facts that it was not combined with other methods that could influence the performance and that only 300 values were taken into account.

The middle table shows the same results keeping the peak clustering method fixed. All measures are based on 1800 ($6 \cdot 6 \cdot 50$) values, except for both VN variants which could not be combined with other peak detection methods and therefore lead to 300 accuracies. The VN variants achieve the best median accuracies, 0.851 for the automatic and 0.821 for the manual version. Grid Square has the lowest value (0.761), whereas all others have the same median of 0.791.

The last step are the classification methods whose measures are shown in the bottom table in Table 1. The best methods are the tree based classification algorithms, especially RF and GBM, both with median accuracy of 0.836.

Evaluation of combinations of all steps We now consider all 156 analysis processes separately. Figure 3 shows the results in terms of accuracy. Each of the 156 boxplots represents one of the combinations of peak detection, peak clustering and classification. Each box is based on 50 points, the replications of the 10-fold CV.

The six panels contain the results of the different classification algorithms. The horizontal line at an accuracy of 0.8 serves as orientation. Most of the boxes for both variants of SVMs lie under this line. The median accuracies of kNN are more spread out with values roughly between 0.6 and 0.85. The boxes for the three tree based algorithms are overall higher than the other boxes, so they outperform the SVMs or kNN on this dataset. The extensions RF and GBM in turn outperform the simple Classification Tree for many combinations of peak detection. The boxes of the RF are a little shorter than those of the GBM, so the results are more stable and less depending on the splits of the CV.

In each panel of Figure 3 the boxes are ordered by the peak detection methods (annotation on the x-axis) and colored by peak clustering methods. Considering just RF and GBM, the best peak detection methods are combinations with LM, PME and SGLTR depending on the peak clustering methods. Considering only these classification and peak detection methods, mostly Grid Square leads to the best accuracies. The combination of OPME and Grid Square seems to be considerably inferior to all other combinations, with a lower median and higher variability of accuracy.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1294v1 | CC-BY 4.0 Open Access | rec: 11 Aug 2015, publ: 11 Aug 2015

