Kopczynski and Rahmann *Algorithms for Molecular Biology* (2015) 10:17
DOI 10.1186/s13015-015-0045-5

AMB ALGORITHMS FOR MOLECULAR BIOLOGY

RESEARCH Open Access

An online peak extraction algorithm for ion mobility spectrometry data

Dominik Kopczynski[1] and Sven Rahmann[1,2*]

Abstract

Ion mobility (IM) spectrometry (IMS), coupled with multi-capillary columns (MCCs), has been gaining importance for biotechnological and medical applications because of its ability to detect and quantify volatile organic compounds (VOC) at low concentrations in the air or in exhaled breath at ambient pressure and temperature. Ongoing miniaturization of spectrometers creates the need for reliable data analysis on-the-fly in small embedded low-power devices. We present the first fully automated online peak extraction method for MCC/IMS measurements consisting of several thousand individual spectra. Each individual spectrum is processed as it arrives, removing the need to store the measurement before starting the analysis, as is currently the state of the art. Thus the analysis device can be an inexpensive low-power system such as the Raspberry Pi.

The key idea is to extract one-dimensional peak models (with four parameters) from each spectrum and then merge these into peak chains and finally two-dimensional peak models. We describe the different algorithmic steps in detail and evaluate the online method against state-of-the-art peak extraction methods.

Keywords: Ion mobility spectrometry, Peak detection, Automated data analysis, Online analysis

Introduction

Ion mobility (IM) spectrometry (IMS), coupled with multi-capillary columns (MCCs), MCC/IMS for short, has been gaining importance for biotechnological and medical applications. With MCC/IMS, one can measure the presence and concentration of volatile organic compounds (VOCs) in the air or in exhaled breath with high sensitivity. In contrast to other technologies, such as mass spectrometry coupled with gas chromatography (GC/MS), MCC/IMS works at ambient pressure and temperature. Several diseases like chronic obstructive pulmonary disease (COPD) [1], sarcoidosis [2] or lung cancer [3] can potentially be diagnosed early with MCC/IMS technology. IMS is also used for the detection of drugs [4] and explosives [5]. Constant monitoring of VOC levels is of interest in biotechnology, e.g., for watching fermenters with yeast producing desired compounds [6] and in medicine, e.g., monitoring propofol levels in the exhaled breath of patients during surgery [7].

IMS technology is moving towards miniaturization and small mobile devices. This creates new challenges for data analysis: The analysis should be possible *within* the measuring device without requiring additional hardware like an external laptop or a compute server. Ideally, the spectra can be processed on a small embedded chip or small device like a Raspberry Pi or similar hardware with restricted resources. Algorithms in small mobile hardware face constraints, such as the need to use little energy (hence little random access memory), while maintaining prescribed time constraints.

The basis of MCC/IMS analysis is *peak extraction*, by which we mean a representation of all high-intensity regions (peaks) in the measurement by using a few descriptive parameters per peak instead of the full measurement data. State-of-the-art software (like IPHEx [8], Visual Now [9], PEAX [10]) only extracts peaks when the whole measurement is available, which may take up to 10 minutes because of the pre-separation of the analytes in the MCC. Our own PEAX software in fact defines modular pipelines for fully automatic peak extraction and compares favorably with a human domain expert doing the same work manually when presented with a whole MCC/IMS measurement. However, storing the whole

*Correspondence: Sven.Rahmann@uni-due.de
[1] Bioinformatics for High-Throughput Technologies, Computer Science XI, and Collaborative Research Center SFB 876, TU Dortmund, Dortmund, Germany
[2] Genome Informatics, Institute of Human Genetics, University Hospital Essen, University of Duisburg-Essen, Essen, Germany

measurement is not desirable or possible when the memory and CPU power is restricted. Here we introduce a method to extract peaks and estimate a parametric representation while the measurement is being captured. This is called *online peak extraction*, and this article presents the first algorithm for this purpose on MCC/IMS data. An extended abstract of this work has been published at WABI'14 [11].

Section 'Background' introduces the necessary background on the data produced by an MCC/IMS experiment, on peak modeling and on optimization methods. The basic idea of our algorithm is to process each IM spectrum as soon as it arrives (and before the next one arrives). After appropriate pre-processing including denoising and baseline correction described in Section 'Denoising and baseline correction', the single spectra are reduced into a mixture of parametric one-dimensional peak models, described in Section 'Reducing a spectrum to peak models'. Accordingly, in Section 'Aligning consecutive spectrum peak lists' the approach of connecting models from two subsequent spectra into peak chains is explained. The main challenge is then to merge the peak chains into two-dimensional peak models, described in Section 'Estimating 2-D peak models'. In Section 'Peak clustering' we introduce a novel approach for clustering peaks among several measurements e.g. for time series. An evaluation of our approach is presented in Section 'Evaluation' including a listing of all settings of the MCC/IMS as well as an explanation of all adjustable parameters, while Section 'Discussion and conclusion' contains a concluding discussion.

Background

Ion mobility spectrometers and their functions are well documented [12], and we do not go into technical details. Instead, we characterize the data generated by an MCC/IMS experiment (Section 'Data from MCC/IMS measurements'). In Section 'Peak models' we describe a previously used parametric peak model, and in Section 'Optimization methods' we review two optimization methods that are being used as subroutines in this work.

Data from MCC/IMS measurements

In an MCC/IMS experiment, a mixture of several unknown volatile organic compounds (VOCs) is separated in two dimensions: first by retention time r in the MCC (the time required for a particular compound to pass through the MCC) and second by drift time d through the IM spectrometer. Instead of the drift time itself, a quantity normalized for pressure and temperature called the *inverse reduced mobility* (IRM) t is used to compare spectra taken under different or changing conditions. Thus we obtain a time series of IM spectra (one spectrum each 100 ms at each retention time point), and each spectrum is a vector of ion concentrations (measured by voltage change on a Faraday plate) at each IRM.

Let R be the set of (equidistant) retention time points and let T be the set of (equidistant) IRMs where a measurement is made. If D is the corresponding set of drift times (each 1/250000 second for 50 ms, that is 12 500 time points), there exists a constant $C_{t|d} > 0$ depending on external conditions [12] such that $T = C_{t|d} \cdot D$. Then the data 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, and our goal is to discover them and to describe them by parametric models. Comparing peak coordinates with reference databases may reveal the identity of the corresponding compound. A peak caused by a VOC occurs over several IM spectra. We mention some properties of MCC/IMS data that complicate the analysis.

- An IM spectrometer uses an ionized carrier gas. These ions are present in every spectrum in addition to the analyte ions, and they create the *reactant ion peak* (RIP). In the whole IMSC it is present as high-intensity chromatogram at a specific IRM (Figure 1). When no analytes are injected into the device, the spectra contain only the RIP and are called *RIP-only spectra*.
- Every spectrum contains a tailing of the RIP, so the RIP is right-skewed (Figure 2). To extract peaks, the effect of the RIP and its tailing must be estimated and removed.
- At higher concentrations, compounds can form dimer ions, and one may observe both the monomer and dimer peak from one compound. This means that there is not necessarily a one-to-one correspondence between peaks and compounds, and our work focuses on peak detection, not compound identification.
- An IM spectrometer may operate in positive or negative mode, depending on which type of ions (positive or negative) one wants to detect. In either case, signals are reported in positive units. All experiments described here were done in positive mode.

Peak models

For our purpose of analyzing MCC/IMS measurements, a peak is characterized by the following assumptions.

Assumptions 1. *An n-dimensional peak P is a product of n log-concave functions with two inflection points in each dimension. The peak width at half height* $\omega_{1/2,i}$ *can be*

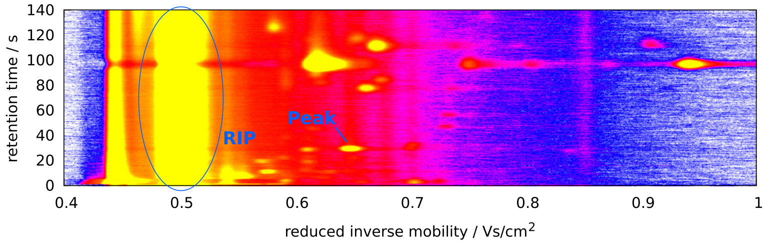

Figure 1 Visualization of a raw measurement (IMSC) as a heat map; signal color: white (lowest) < blue < purple < red < yellow (highest). The constantly present reactant ion peak (RIP) with mode at 0.48 Vs/cm^2 and exemplarily one VOC peak are annotated.

calculated with respect to the mode for each dimension i. At its mode $(m_1, \ldots, m_n)$*, peak P exceeds the average background noise level by a certain factor times the standard deviation of the noise.*

For MCC/IMS measurements, we have $n = 2$ dimensions, and in both retention time dimension and IRM dimension, we use the shifted Inverse Gaussian distribution g [13] as peak model function:

$$g(x;\mu,\lambda,o) := \frac{1[x>o]}{\sqrt{2\pi}} \cdot \\ \cdot \sqrt{\frac{\lambda}{(x-o)^3}} \cdot \exp\left(-\frac{\lambda\left((x-o)-\mu\right)^2}{2\mu^2(x-o)}\right). \qquad (1)$$

Its parameters are the shift (or offset) o, the relative mean $\mu > 0$ (to the right of o) and the shape parameter $\lambda > 0$. A peak is then given as the product of two shifted Inverse Gaussians, scaled by a volume factor v, i.e., by seven parameters; so the density function of a peak is $p(r,t) := v \cdot g(r,\mu_r,\lambda_r,o_r) \cdot g(t,\mu_t,\lambda_t,o_t)$ for all $r \in R, t \in T$.

