AutoGCA - a software tool for the automated analysis of gas chromatograms

Henning Schröder[a], Tomass Andersons[a], Alexander Brächer[c], Roland Peschke[c], Martina Beese[a,b], Christoph Kubis[b], Mathias Sawall[a], Robert Franke[c,d], Klaus Neymeyr[a,b]

[a] *Universität Rostock, Institut für Mathematik, Ulmenstrasse 69, 18057 Rostock, Germany*
[b] *Leibniz-Institut für Katalyse, Albert-Einstein-Strasse 29a, 18059 Rostock, Germany*
[c] *Evonik Oxeno GmbH & Co. KG, Paul-Baumann Straße 1, 45772 Marl, Germany*
[d] *Lehrstuhl für Theoretische Chemie, Ruhr-Universität Bochum, 44780 Bochum, Germany*

Abstract

Gas chromatography (GC) is an important tool in analytical chemistry and large amounts of data are routinely produced. Despite the advances in the available software, the analysis of a dataset can still be time-consuming task, in part due to the manual or semi-automated marking of peaks. An automation of this tedious task is proposed in this work.

Keywords: Baseline, gas chromatography, peak detection, peak fitting

1. Introduction

With the widespread availability and use of different chromatographic techniques comes the need to interpret large amounts of experimental data. To extract valuable information from a chromatogram, the area and position of the contributing peaks must be determined. We call this the Peak Extraction Task (PET). It can be solved easily if the peaks are clearly separated. Therefore, we focus on chromatograms with a high number of at least partially overlapping peaks that might be affected by baseline distortion. Due to their difficulty, these cases are typically analyzed manually or semi-automatically, e.g., with peak suggestions that must be fine-tuned with a high expenditure of time. Although the full automation of PET has been extensively researched [1, 2, 3, 4, 5, 6], it often has poor applicability for highly overlapping or baseline distorted peaks, or there is a large number of parameters to be manually tuned.

This paper presents the concepts of AutoGCA (Automatic Gas Chromatogram Analysis), a fully automated algorithm for the PET. The novel aspects are the use of a fully asymmetric peak model, an automated noise level estimation and baseline approximation. Despite the name, the concepts are applicable or at least transferable to general chromatographic applications. All source code is available online on GitHub `https://github.com/henning1419/gcd` and is written in MATLAB. The significance and application of this work arose in cooperation with Evonik Oxeno GmbH & Co. KG.

The motivation for automating tedious chromatogram evaluations is to save time of researchers and laboratory staff, see [2] and more recently [1] as examples. Standard open-source methods are also available and can be combined to reduce manual work [7]. The use of clustering for parallelization in our work has been inspired by the work on peak detection in multi-overlapping chromatographic signals by Vivó-Truyols et al. [4]. Some review papers also consider a multitude of methods that are relevant for solving the PET. One such work is [8], which lists approaches for background correction, peak detection, and peak properties, among others. Similar challenges are also encountered when comparing chromatograms between different runs, which require peak tracking algorithms, see [9] and an updated approach in [10], as well as in 3-way datasets [11].

Convolutional neural networks are a promising alternative approach for peak detection, but they also require a complex training process and labeled training data [12, 13]. A full workflow for LC-MS is given by Peakonly [14] and PeakBot [15].

Another recent and comprehensive method is MOCCA [16] that applies deconvolution based on the L-BFGS-B optimizer. A further promising approach uses continuous wavelet transform and is capable of peak detection for

complex chromatograms [17]. The challenging nature of the datasets in [17] is comparable to requirements on the AutoGCA.

1.1. List of symbols

g	chromatographic signal,
t	retention time grid,
$s, \tilde{s}$	original signal segment and smoothed signal segment,
k	index of the center of the largest peak in a segment,
i_r, i_l	indices (right and left) for the peak width calculation,
b	baseline,
I	interval between two adjacent supporting points of the baseline,
g_B	baseline-corrected chromatogram,
w	windows for the noise level approximation,
L	approximated noise level,
$g(x;\mu,\sigma)$	Gaussian with the center μ and standard deviation σ,
$l(x;\mu,\sigma)$	a nearly linear curve model,
$p(x;\mu,h,\sigma_L,\sigma_R,\alpha_L,\alpha_R)$	peak model with center μ, height h, standard deviation σ_L (left) and σ_R (right) and ratio of the nearly linear curve α_L (left) and α_R (right),
$r^{(i)}$	the residual of a cluster segment after i fitted peaks.

Moreover, all Greek letters with the exception of μ and σ denote parameters of AutoGCA and are listed in Sec. 3.2.

2. The overview of the algorithm

For a given chromatogram, AutoGCA extracts the positions, areas, and other complementary features of the underlying peaks. The general concept is outlined in Fig. 1 and explained in this section.