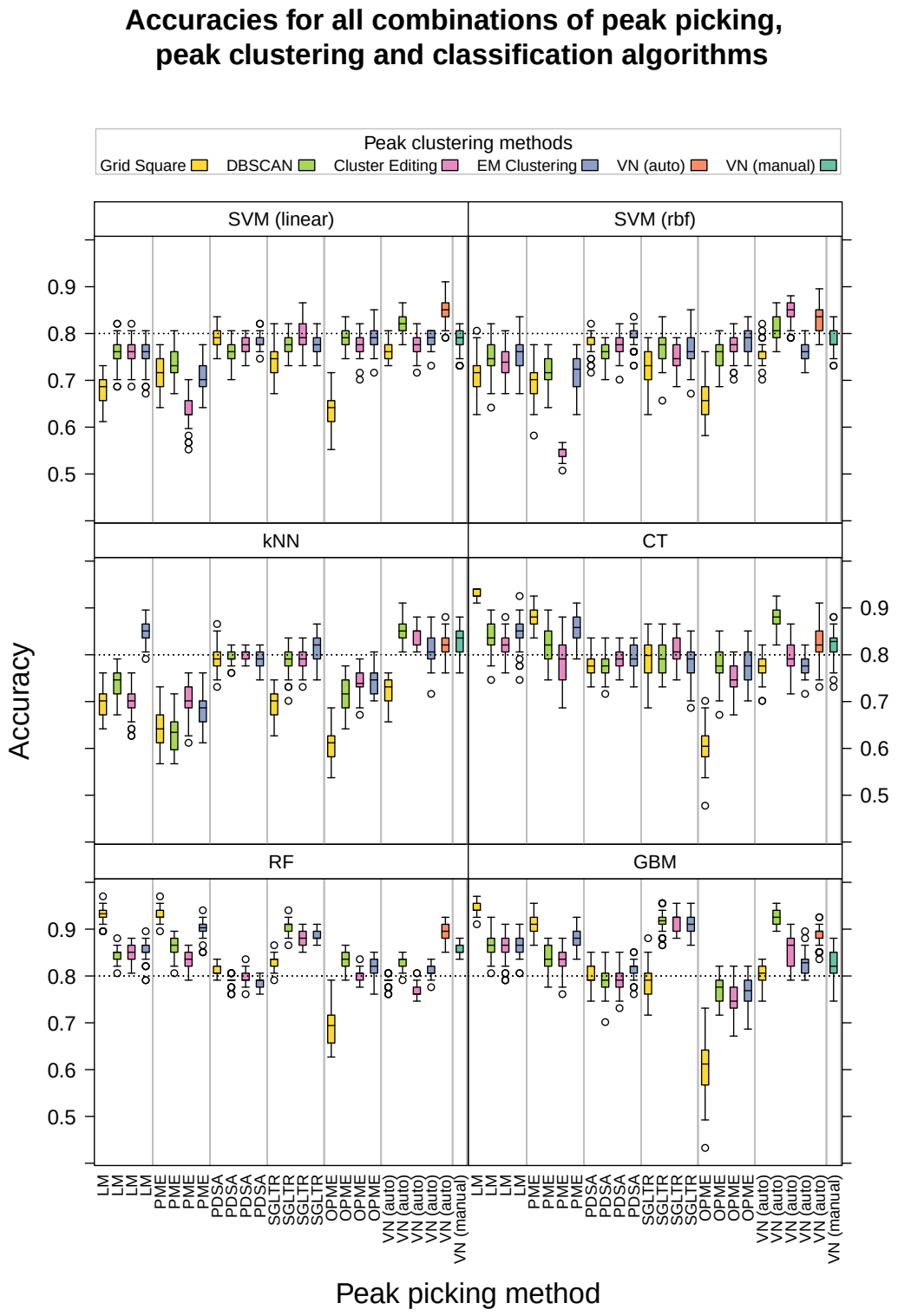

Figure 3. Accuracies for all combinations of peak detection, peak clustering and classification algorithms

Table 2. Median performance measures for the best combinations of peak detection, peak clustering and classification algorithms in terms of accuracy

Peak detection	Peak clustering	Classification	Accuracy	Sensitivity	Specificity	AUC
LM	Grid Square	GBM	0.940	0.967	0.946	0.960
LM	Grid Square	RF	0.933	0.900	0.946	0.977
LM	Grid Square	CT	0.925	0.933	0.919	0.927
PME	Grid Square	RF	0.925	0.933	0.919	0.985
SGLTR	Cluster Editing	GBM	0.925	0.867	0.973	0.974
VN (auto)	DBSCAN	GBM	0.925	0.867	0.973	0.921
SGLTR	DBSCAN	GBM	0.918	0.833	0.973	0.979
PME	Grid Square	GBM	0.910	0.900	0.919	0.930
SGLTR	DBSCAN	RF	0.910	0.833	0.973	0.980
SGLTR	EM Clustering	GBM	0.910	0.867	0.973	0.973
PME	EM Clustering	RF	0.903	0.900	0.919	0.918
VN (auto)	VN (auto)	RF	0.896	0.767	0.986	0.964

To summarize the best results, the twelve combinations with the best median accuracy values (between 0.896 and 0.940) are displayed in Table 2. The fully manual procedure (VN manual) is not among the best methods, but combined with the classification algorithm RF it achieves an accuracy of 0.851 and is ranked at position 25 out of 156. In terms of median accuracy this means that the best method (LM combined with Grid Square and GBM) with an accuracy of 0.94 classifies on average 6 more out of 67 observations correctly than VN (manual) with RF.

At first sight it is a little surprising that the results from Table 1 do not coincide with the results of single combinations. From the first overview we would have preferred either VN (manual) or VN (automatic), but definitely not Grid Square. Although the VN variants were competitive in the end, the failure of the combination OPME with Grid Square and the SVMs or kNN (compare Figure 3) obscure the excellent results achieved when combining Grid Square with LM and PME and the tree based classification algorithms. This outlines the importance of comparing whole processes instead of single steps.

Some AUC values achieve extremely high values up to 0.98. During classification the cut-point for assigning an observation to a class was 0.5 for the predicted probability. Sometimes it is worth adjusting this value in order to achieve better classification results. This should be one objective for future research. Taking a look at sensitivity and specificity, it can be seen that they are mostly either equal or that specificity is considerably higher, potentially caused by unequal sample sizes. The control group was larger (37 compared to 30), hence a model trained on accuracy is better if it is good at assigning the control cases to the right class. This can be influenced by varying the cut-point as explained above.

DISCUSSION AND CONCLUSION

We presented a detailed comparison of analysis processes starting with raw MCC/IMS measurements to classification of diseases, using a two-class dataset (Pseudomonas patients vs. healthy controls). Manual peak picking was compared against 25 automated methods and combined with six different classification methods, using 50 times repeated 10-fold cross-validation. The aim was to find out which combinations of peak detection, peak clustering and classification can distinguish the two groups most accurately and how automated peak detection methods compare to manual ones.

Our results indicate that specific automatic peak defining methods can result in classification accuracies as good as the manual procedure. This does not imply that these methods are better at peak detection itself. The best peak defining method combinations often included LM, SGLTR, VN (auto) or PME for the step peak detection, and Grid Square, DBSCAN and EM Clustering for the step peak clustering. The best results for classification were achieved by tree based methods, especially by GBM and RF. It is encouraging that (fast) automated methods can keep up with the current state of the art and that even online methods for peak detection achieve good results.