Since the parameters μ, λ, o of a shifted Inverse Gaussian may be different even though the resulting distributions have a similar shape, it is more intuitive to describe the shifted Inverse Gaussian in terms of three different *descriptors*: the (absolute) mean $\mu' = o + \mu$, the standard deviation σ and the mode m. There is a bijection between (μ, λ, o) and (μ', σ, m) [13] summarized in Appendix A.

We also make use of the following empirically observable properties of peaks in real IMSCs that concern the peak widths on both the IRM axis and the retention time axis. The width can be described as the length $\omega_{1/2}$ of the interval around the mode where the peak height is at least half of its maximum height. For a (symmetric) Gaussian distribution, there is a linear relation between the standard deviation σ and $\omega_{1/2}$:

$$\omega_{1/2} = \phi \cdot \sigma \qquad \text{with } \phi = 2\sqrt{2 \ln 2} \approx 2.3548\,. \qquad (2)$$

This relation approximately holds as well for not too skewed Inverse Gaussian distributions and is a good approximation to estimate its descriptor σ approximately from an empirically observed $\omega_{1/2}$.

Given the mode d^* of a peak in drift time (in ms), we can estimate its descriptors (m, σ, μ') in IRM units as follows. Recall that the IRM mode (in V s cm^{-2}) is simply $m = C_{t|d} \cdot d^*$, where $C_{t|d}$ is the conversion constant between drift time and IRM (see Section 'Data from MCC/IMS measurements'). Spangler *et al.* [14] empirically derived that $\omega_{1/2} = \sqrt{(11.09\,\mathcal{D}\,d^*)/v_d^2 + d_{\text{grid}}^2}$, where $\mathcal{D}$ is the diffusion coefficient, v_d the drift velocity. Using the Einstein relation [15], $\mathcal{D}$ can be computed as $\mathcal{D} = k\mathcal{K}_B\mathcal{T}/q$, where k is the ion mobility, $\mathcal{K}_B$ the Boltzmann constant, $\mathcal{T}$ the absolute temperature and q the electric charge. We then use (2) to estimate $\sigma \approx \omega_{1/2}/\phi$. Finally, the mean is empirically found to be $\mu' \approx C_{t|d} \cdot \left(d^* + \sqrt{(4.246 \cdot 10^{-5})^2 + (d^*)^2/585048.1633}\right)$.

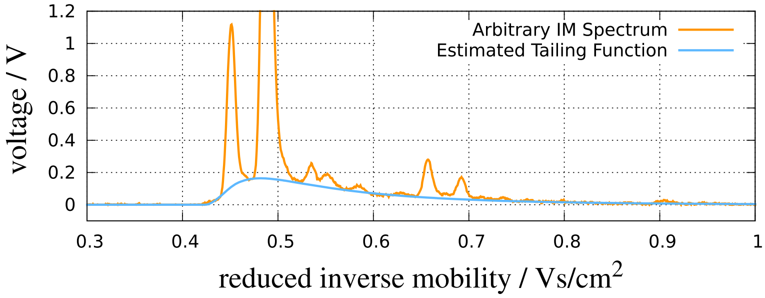

Figure 2 A spectrum and its estimated tailing function.

On the retention time axis, the peak width $\omega_{1/2}$ grows approximately linearly with retention time, i.e., there are constants `r_width_offset` > 0 and `r_width_factor` > 0 such that width of a peak with maximum at retention time r is approximately

$$\xi(r) := r \cdot \texttt{r\_width\_factor} + \texttt{r\_width\_offset}. \tag{3}$$

Optimization methods

The online peak extraction algorithm makes use of non-linear unconstrained minimization, similar to non-linear least squares, and of the EM algorithm. Both methods are summarized here.

Non-linear Least Squares

The NLLS method is an iterative method to estimate parameters $\theta = (\theta_1, \ldots, \theta_q)$ of a supposed parametric function f, given n observed data points $(x_1, y_1), \ldots, (x_n, y_n)$ with $y_i = f(x_i; \theta)$. The idea is to minimize the quadratic error $\sum_{i=1}^{n} r_i^2(\theta)$ between the function and the observed data, where $r_i(\theta) := y_i - f(x_i; \theta)$ is the *residual* of the i-the datapoint. The necessary optimality condition is $\sum_i r_i(\theta) \cdot \partial r_i(\theta)/\partial\theta_j = 0$ for all j. If f is linear in θ (e.g., a polynomial in x with θ being the polynomial coefficients, a setting called polynomial regression), then the optimality condition results in a linear system, which can be solved in closed form. However, often f is not linear in θ and we obtain a non-linear system, which is solved iteratively, given initial parameter values, by linearizing it in each iteration. Details and different algorithms for NLLS can be found in the literature ([16], Chapter 10). In this paper, we use a different, non-symmetric loss function, but apply similar techniques to solve the problem (see below).

The EM algorithm for mixtures with heterogeneous components

The observed data $x = (x_1, \ldots, x_n)$ is viewed as a *sample* from a *mixture* of probability distributions, where the mixture density is specified by $f(x_i \,|\, \omega, \theta) = \sum_{c=1}^{C} \omega_c\, f_c(x_i \,|\, \theta_c)$. Here c indexes the C different component distributions f_c, where θ_c denotes the parameters of f_c, and $\theta = (\theta_1, \ldots, \theta_C)$ is the collection of all parameters. The mixture coefficients satisfy $\omega_c \geq 0$ for all c, and $\sum_c \omega_c = 1$. Unlike in most applications, where all component distributions f_c are multivariate Gaussians, here the f_c are of different types (e.g., uniform and Inverse Gaussian). The goal is to determine the parameters ω and θ such that the probability of the observed sample is maximal (maximum likelihood paradigm). Since the resulting optimization problem is non-convex in (ω, θ), the EM algorithm is an iterative method that will converge to a local optimum [17] in parameter space. The EM algorithm consists of two repeated steps: The E-step (expectation) estimates the expected membership of each data point in each component and then the component weights ω, given the current model parameters θ. The M-step (maximization) estimates maximum likelihood parameters θ_c for each parametric component f_c individually, using the expected memberships as hidden variables that decouple the model.

E-Step. To estimate the expected membership $W_{i,c}$ of data point x_i in each component c, the component's relative probability at that data point is computed, such that $\sum_c W_{i,c} = 1$ for all i. Then the new component weight estimates ω_c^+ are the averages of $W_{i,c}$ across all n data points.

$$W_{i,c} = \frac{\omega_c f_c(x_i \,|\, \theta_c)}{\sum_k \omega_k f_k(x_i \,|\, \theta_k)}, \qquad \omega_c^+ = \frac{1}{n} \sum_{i=1}^{n} W_{i,c}, \tag{4}$$

Convergence. After each M-step of an EM cycle, we compare $\theta_{c,q}$ (old parameter value) and $\theta_{c,q}^+$ (updated parameter value), where q indexes the elements of θ_c, the parameters of component c. We say that the algorithm has converged when the relative change $\kappa_{c,q} := |\theta_{c,q}^+ - \theta_{c,q}| \,/\, \max\left(|\theta_{c,q}^+|, |\theta_{c,q}|\right)$ drops below a given threshold `thresh` for all c, q, (if $\theta_{c,q}^+ = \theta_{c,q} = 0$, we set $\kappa_{c,q} := 0$).

Having reviewed the necessary background, we now describe the methods we use for peak extraction from IMSCs.

Denoising and baseline correction

Background

A major challenge during peak detection in an IM spectrum is to find peaks that only slightly exceed the background noise level in a spectrum $S = (S_t)$. To determine whether the intensity S_t at coordinate t belongs to a peak region or can be solely explained by background noise, we propose a method based on the EM algorithm. It runs in $\mathcal{O}(\tau |T|)$ time where τ is the number of EM iterations.