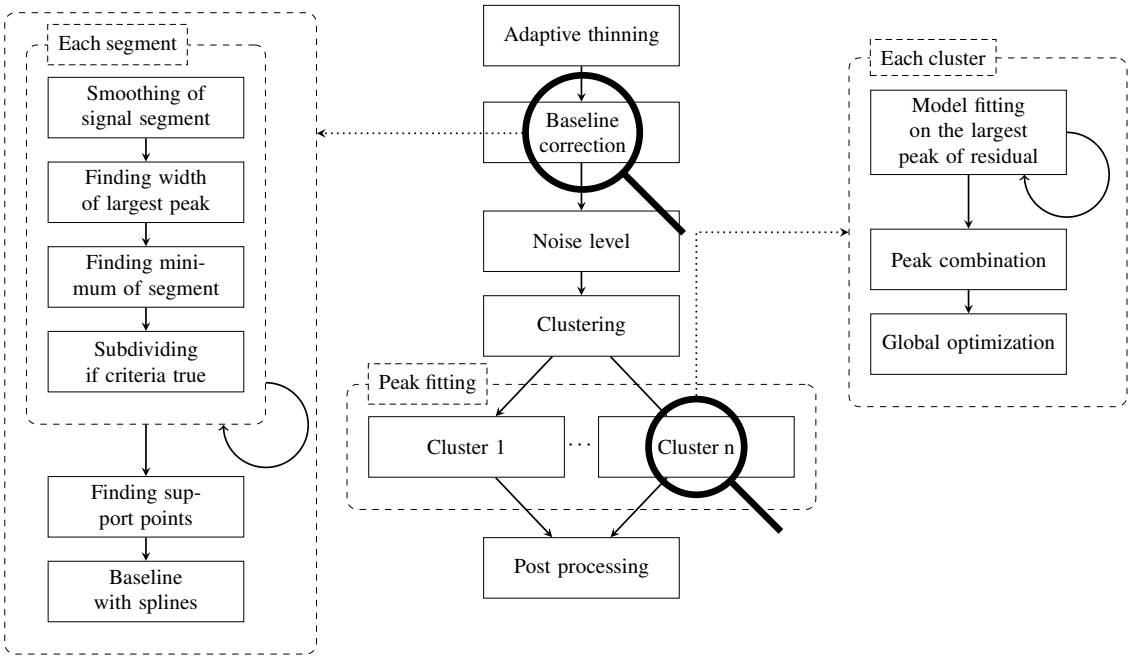

Figure 1: A flowchart describing the AutoGCA algorithm.

2.1. Standardization along the retention time axis

The AutoGCA algorithm requires only the chromatographic signal $g \in \mathbb{R}^n$ and the corresponding retention time grid $t \in \mathbb{R}^n$ as inputs. In a first step a simple linear interpolation is used to downscale g to γ_{downsc} data points per minute of retention time. This value is chosen to ensure that even narrow peaks of interest are preserved by a sufficient number of data points, but also reduces the computational time of all subsequent steps. In addition, subsequent algorithmic steps can be based on this standardized input and parameter values can be fine-tuned, see Sec. 3.2. For simplicity, the corrected signal is also denoted g.

2.2. Baseline correction

A fundamental problem in the analysis of chromatograms is the drift of the baseline, see Fig.2. This can also be interpreted as low frequency noise, which must be subtracted. It differs from classical noise, which is characterized by high-frequency stochastic signals, see [18]. Baseline correction is a well-studied field with many established methods. Considerations of baseline drift correction for GC datasets date back to the work of Wilson and McInnes [19] in 1965. Since then, several approaches have been successfully implemented and compared, e.g. [18, 20]. Various reviews have also been written on the subject, see for example [19, 21] or more recently [22]. Baseline correction is often part of more general reviews, such as [8]. An interesting alternative approach is to apply baseline correction after peak fitting [6].

Automated GC profile analysis requires a robust correction method that ensures accurate results by preventing distortion of peak areas. This is particularly challenging in areas with many overlapping peaks, because automated methods often tend to overestimate baselines, thereby cropping some areas of the peaks. We propose a method that determines a smooth curve that, firstly, lies predominantly below the signal g and secondly, is as close to the signal as possible, except in areas with peaks or peak clusters. Our algorithm is based on the assumption that the baseline of two adjacent narrow signal segments containing only a few peaks, can be determined by connecting the minimal points within each signal segment. The challenge is to reliably determine such segments.

The algorithm starts with the entire signal g and recursively divides it into segments. Each segmentation step involves smoothing the signal, detecting the largest peak, and checking whether further division is necessary. For each signal segment s with a number of $\text{len}(s)$ data points, a termination criterion is evaluated first: If $\text{len}(s) \cdot \alpha < \gamma_{\text{base}}$ holds, further segmentation is stopped. If this criterion is not met, the width of the largest peak is determined. The signal segment s is smoothed with a Savitzky-Golay filter (window width ζ_w, degree ζ_d) to obtain $\tilde{s}$. The maximum of $\tilde{s}$ indicates the center of the largest peak. Let k be the index of the corresponding data point, and we set $i = k$. To determine the peak width:

- for the right (decreasing) flank, increment i from k until the condition $\tilde{s}_i < \tilde{s}_{i+1}$ (increasing trend between two adjacent points) has happened τ times, resulting in i_r.
- for the left flank, decrement i from k analogously to find i_l.