These results were obtained using just a single dataset with 67 observations. Although the classification accuracies are unbiased due to cross-validation and also clear tendencies in favor of certain methods are observable, the choice of the "best" combinations is overoptimistic and lacks external validation. To validate the success of these combinations, application on more datasets is required, and we will extend our evaluation accordingly in the future.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1294v1 | CC-BY 4.0 Open Access | rec: 11 Aug 2015, publ: 11 Aug 2015

REFERENCES

Bader, S., Urfer, W., and Baumbach, J. I. (2005). Processing ion mobility spectrometry data to characterize group differences in a multiple class comparison. *International Journal for Ion Mobility Spectrometry*, 8:1–4.

Bader, S., Urfer, W., and Baumbach, J. I. (2008). Preprocessing of ion mobility spectra by lognormal detailing and wavelet transform. *International Journal for Ion Mobility Spectrometry*, 11(1-4):43–49.

Böcker, S., Briesemeister, S., and Klau, G. W. (2011). Exact algorithms for cluster editing: Evaluation and experiments. *Algorithmica*, 60(2):316–334.

Bödeker, B., Vautz, W., and Baumbach, J. I. (2008). Peak finding and referencing in MCC/IMS-data. *International Journal for Ion Mobility Spectrometry*, 11(1):83–87.

Bödeker, B., Vautz, W., and Baumbach, J. I. (2008). Visualisation of MCC/IMS-data. *International Journal for Ion Mobility Spectrometry*, 11(1-4):77–81.

Cumeras, R., Figueras, E., Davis, C. E., Baumbach, J. I., and Grácia, I. (2015a). Review on ion mobility spectrometry. part 1: current instrumentation. *Analyst*, 140:1391–1410.

Cumeras, R., Figueras, E., Davis, C. E., Baumbach, J. I., and Grácia, I. (2015b). Review on ion mobility spectrometry. part 2: hyphenated methods and effects of experimental parameters. *Analyst*, 140:1376–1390.

D'Addario, M., Kopczynski, D., Baumbach, J. I., and Rahmann, S. (2014). A modular computational framework for automated peak extraction from ion mobility spectra. *BMC Bioinformatics*, 15(1):25.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Egorov, A., König, A., Köppen, M., Kühn, H., Kullack, I., Kuthe, E., Mitkovska, S., Niehage, R. Pawelko, A., Sträßer, M., and Striewe, C. (2013). Ressourcenbeschränkte Analyse von Ionenmobilitätsspektren mit dem Raspberry Pi. Technical report, Faculty of computer science, TU Dortmund.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining (KDD), Proceedings of first international conference*, volume 96, pages 226–231.

Fink, T., Baumbach, J., and Kreuer, S. (2014). Ion mobility spectrometry in breath research. *Journal of Breath Research (J. Breath Res.)*, 8(2):027104.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning, Second Edition*. Springer Series in Statistics, New York, USA.

Hauschild, A. C., Kopczynski, D., D'Addario, M., Baumbach, J. I., Rahmann, S., and Baumbach, J. (2013). Peak detection method evaluation for ion mobility spectrometry by using machine learning approaches. *Metabolites*, 3(2):277–293.

Jünger, M., Bödeker, B., and Baumbach, J. I. (2010). Peak assignment in multi-capillary column–ion mobility spectrometry using comparative studies with gas chromatography–mass spectrometry for voc analysis. *Analytical and bioanalytical chemistry*, 396(1):471–482.

Kopczynski, D., Baumbach, J., and Rahmann, S. (2012). Peak modeling for ion mobility spectrometry measurements. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1801–1805, New York, NY, USA. IEEE.

Kopczynski, D. and Rahmann, S. (2015). An online peak extraction algorithm for ion mobility spectrometry data. *Algorithms for Molecular Biology*, 10(1):17.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2014). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6-4.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rabis, T., Sommerwerck, U., Anhenn, O., Darwiche, K., Freitag, L., Teschler, H., Bödeker, B., Maddula, S., and Baumbach, J. I. (2011). Detection of infectious agents in the airways by ion mobility spectrometry of exhaled breath. *International Journal for Ion Mobility Spectrometry (Int. J. Ion Mobility Spectrom.)*, 14:187–195.

Rahmann, S., Wittkop, T., Baumbach, J., Martin, M., Truss, A., and Böcker, S. (2007). Exact and heuristic algorithms for weighted cluster editing. In *Computational Systems Bioinformatics Conference*, volume 6, pages 391–401.

Ridgeway, G. and with contributions from others (2013). *gbm: Generalized Boosted Regression Models*. R package version 2.1.

Savitzky, A. and Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639.

Schliep, K. and Hechenbichler, K. (2014). *kknn: Weighted k-Nearest Neighbors*. R package version 1.2-5.

Therneau, T., Atkinson, B., and Ripley, B. (2015). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-9.