Mixture model

Based on observations of IM spectra signal intensities, we assume that

- the noise intensity has a Gaussian distribution over low intensity values with mean μ_N and standard deviation σ_N,

$$p_\mathrm{N}(s \,|\, \mu_\mathrm{N}, \sigma_\mathrm{N}) = \frac{1}{\sqrt{2\pi}\,\sigma_\mathrm{N}} \cdot \exp\left(-(s - \mu_\mathrm{N})^2/(2\,\sigma_\mathrm{N}^2)\right)$$

- the true signal intensity has an Inverse Gaussian distribution with mean μ_S and shape parameter λ_S, i.e.,

$$p_S(s \mid \mu_S, \lambda_S) = \sqrt{\lambda_S/(2\pi s^3)} \cdot \exp\left(-\lambda_S(s-\mu_S)^2/(2\mu_S^2 s)\right)$$

- there is an unspecific background component which is not well captured by either of the two previous distributions; we model it by the uniform distribution over all intensities,

  $$p_B(s) = 1/(\max(S) - \min(S)),$$

  and we expect the weight ω_B of this component to be close to zero in standard IM spectra. High weights indicate an anomaly during the measurement.

We interpret the observed spectrum S as a sample of a mixture of these three components with unknown mixture coefficients. To illustrate this approach, consider Figure 3, which shows the empirical intensity distribution (histogram) of an arbitrary spectrum, together with the estimated components (except the uniform distribution, which has the expected coefficient of almost zero).

It follows that there are six independent parameters to estimate: μ_N, σ_N, μ_S, λ_S and weights $\omega_N, \omega_S, \omega_B$ (noise, signal, background, where $\omega_B = 1 - \omega_N - \omega_S$).

Initial parameter values

Background noise intensities are assumed to follow a Gaussian distribution at small intensity values. We can determine its approximate mean μ_N and standard deviation σ_N by considering the first and last 10% of data points in each spectrum.

The initial weight of the noise component is set to cover most points covered by this Gaussian distribution, i.e., $\omega_N := |\{t \in T \mid S_t \leqslant \mu_N + 3\,\sigma_N\}| \;/\; |T|$.

We assume that almost all of the remaining weight belongs to the signal component, thus $\omega_S = (1 - \omega_N) \cdot 0.999$, and $\omega_B = (1 - \omega_N) \cdot 0.001$.

To obtain initial parameters for the signal model, let $I' := \{t \in T \mid S_t > \mu_N + 3\,\sigma_N\}$ (the complement of the intensities that are initially assigned to the noise component). We set $\mu_S = \left(\sum_{t \in I'} (S_t - \mu_N)\right)/|I'|$ and $\lambda_S = \left(\sum_{t \in I'} (1/(S_t - \mu_N) - 1/\mu_S)\right)^{-1}$ (which are the maximum likelihood estimators for Inverse Gaussian parameters).

E-step

The hidden parameters $W_{t,c}$ are computed using (4), where the three component distributions f_c are the three component densities p_N, p_S, p_B with their parameters and the data x is a mean-smoothed version of the original spectrum S: $x_t := \frac{1}{2\alpha+1} \cdot \sum_{t'=t-\alpha}^{t+\alpha} S_{t'}$, where the smoothing window margin is $\alpha := (1/2) \cdot d_{\text{grid}} \cdot C_{\text{t|d}} \cdot |T|/T_{\text{last}}$. (Here d_{grid} is the grid opening time of the spectrometer and T_{last} is the maximum IRM in T).

Maximum likelihood estimators

In the maximization step (M-step) we estimate maximum likelihood parameters for the non-uniform components using the original intensities of S again.

$$\mu_N = \frac{\sum_t W_{t,N} \cdot S_t}{\sum_t W_{t,N}}, \tag{5}$$

$$\mu_S = \frac{\sum_t W_{t,S} \cdot (S_t - \mu_N)}{\sum_t W_{t,S}}, \tag{6}$$

$$\sigma_N^2 = \frac{\sum_t W_{t,N} \cdot (S_t - \mu_N)^2}{\sum_t W_{t,N}}, \tag{7}$$

$$\lambda_S = \frac{\sum_t W_{t,S}}{\sum_t W_{t,S} \cdot (1/(S_t - \mu_N) - 1/\mu_S)} \tag{8}$$

for all $t \in T$.

Final step

After convergence, we correct the baseline and remove noise: We first subtract μ_N from the signal value and then reduce the remaining value by the estimated noise weight. The corrected spectrum S^+ is

$$S_t^+ := \max\left\{(1 - W_{t,N})(S_t - \mu_N), 0\right\}, \quad t \in T.$$

Reducing a spectrum to peak models

Background

The idea of processing a single (noise-reduced) IM spectrum S is to deconvolute it into separate components described with statistical distribution functions. Several components appear in each spectrum besides the peaks, namely the previously described RIP and the tailing

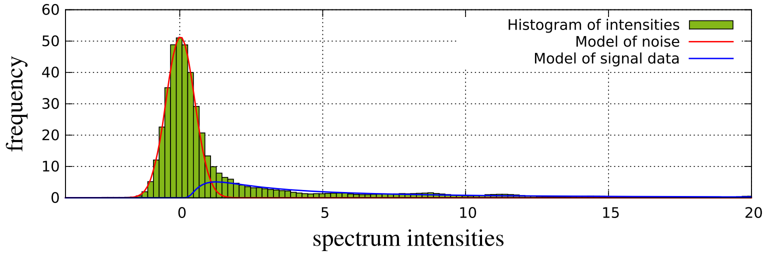

Figure 3 Histogram of an arbitrary IM spectrum (green bars) and estimated distribution of the noise component (red line) and of the signal component (blue line). Parameters for both components were estimated with the EM algorithm.

described in Section 'Data from MCC/IMS measurements'. and background noise. We first determine and remove the RIP tailing function and then determine the peak parameters (including the RIP).

Determining the tailing function

The tailing function appears as a baseline in every spectrum (see Figure 2 for an example). Its shape and scale changes from spectrum to spectrum; so it has to be determined in each spectrum and subtracted in order to extract peaks from the remaining signal in the next step. Empirically, we observe that the tailing function $f(t)$ can be described by a scaled shifted Inverse Gaussian, $f(t) = \nu \cdot g(t; \mu, \lambda, o)$ with g given by (1). The goal is to determine the parameters $\theta = (\nu, \mu, \lambda, o)$ such that $f_\theta(t)$ under-fits the given data $S = (S_t)$, as shown in Figure 2.

Let $r_\theta(t) := S(t) - f_\theta(t)$ be the residual function for a given choice θ of parameters. As we want to penalize $r(t) < 0$ but not (severely) $r(t) > 0$, we use the following non-symmetric loss function that depends on a threshold parameter $\gamma > 0$:

$$e_t(\theta; \gamma) := \begin{cases} r_\theta(t)^2/2 & \text{if } r_\theta(t) < \gamma\,, \\ \gamma \cdot r_\theta(t) - \gamma^2/2 & \text{if } r_\theta(t) \geq \gamma\,. \end{cases}$$

That is, the loss at time t is the residual squared when it has a negative or small positive value less than the given threshold $\gamma > 0$, but becomes a linear function of the residual for larger positive residuals.

The goal is to find θ to minimize the total loss $L(\theta) := \sum_t e_t(\theta; \gamma)$ for the given spectrum S and given $\gamma > 0$. We use gradient descent to solve this nonlinear optimization problem, to which we refer as non-linear loss minimization (NLLM).
To estimate the tailing function,

1. we determine reasonable initial values for the parameters $\theta = (\nu, \mu, \lambda, o)$ (see below),
2. we solve NLLM with $\gamma = \sigma_{\text{N}}^2$ to estimate the scaling factor ν, leaving the other parameters fixed,
3. we solve NLLM with $\gamma = \sigma_{\text{N}}^2$ to estimate all four parameters,
4. we solve NLLM with $\gamma = \sigma_{\text{N}}^2/100$ to re-estimate the scaling factor ν

where σ_{N} is the standard deviation of the noise as described in Section 'Denoising and baseline correction'.

The initial parameter values (ν, μ, λ, o) are determined as follows: For the scaling factor, we initially set $\nu = (1/2) \sum_{t \leqslant |T|} S_t$. For the other parameters, we first estimate the descriptors (μ', σ, m) as described below and then use the correspondence to the parameters listed in Appendix A. The initial σ is set to the standard deviation of the whole RIP-only spectrum. We determine the initial m as the RIP mode. It is a property of the Inverse Gaussian distributions under consideration such that the mean μ' can only range within the interval $I = [m, m + 0.7\sigma]$. To obtain an appropriate value for μ', an auxiliary offset variable o' is set to the largest IRM left of the RIP mode where the signal is below σ_{N}, and μ' it is increased in small steps within I. The candidate descriptors (μ', σ, m) are converted into corresponding parameters (μ, λ, o) until $o \geq o'$. The so obtained parameters constitute the initial parameter values.

Extracting peak parameters from a single spectrum

To extract all peaks from a spectrum (from left to right), we repeat three sub-steps:

1. scanning for a potential peak, starting where the previous iteration stopped;
2. determining peak parameters (Inverse Gaussian distribution);
3. subtracting the peak from the spectrum and continuing with the remainder.

Scanning. The algorithm scans for peaks, starting at the left end of S, by sliding a window of a given width across S and fitting a quadratic polynomial model to the data points within the window. The window width (in index units) is related to the grid opening time d_{grid} of the spectrometer and given as $d_{\text{grid}}/D_{\text{last}} \cdot |D|$ data points, where D_{last} is the maximum (last) drift time measured.

The model is built in drift time (not IRM). Let $f(d; \theta) = \theta_2\, d^2 + \theta_1\, d + \theta_0$ be the fitted quadratic polynomial inside the window. We call a window a *peak window* if the following conditions are fulfilled:

- the extreme drift time $d^* = \theta_1/(2\theta_2)$ lies within the drift times of the window;
- the extreme drift time d^* indicates a maximum (i.e., $\theta_2 < 0$);
- the maximum is sufficiently high above the noise level (which is zero after preprocessing): $f(d^*; \theta) \geq \sigma_{\text{N}}$

The first condition can be more strongly restricted to achieve more reliable results, by shrinking the interval towards the center of the window. If no peak is found, the moving window is shifted one index forward. If a peak is detected, the window is shifted half the window length forward before the next scan begins, but first the peak parameters are computed.

Determining peak parameters. As described in Section 'Peak models', we can estimate all peak descriptors (m, σ, μ') from its mode d^* in drift time. We convert them into the parameters (μ, λ, o) of the Inverse Gaussian parameterization (see Appendix A). The scaling factor ν for the peak is $\nu = f(d^*; \theta)/g(m; \mu, \lambda, o)$.

The model function is subtracted from the spectrum, and the next iteration is started with a window shifted by α index units (consider Section 'E-step'). For each spectrum, the output of this step is a *spectrum peak list*, which is a set of parameters for a mixture of weighted Inverse Gaussian models describing the peaks.

Aligning consecutive spectrum peak lists

Background

Having a set of peak parameters for each spectrum, the question arises how to merge the sets $P = (P_i)$ and $P^+ = (P_j^+)$ of two consecutive spectra. For each peak P_i, we have stored the Inverse Gaussian parameters μ_i, λ_i, o_i, the peak descriptors μ_i', σ_i, m_i (mean, standard deviation, mode) and the scaling factor ν_i, and similarly so for the peaks P_j^+. The idea is to compute a global alignment similar to the Needleman-Wunsch method [18] between P and P^+. We need to specify how to score aligning P_i to P_j^+ and how to score leaving a peak unaligned (i.e., a gap).

Scoring peak alignments

The score Z_{ij} for aligning P_i to P_j^+ is chosen by evaluating P_i's density function at the new mode m_j^+ and comparing it to "typical" value an approximate standard deviation away from the mode (at $m_i + \delta$, where $\delta := d_{\text{grid}} \cdot C_{\text{t|d}}/\phi$), resulting in the log-odds score

$$\zeta_{i,j} = \ln\left(\frac{g(m_j^+;\ \mu_i, \lambda_i, o_i)}{g(m_i + \delta;\ \mu_i, \lambda_i, o_i)}\right).$$

Alternatively, leaving a peak unmatched results in a gap score of zero.

Applying Needleman-Wunsch global alignment, we can compute the optimal score of aligning the first i peaks in the former spectrum with the first j peaks in the current spectrum by dynamic programming. We initialize a matrix Z, setting all $Z_{i,0}$ and $Z_{0,j}$ to zero and then compute, for $i \geq 1$ and $j \geq 1$,

$$Z_{i,j} = \max \begin{cases} Z_{i-1,j-1} + \zeta_{i,j}, \\ Z_{i-1,j}, \\ Z_{i,j-1}. \end{cases}$$

Obtaining peak chains

The alignment is obtained with a traceback, recording the optimal case in each cell, as usual. There are three cases to consider.

- If P_j^+ is not aligned with a peak in P, potentially a new peak starts at this retention time. Thus model P_j^+ is put into a new peak chain.
- If P_j^+ is aligned with a peak P_i, the chain containing P_i is extended with P_j^+.
- All peaks P_i that are not aligned to any peak in P^+ indicate the end of a peak chain at the current retention time.

All completed peak chains are forwarded to the next step, two-dimensional peak model estimation.

Estimating 2-D peak models

Background

Let $C = (P_1, \ldots, P_n)$ be a chain of one-dimensional Inverse Gaussian models. The goal of this step is to estimate a two-dimensional peak model (product of two one-dimensional Inverse Gaussians) from the chain, as described in Section 'Peak models', or to reject the chain if the chain does not fit such a model well. Potential problems are that a peak chain may contain noisy 1-D peaks truncated at their borders, consist only of noise or in fact consist of several consecutive 2-D peaks at the same drift time and successive retention times.

Estimating the parameters

As discussed in Section 'Peak models', the half-height width $\omega_{1/2}$ in retention time of a peak centered at retention time r can be described by an affine function $\xi(r)$, Eq. (3), and $\omega_{1/2}$ can be converted to the corresponding number of data points (window width).

We have the parameters $(\hat{\nu}_i, \hat{\mu}_{i,\text{t}}, \hat{\lambda}_{i,\text{t}}, \hat{o}_{i,\text{t}})$ for each individual peak $i = 1, \ldots, n$ in a peak chain, and the corresponding descriptors $(\hat{\mu}_{i,\text{t}}', \hat{\sigma}_{i,\text{t}}, \hat{m}_{i,\text{t}})$, as well as the associated retention time r_i and peak height $h_i = \hat{\nu}_i \cdot g(\hat{m}_{i,\text{t}}; \hat{\mu}_{i,\text{t}}, \hat{\lambda}_{i,\text{t}}, \hat{o}_{i,\text{t}})$.

We proceed similarly to Section 'Extracting peak parameters from a single spectrum' by fitting quadratic polynomials $b(r;\theta) = \theta_2 r^2 + \theta_1 r + \theta_0$ in sliding windows of the appropriate width $\xi(r_i)$ such that $h_i \approx b(r_i;\theta)$.

Having found a window that fits a peak, we estimate initial descriptors for an Inverse Gaussian model in retention time as follows:

$$\begin{aligned} \nu_\text{r}' &= -\theta_1^2/(4\theta_2) + \theta_0, \\ \sigma_\text{r} &= \sqrt{\nu_\text{r}'/(2|\theta_2|)}, \\ m_\text{r} &= -\theta_1/(2\theta_2), \\ \mu_\text{r}' &= m_\text{r} + \xi(m_\text{r}) \,/\, (4\,\phi). \end{aligned}$$

The descriptors are then converted into model parameters (see Appendix A).

After processing each window, we have obtained a list of size, say, k, of Inverse Gaussian distributions. We expect these distributions to be a mixture of k overlapping peaks in a single peak chain. To obtain an optimal deconvolution, we first normalize the volume factors ν_r' of the k components to obtain $\nu_{\text{r},j}$ such that $\sum_{j=1}^{k} \nu_{\text{r},r} = 1$ and then apply the EM algorithm. As a byproduct, we obtain an $(n \times k)$ matrix $M = (M_{i,j})$ that determines the membership probability for each of the n data points (r_i, h_i) to each of the k models.

To obtain the Inverse Gaussian distribution parameters in the IRM dimension for each of the k models, we first compute model descriptors using a membership-weighted average over the individual model descriptors: For $j \in \{1, \ldots, k\}$, let

$$\overline{M}_j := \sum_{i \leqslant n} M_{i,j},$$

$$\mu'_{j,\mathrm{t}} := \frac{1}{\overline{M}_j} \sum_{i \leqslant n} M_{i,j} \cdot \hat{\mu}'_{i,\mathrm{t}},$$

$$\sigma_{j,\mathrm{t}} := \frac{1}{\overline{M}_j} \sum_{i \leqslant n} M_{i,j} \cdot \hat{\sigma}_{i,\mathrm{t}},$$

$$m_{j,\mathrm{t}} := \frac{1}{\overline{M}_j} \sum_{i \leqslant n} M_{i,j} \cdot \hat{m}_{i,\mathrm{t}}.$$

We then convert these descriptors back into model parameters (Appendix A). The final peak volume is computed as $v_j^* = v'_{j,\mathrm{r}} \cdot \sum_{i \leqslant n} v_{i,\mathrm{t}}$.

For every model $j \in \{1, \ldots, k\}$, we check the following conditions:

- The width at half height in the retention time dimension has approximately the expected size (cf. Eqs. (2), (3)): $\xi(m_{j,\mathrm{r}})/2 \leqslant \sigma_{j,\mathrm{r}}/\phi < 2 \cdot \xi(m_{j,\mathrm{r}})$,
- The peak height at its maximum is sufficiently above the noise level: $v_j^* \cdot g(m_{j,\mathrm{t}}; \mu_{j,\mathrm{t}}, \lambda_{j,\mathrm{t}}, o_{j,\mathrm{t}}) \cdot g(m_{j,\mathrm{r}}; \mu_{j,\mathrm{r}}, \lambda_{j,\mathrm{r}}, o_{j,\mathrm{r}}) \geq \texttt{noise\_margin} \cdot \sigma_\mathrm{N}$, where $\texttt{noise\_margin} > 0$ is a tunable parameter,
- the Inverse Gaussian peak model g in retention time correlates well (in terms of the Pearson product-moment correlation coefficient ρ) with its quadratic approximation b in a window around the mode. More precisely, consider the window $W = [\, m_{j,\mathrm{r}} - \xi(m_{j,\mathrm{r}})/\phi,\ m_{j,\mathrm{r}} + \xi(m_{j,\mathrm{r}})/\phi]$, the model vector $G = g(x; \mu_{j,\mathrm{r}}, \lambda_{j,\mathrm{r}}, o_{j,\mathrm{r}})$ for $x \in W$ and the quadratic approximation vector $B = b_j(x; \theta)$ for $x \in W$, and test whether the Pearson correlation satisfies $\rho(G, B) \geq \rho_{\mathtt{min}}$.

If all conditions are satisfied, we have identified a 2-D peak model $(v_j^*, \mu_{j,\mathrm{t}}, \lambda_{j,\mathrm{t}}, o_{j,\mathrm{t}}, \mu_{j,\mathrm{r}}, \lambda_{j,\mathrm{r}}, o_{j,\mathrm{r}})$. Otherwise the model is discarded.

Peak clustering

Background

We now consider a series of IMS measurements, for each of which we have extracted peaks available in the form of parameter vectors or descriptors. The question arises how to decide which descriptors in different measurements represent the same peak (and hence potentially the same VOC).

Let X be the union of peak locations in all measurements, let $|X| =: n$, and let $X_{i,\mathrm{R}}$ be the retention time of peak i and $X_{i,\mathrm{T}}$ its IRM. We introduce a clustering approach using the EM algorithm with two-dimensional Gaussian mixtures that differs from the standard approach by its ability to dynamically adjust the number of clusters in the process.

Mixture model

We assume that the measured retention times and IRMs belonging to peaks from the same compound are independently normally distributed in both dimensions around the (unknown) true retention time and IRM. Let $\theta_j := (\mu_{j,\mathrm{R}}, \sigma_{j,\mathrm{R}}, \mu_{j,\mathrm{T}}, \sigma_{j,\mathrm{T}})$ be the parameters for component j, and let $f_j(x' given \theta_j$ be a two-dimensional Gaussian product distribution for a peak location $x = (x_\mathrm{R}, x_\mathrm{T})$ with these parameters.

The mixture distribution is $f(x) = \sum_{j=1}^{C} \omega_j f_j(x \mid \theta_j)$ with a yet undetermined number C of clusters. Note that there is no "background" model component.

Initial parameter values

In the beginning, we initialize the algorithm with as many clusters as peaks, i.e., we set $C := n$. This assignment makes a background model obsolete, because all peaks are assigned to at least one cluster. All clusters get as start parameters for $\mu_{j,\mathrm{R}}, \mu_{j,\mathrm{T}}$ the original retention time and IRM of peak location X_j, respectively, for $j = 1, \ldots, n$. We set $\sigma_{j,\mathrm{T}} := \texttt{t\_width} > 0$ and $\sigma_{j,\mathrm{R}} := \xi(X_{j,\mathrm{R}})/\phi$.

Dynamic adjustment of the number of clusters

After computing weights in the E-step, but before starting the M-step, we dynamically adjust the number of clusters by merging clusters whose centers are close to each other. Every pair $j < k$ of clusters is compared in a nested for-loop. When $|\mu_{j,\mathrm{T}} - \mu_{k,\mathrm{T}}| < \texttt{t\_width}$ and $|\mu_{j,\mathrm{R}} - \mu_{k,\mathrm{R}}| < \xi\big(\max\{\mu_{j,\mathrm{R}}, \mu_{k,\mathrm{R}}\}\big)$, then clusters j and k are merged by summing the EM weights: $\omega^+ := \omega_j + \omega_k$ and $W_{i,+} := W_{i,j} + W_{i,k}$ for all i. The summed weights are assigned to the location of the cluster with larger weight. (The recomputation of the parameters happens immediately after merging in the maximization step). The comparison order may matter in rare cases for deciding which peaks are merged first, but since new means and variances are computed, possible merges that were omitted in the current iteration will be performed in the next iteration.

This merging step is applied after in the second EM iteration, since the cluster means need at least one iteration to move towards each other.

Maximum likelihood estimators

The maximum likelihood estimators for mean and variance of a two-dimensional Gaussian are the standard ones, taking into account the membership weights,

$$\mu_{j,d} = \frac{\sum_{i=1}^{n} W_{i,j} \cdot X_{i,d}}{\sum_{i=1}^{n} W_{i,j}}, \qquad d \in \{\mathrm{T,R}\}, \tag{9}$$

$$\sigma_{j,d}^2 = \frac{\sum_{i=1}^{n} W_{i,j} \cdot (X_{i,d} - \mu_{j,d})^2}{\sum_{i=1}^{n} W_{i,j}}, \qquad d \in \{\mathrm{T,R}\}, \tag{10}$$

for all components $j = 1, \ldots, C$.

One problem using this approach emerges from the fact that initially each cluster contains only one peak, leading to an estimated variance of zero in many cases. To prevent this, minimum values are enforced such that $\sigma_{j,\mathrm{T}} \geq$ `t_width` and $\sigma_{j,\mathrm{R}} \geq \xi(\mu_{j,\mathrm{R}})/\phi$ for all j.

Final step

The EM loop terminates when no merging occurs and the convergence criteria for all parameters are fulfilled. The resulting membership weights determine the number of clusters as well as the membership coefficient of peak location X_i to cluster j. If a hard clustering is desired, the merging step has to be traced.

Evaluation

We evaluate different properties of the online method:

1. the quality of reducing a single spectrum to a peak list (denoising/baseline correction (Section 'Denoising and baseline correction') and spectrum reduction (Section 'Reducing a spectrum to peak models'),
2. the execution time of both steps,
3. the quality of the new clustering approach,
4. the correlation between manual annotations on full IMSCs by a computer-assisted expert and our automated online extraction method.

Parameters. For evaluation measurements, the MCC was adjusted to a temperature of 40°C and throughput of 150 mL min^{-1}. The IMS had a voltage of 4380 V, a grid opening time of 300 μs and a throughput of 150 mL min^{-1}. We chose the following parameters [9]:

- `r_width_offset` = 2.5 s (width offset for peaks in retention time),
- `r_width_factor` = 0.06 (width slope for peaks in retention time),
- `t_width` = 0.003 V s cm^{-2} (standard deviation for peaks in IRM),
- `thresh` (convergence threshold; value varies within evaluation),
- `noise_margin` = 4 (factor multiplied with standard deviation of background noise for minimal peak height),
- $\rho_{\mathtt{min}}$ = 0.95 (minimal Pearson product-moment correlation coefficient).

Quality of single spectrum reduction

In a first experiment, we tested the quality of the spectrum reduction method using an idea by Munteanu and Wornowizki [19] that determines the agreement between an observed set of data points, interpreted as an empirical distribution function F (the data) and a model distribution G (the mixture distribution obtained from the peak list parameters). The approach writes $F = \tilde{s} \cdot G + (1 - \tilde{s}) \cdot H$ with $\tilde{s} \in [0, 1]$, where H is a non-parametric distribution whose inclusion ensures the fit of the model G to the data F. If the weight $\tilde{s}$ is close to 1.0, then F is a plausible sample from G. We compare the original spectra and reduced spectra (peaks from peak lists) from a previously used dataset [20]. This set contains 69 measurements preprocessed with a 5 × 5 average. Every measurement contains 1200 spectra. For each spectrum in all measurements, we computed the reduced spectrum model and determined $\tilde{s}$. Over 92% of all 82 000 models achieved $\tilde{s} =$ 1 and over 99% reached $\tilde{s} \geq 0.9$. No $\tilde{s}$ dropped below 85%. In summary, spectrum reduction provides an accurate parametric representation of most spectra.

Execution time

We tested our method on two different platforms, (1) a desktop PC with Intel(R) Core(TM) i5 2.80GHz CPU, 8GB memory, Ubuntu 12.04 (64bit) OS and (2) a Raspberry Pi [21] type B with ARM1176JZF-S 700MHz CPU, 512 MB memory, Raspbian Wheezy (32bit) OS, once with the factory defaults of 700 MHz and once overclocked up to 900 MHz. The Raspberry Pi was chosen because it is a complete credit-card-sized low-cost single-board computer with low CPU and power consumption (3.5 w). This kind of device is appropriate for data analysis in future mobile measurement devices.

Recall that each spectrum contains 12 500 data points. It is current practice to analyze not the full spectra, but aggregated ones, where five consecutive values are averaged. Here we consider the full spectra, slightly aggregated ones (average over two values, 6 250 data points) and standard aggregated ones (average over five values, 2 500 data points). We measured the average execution time of denoising, baseline correction and consecutive spectrum reduction. Table 1 shows the results. It is remarkable that at the highest resolution (Average 1) the Raspberry Pi with 900 MHz keeps barely the time bound of 100 ms between consecutive spectra. At lower resolutions, the Raspberry Pi satisfies the time restrictions easily. The desktop PC copes with the analysis effortless on any setting.

We found that in the steps that use the EM algorithm, on average 25–30 EM iterations were necessary for a precision of `thresh` := 0.001 (i.e., 0.1%) (see Convergence in Section 'The EM Algorithm for mixtures with heterogeneous components'). Relaxing the threshold from 0.001 to 0.01 halved the number of iterations without noticeable difference in the resulting estimated parameters.

Table 1 Average processing time of denoising, baseline correction and spectrum reduction on two platforms with different clock rates, averaging methods (single spectra, averages of 2 and 5 spectra) and convergence thresholds `thresh`

`thresh`	*Platform*	*Avg 1*	*Avg 2*	*Avg 5*
0.1%	Desktop PC	4.36 ms	2.09 ms	0.88 ms
	Rasp. Pi (700 MHz)	119.48 ms	55.02 ms	21.82 ms
	Rasp. Pi (900 MHz)	97.19 ms	43.62 ms	17.42 ms
1.0%	Desktop PC	4.26 ms	2.01 ms	0.66 ms
	Rasp. Pi (700 MHz)	116.69 ms	52.63 ms	16.99 ms
	Rasp. Pi (900 MHz)	94.03 ms	41.46 ms	13.48 ms

Clustering

To evaluate peak clustering methods, we simulate peak locations according to locations in real MCC/IMS datasets, together with the true partition $\mathcal{P}$ of peaks.

Most of the detected peaks appear in a small dense area early in the measurement. The remaining peaks are distributed widely, which is referred to as the sparse area (we let the areas overlap such that the dense are is contained in the sparse area). The areas approximately have the following boundaries (in units of (V s cm^{-2}, s) from lower left to upper right point, cf. Figure 1:

measurement: $(0, 0), (1.45, 600)$
dense area: $(0.5, 4), (0.7, 60)$
sparse area: $(0.5, 4), (1.2, 450)$

Peak clusters are ellipsoidal and dense. From [9], we know the minimum required distance between two peaks in order to be identified as two separate compounds. We simulate 30 peak cluster centroids in the dense area and 20 in the sparse area, all picked randomly and uniformly distributed in the respective area. We then randomly pick the number of peaks per cluster. We also randomly pick the distribution of peaks within a cluster. Since we do not know the actual distribution model, we decided to simulate with three models: normal (n), exponential (e) and uniform (u) distribution with the following densities:

$$\begin{aligned}
&f_{\mathrm{n}}(r, t \mid \mu_{\mathrm{t}}, \sigma_{\mathrm{t}}, \mu_{\mathrm{r}}, \sigma_{\mathrm{r}}) \\
&\quad = \mathcal{N}(t \mid \mu_{\mathrm{t}}, \sigma_{\mathrm{t}}) \cdot \mathcal{N}(r \mid \mu_{\mathrm{r}}, \sigma_{\mathrm{r}}) \\
&f_{\mathrm{e}}(r, t \mid \mu_{\mathrm{t}}, \lambda_{\mathrm{t}}, \mu_{\mathrm{r}}, \lambda_{\mathrm{r}}) \\
&\quad = \lambda_{\mathrm{t}} \lambda_{\mathrm{r}} \exp\big(-(\lambda_{\mathrm{t}}|t-\mu_{\mathrm{t}}| + \lambda_r |r-\mu_{\mathrm{r}}|)\big)/4 \\
&f_{\mathrm{u}}(r, t \mid \mu_{\mathrm{t}}, \nu_{\mathrm{t}}, \mu_{\mathrm{r}}, \nu_{\mathrm{r}}) \\
&\quad = \begin{cases} (\pi \nu_{\mathrm{t}} \nu_{\mathrm{r}})^{-1} & \text{if} \frac{|t-\mu_{\mathrm{t}}|^2}{\nu_{\mathrm{t}}^2} + \frac{|r-\mu_{\mathrm{r}}|^2}{\nu_{\mathrm{r}}^2} \leqslant 1 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

Here $(\mu_{\mathrm{t}}, \mu_{\mathrm{r}})$ is the coordinate of the centroid with RIM in Vs/cm^2 and retention time in s. For the normal distribution, we used $\sigma_{\mathrm{t}} = 0.002$ and $\sigma_{\mathrm{r}} = \mu_{\mathrm{r}} \cdot 0.002 + 0.2$. For the exponential distribution, we used $\lambda_{\mathrm{t}} = (1.45 \cdot 2500)^{-1}$ (reduced mobility width for in single cell within M) and $\lambda_{\mathrm{r}} = 1/(\mu_{\mathrm{r}} \cdot 0.002 + 0.2)$. For the uniform distribution, we used an ellipsoid with radii $\nu_{\mathrm{t}} = 0.006$ and $\nu_{\mathrm{r}} = \mu_{\mathrm{r}} \cdot 0.02 + 1$.

We compared our adaptive EM clustering with two common clustering methods: k-means and DBSCAN. Since k-means needs a fixed number of clusters k and appropriate start values for the centroids, used k-means++ [22] for estimating good starting values and give it an advantage by assigning the true number of partitions. DBSCAN has the advantages that it does not need to know the number of clusters in advance and that it can find non-linear cluster boundaries, but it does not easily yield parametric cluster descriptors.

To measure the quality of an obtained clustering $\mathcal{C}$ we use the Fowlkes-Mallows index (FMI, [23]) and the normalized variation of information (NVI) score [24].

For the FMI one considers all pairs of data points. If two data points belong to the same true partition of $\mathcal{P}$, they are called *connected*. Accordingly, a pair of data points is called *clustered* if they are clustered together by the clustering method are evaluating. Pairs of data points that are both connected and clustered are called true positives (TP). False positives (FP, not connected but clustered) and false negatives (FN, connected but not clustered) are computed analogously. The FMI is the geometric mean of precision and recall: $\mathrm{FMI}(\mathcal{P}, \mathcal{C}) := \sqrt{TP/(TP+FP) \cdot TP/(TP+FN)}$, where $\mathcal{P}$ is the true partition and $\mathcal{C}$ is the clustering. We have $\mathrm{FMI}(\mathcal{P}, \mathcal{C}) \in [0, 1]$, and $\mathrm{FMI}(\mathcal{P}, \mathcal{C}) = 1$ indicates perfect agreement. The FMI is difficult to interpret when the number of clusters in $\mathcal{C}$ and $\mathcal{P}$ differs significantly.

Therefore we use a second measure that considers cluster sizes only, the normalized variation of information (NVI). To compute the NVI, an auxiliary $(|\mathcal{P}| \times |\mathcal{C}|)$-dimensional matrix $A = (a_{i,j})$ is computed, where $a_{i,j}$ is the number of data points within partition i that are assigned to cluster j. The NVI score is defined via entropies; let n be the number of data points and

$$\begin{aligned}
H(\mathcal{P}) &:= -\sum_{i \leqslant |\mathcal{P}|} \frac{\sum_{j \leqslant |\mathcal{C}|} a_{i,j}}{n} \log \frac{\sum_{j \leqslant |\mathcal{C}|} a_{i,j}}{n}, \\
H(\mathcal{C}) &:= -\sum_{j \leqslant |\mathcal{C}|} \frac{\sum_{i \leqslant |\mathcal{P}|} a_{i,j}}{n} \log \frac{\sum_{i \leqslant |\mathcal{P}|} a_{i,j}}{n}, \\
H(\mathcal{P}|\mathcal{C}) &:= -\sum_{j \leqslant |\mathcal{C}|} \sum_{i \leqslant |\mathcal{P}|} \frac{a_{i,j}}{n} \log \frac{a_{i,j}}{\sum_{i' \leqslant |\mathcal{P}|} a_{i',j}}, \\
H(\mathcal{C}|\mathcal{P}) &:= -\sum_{j \leqslant |\mathcal{C}|} \sum_{i \leqslant |\mathcal{P}|} \frac{a_{i,j}}{n} \log \frac{a_{i,j}}{\sum_{j' \leqslant |\mathcal{C}|} a_{i,j'}}, \\
NVI(\mathcal{P}, \mathcal{C}) &:= \begin{cases} \frac{H(\mathcal{P}|\mathcal{C}) + H(\mathcal{C}|\mathcal{P})}{H(\mathcal{P})} & \text{if } H(\mathcal{P}) \neq 0, \\ H(\mathcal{C}) & \text{otherwise.} \end{cases}
\end{aligned}$$

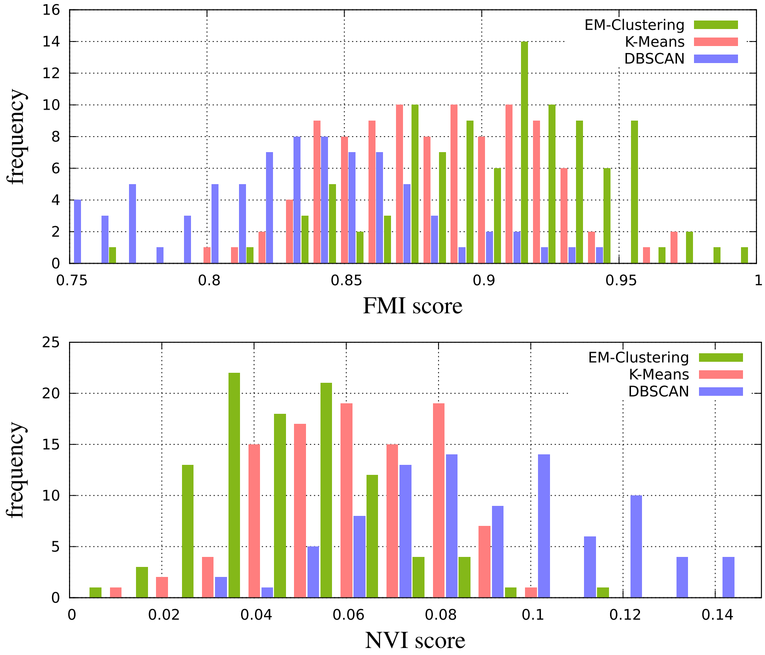

Figure 4 Histograms of Fowlkes-Mallows index (FMI; higher is better) and normalized variation of information (NVI; lower is better) comparing 100 simulated measurements containing partitioned peak locations with their clusters produced by the different methods.

Here $NVI(\mathcal{P}, \mathcal{C}) = 0$ indicates perfect agreement between the cluster size distributions. Together, an FMI of 1 and an NVI score of 0 indicate a perfect clustering.

For the first test, we evaluated 100 sets of data points distributed as described above. The cluster model (normal, exponential or uniform) was drawn randomly. The results show that even with the advantage that k-means knows the true k, our adaptive EM clustering performs best on average in terms of FMI and NVI score (Figure 4).

For the second test, we additionally inserted 200 uniformly distributed (noise) peaks into the measurement area. All these peaks are singletons and have no matching

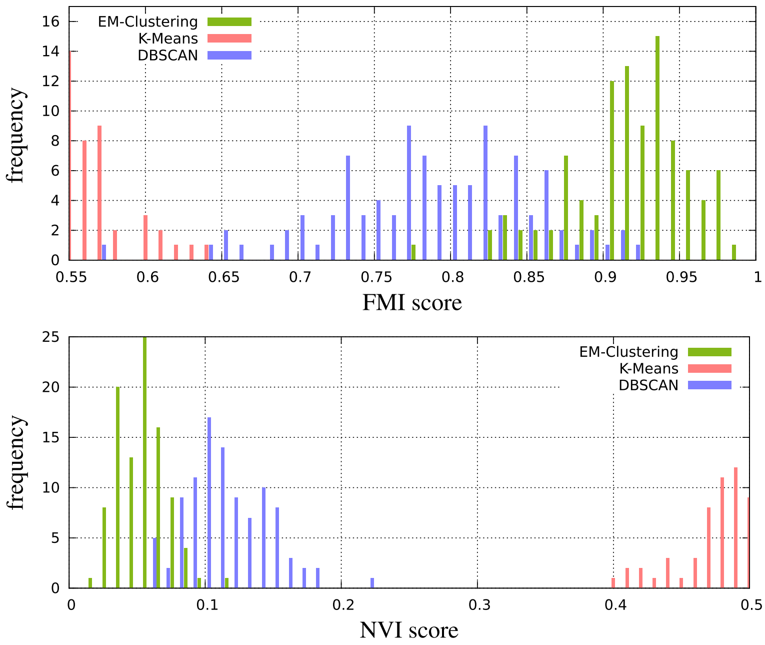

Figure 5 Histograms similar to Figure 4, but in a more realistic noisy scenario (see text). An FMI of 1 and NVI of 0 would be optimal.

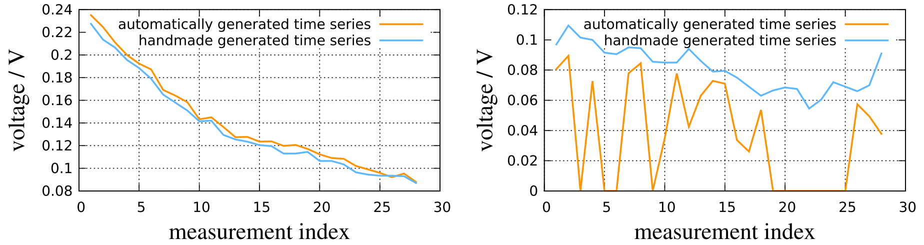

Figure 6 Time series of discovered intensities of two peaks. Left: A peak with agreement between manual and automated online annotation. Right: A peak where the online method fails to extract the peak in several measurements. If one treated zeros as missing data, the overall trend would still be visible.

peaks. The results (Figure 5) show that the adaptive EM clustering still performs best on average, whereas k-means fails.

Comparison of automated online peak extraction with manual offline annotation

The fourth experiment compares extracted peaks from a time series of measurements of two automated methods to an expert manual annotation. The automated methods are our online analysis process described here and automated peak detection using the commercial VisualNow software.

Here 15 rats were monitored in 20 minute intervals for up to a day. Each rat resulted in 30–40 measurements (a time series) for a total of 515 measurements. To track peaks over time, we used the previously described EM clustering method.

As an example, Figure 6 shows time series of intensities of two peaks detected by computer-assisted manual annotation and using our online algorithm. The example shows that there are cases where the sensitivity of the online algorithm is not perfect; this is mainly true for peaks whose intensity only slightly exceeds the background noise.

To obtain an overview over all time series, we computed the cosine similarity $\gamma \in [-1, +1]$ time series of peak intensities discovered by manual annotation and each automated method. We also computed the recall automated method for each time series, that is, the relative fraction of measurements where the peak was found by the algorithm among those where it was found by manual annotation. Figure 7 shows overall good agreement between both automated methods (our online method and automated VisualNow peak extraction) with

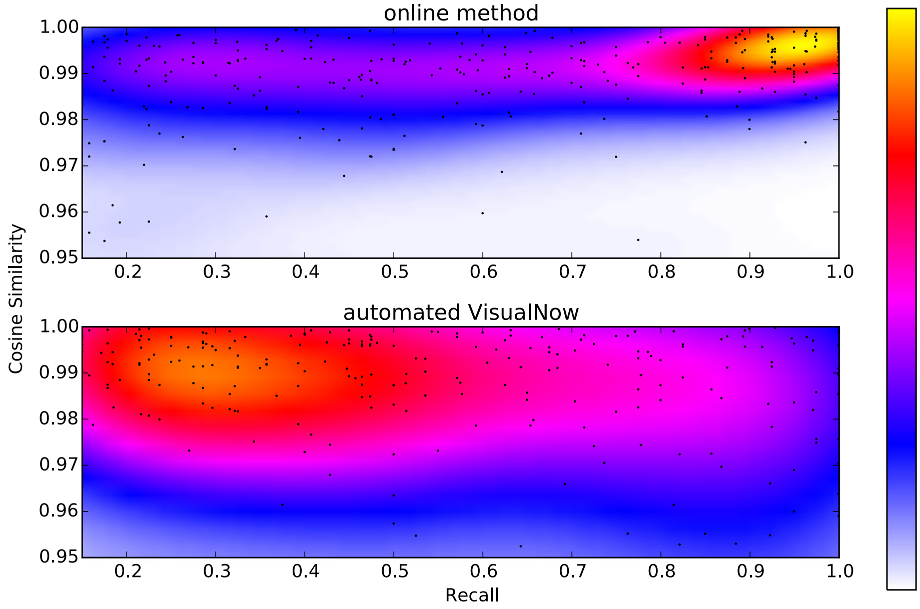

Figure 7 Kernel density estimation (kde) plots of recall and cosine similarity of peak intensity, comparing automatically picked peaks from our online algorithm and VisualNow against expert annotation. Each dot corresponds to a time series of one peak. Optimal results would be a recall of 1.0 and a cosine similarity of 1.0 for each time series.

the expert manual annotation. The cosine similarity of the inferred time series is in better agreement than the more variable recall. When comparing automated methods against each other, we outperform VisualNow in terms of sensitivity and computation time: About 31% of the points extracted by the online method exceed 90% recall and 98% cosine similarity whereas only 5% of the time series extracted by VisualNow achieve these values. The peak detection of one measurement takes about 2 seconds on average (when the whole measurement is available at once) with the online method and about 20 seconds with VisualNow on the desktop computer described above. VisualNow only provides the position and signal intensity of the peak's maximum, whereas our method additionally provides shape parameters.

Problems of our online method stem from low-intensity peaks only slightly above the detection threshold, and resulting fragmentary or rejected peak chains.

Discussion and conclusion

We presented the first approach to extract peaks from MCC/IMS measurements while they are being captured, with the long-term goal to remove the need for storing full measurements before analyzing them in small embedded devices. Our method is fast and satisfies the time restrictions even on a low-power CPU platform like a Raspberry Pi and outperforms existing software.

While performing well on single spectra, there is room for improvement in merging one-dimensional peak models into two-dimensional peak models. Our method has to be further evaluated in clinical studies or biotechnological monitoring settings. It also has not been tested with the negative mode of an IMS for lack of data. In general, the robustness of the method under adversarial conditions (high concentrations with formation of dimer ions, changes in temperature or carrier gas flow in the MCC) has to be evaluated and probably improved.

Appendix A: peak descriptors and parameters

The shifted Inverse Gaussian distribution with parameters o (shift or offset), μ (mean minus shift, also called relative mean) and λ (shape) is given by (1). There is a bijection [13] between (μ, λ, o) and the descriptors (μ', σ, m), which are the absolute mean $\mu' = o + \mu$, the standard deviation σ and the mode m. Given (μ, λ, o), we have

$$\begin{aligned}
\mu' &= \mu + o,\\
\sigma &= \sqrt{\mu^3/\lambda},\\
m &= \mu \cdot \left(\sqrt{1 + (9\mu^2)/(4\lambda^2)} - (3\mu)/(2\lambda)\right) + o,
\end{aligned}$$

and, given (μ', σ, m), we use auxiliary expressions p and q to find

$$\begin{aligned}
p &:= \big(-m(2\mu' + m) + 3 \cdot (\mu'^2 - \sigma^2)\big)/\big(2(m - \mu')\big),\\
q &:= \big(m(3\sigma^2 + \mu' \cdot m) - \mu'^3\big)/\big(2(m - \mu')\big),\\
o &= -p/2 - \sqrt{p^2/4 - q},\\
\mu &= \mu' - o,\\
\lambda &= \mu^3/\sigma^2.
\end{aligned}$$

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

DK developed and implemented the methods and evalutated them in consultation with SR. DK and SR wrote the manuscript. Both authors read and approved the final manuscript.

Acknowledgements

DK and SR acknowledge the support of Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 (http://sfb876.tu-dortmund.de), "Providing Information by Resource-Constrained Data Analysis", subproject TB1. We are grateful to Jörg Ingo Baumbach for help with numerous questions and for providing datasets. We thank Max Wornowizki from subproject C3 for providing methodology for the first evaluation.

Received: 20 March 2015 Accepted: 2 April 2015
Published online: 13 May 2015

References

1. Bessa V, Darwiche K, Teschler H, Sommerwerck U, Rabis T, Baumbach JI, et al. Detection of volatile organic compounds (VOCs) in exhaled breath of patients with chronic obstructive pulmonary disease (COPD) by ion mobility spectrometry. Int J Ion Mobility Spectrom. 2011;14:7–13.
2. Bunkowski A, Bödeker B, Bader S, Westhoff M, Litterst P, Baumbach JI. MCC/IMS signals in human breath related to sarcoidosis – results of a feasibility study using an automated peak finding procedure. J Breath Res. 2009;3(4):046001.
3. Westhoff M, Litterst P, Freitag L, Urfer W, Bader S, Baumbach JI. Ion mobility spectrometry for the detection of volatile organic compounds in exhaled breath of lung cancer patients. Thorax. 2009;64:744–8.
4. Keller T, Schneider A, Tutsch-Bauer E, Jaspers J, Aderjan R, Skopp G. Ion mobility spectrometry for the detection of drugs in cases of forensic and criminalistic relevance. Int J Ion Mobility Spectrom. 1999;2(1):22–34.
5. Ewing RG, Atkinson DA, Eiceman GJ, Ewing GJ. A critical review of ion mobility spectrometry for the detection of explosives and explosive related compounds. Talanta. 2001;54(3):515–29.
6. Kolehmainen M, Rönkkö P, Raatikainen O. Monitoring of yeast fermentation by ion mobility spectrometry measurement and data visualisation with self-organizing maps. Anal Chim Acta. 2003;484(1):93–100.
7. Kreuder AE, Buchinger H, Kreuer S, Volk T, Maddula S, Baumbach J. Characterization of propofol in human breath of patients undergoing anesthesia. Int J Ion Mobility Spectrom. 2011;14:167–75.
8. Bunkowski A. MCC-IMS data analysis using automated spectra processing and explorative visualisation methods. PhD thesis: Bielefeld University; 2011.
9. Bödeker B, Vautz W, Baumbach JI. Peak finding and referencing in MCC/IMS-data. Int J Ion Mobility Spectrom. 2008;11(1):83–7.
10. D'Addario M, Kopczynski D, Baumbach JI, Rahmann S. A modular computational framework for automated peak extraction from ion mobility spectra. BMC Bioinformatics. 2014;15(1):25.
11. Kopczynski D, Rahmann S. An online peak extraction algorithm for ion mobility spectrometry data. In: WABI. Lecture Notes in Computer Science. New York: Springer; 2014. p. 232–46.
12. Eiceman GA, Karpas Z. Ion Mobility Spectrom, Second Edition. New York: Taylor & Francis; 2005.
13. Kopczynski D, Baumbach JI, Rahmann S. Peak modeling for ion mobility spectrometry measurements. In: Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European. New York, NY, USA: IEEE; 2012. p. 1801–5.

14. Spangler GE, Collins CI. Peak shape analysis and plate theory for plasma chromatography. Anal Chem. 1975;47(3):403–7.
15. Einstein A. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. Annalen der Physik. 1905;322(8):549–60.
16. Nocedal J, Wright SJ. Numerical Optimization, 2nd edn. New York: Springer; 2006.
17. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodological). 1977;39:1–38.
18. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol. 1970;48(3):443–53.
19. Munteanu A, Wornowizki M. Demixing empirical distribution functions. Technical Report 2014-02, Collaborative Research Center 876, TU Dortmund. 2014.
20. Hauschild AC, Kopczynski D, D'Addario M, Baumbach JI, Rahmann S, Baumbach J. Peak detection method evaluation for ion mobility spectrometry by using machine learning approaches. Metabolites. 2013;3(2):277–93.
21. Raspberry Pi Foundation. Raspberry Pi. 2014. http://www.raspberrypi.org/.
22. Arthur D, Vassilvitskii S. K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '07. Philadelphia: Society for Industrial and Applied Mathematics; 2007. p. 1027–35.
23. Fowlkes EB, Mallows CL. A method for comparing two hierarchical clusterings. J Am Stat Assoc. 1983;78(383):553–69.
24. Reichart R, Rappoport A. The nvi clustering evaluation measure. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning. Stroudsburg, PA, USA: Association for Computational Linguistics; 2009. p. 165–73. http://dl.acm.org/citation.cfm?id=1596374.1596401.