If $(i_r - i_l)/\text{len}(s) \geq \alpha$ holds (that is, peak width is at least α times the length of the segment), the current segment is characterized as a peak. Otherwise, the peak is considered as narrow and it is assumed that the segment can be subdivided into two segments, see Fig. 3 (top). The data point index at which the segment is subdivided is determined by, firstly, subtracting a linear interpolation of $\tilde{s}$ based on its start and end points, resulting in $\bar{s}$, and secondly, finding the minimum value within the centered β range of $\bar{s}$. This is illustrated in Fig. 3 (center) where the centered range is shown in green color. The segment can be subdivided only within this central range, see the split location in Fig. 3 (center). This process is repeated recursively until segments are too small or are characterized as a peak.

The minima in each segment as well as the first and last point define a list of supporting points, see Fig. 3 (bottom). They are used to calculate a piecewise cubic Hermite interpolating polynomial b which is well suited to approximate the low frequency noise that is the baseline. To ensure that the baseline is predominantly below the signal and so the peaks are fully above the baseline, the ratio

$$\frac{\int_I \min(s - b, 0)\, \mathrm{d}t}{\int_I s - b\, \mathrm{d}t}$$

is calculated for each interval I between two adjacent supporting points. If it is greater than 0.1, then this interval is subdivided as described above and the Hermite splines are recalculated. Finally, the baseline is subtracted from the signal and the corrected chromatogram, denoted by g_B, is used for further calculations.

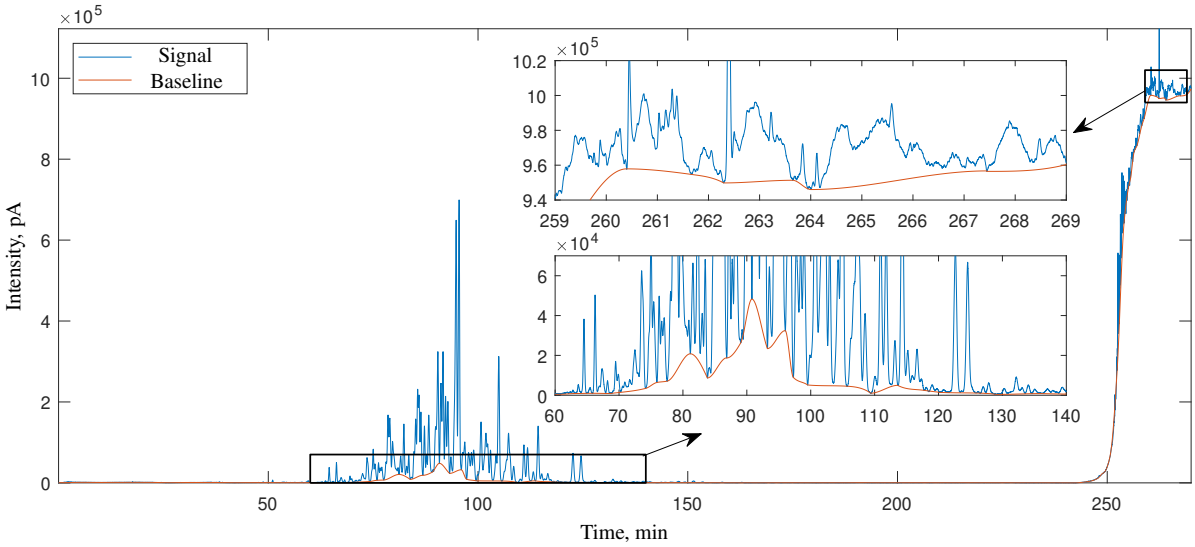

Figure 2: The dataset A1 from Sec. 3.1 after standardization, but before baseline subtraction. The calculated baseline is shown in red.

2.3. Noise level approximation

Unlike the baseline, the high-frequency noise in the signal does not usually produce very distorted results. However, it can still be difficult to distinguish between the noise and very small peaks. This has been recognized in the literature and approaches based on classical measures such as the median absolute deviation are presented in [2]. There they are used in a similar fashion - to validate peaks.

In AutoGCA, however, we assume that there exists a sufficiently long segment in g_B that does not contain any peaks. This is true in most applications. To avoid manual search for peak-free areas, we examine all possible windows w of g_B with the length of γ_{noise} data points and calculate the distance between the maximum and the minimum values of w. We find the minimum of these distances over all windows and use it to approximate the noise level. The minimum will then be automatically located in a peak-free window.

The approximated noise level L is then defined as $L = \lambda \min\{\max(w) - \min(w) \mid \text{all windows } w\}$, where the use of the prefactor λ results from the observation that the subsequent steps perform better if the noise is slightly overestimated. This noise intensity approximation is used not only to ignore peaks below the noise level, but also in several other parts of AutoGCA to achieve robustness in numerical methods or to define termination criteria.

2.4. Clustering

The use of a clustering step is justified similarly to [4], because it allows parallel execution of peak fitting, the most time-consuming part of the algorithm. Initially, g_B is divided into a number of γ_{cluster} equidistant segments. The arithmetic mean value over each segment is subtracted from the data so that the mean of the segment is zero. Then it is checked whether the segment contains values outside the interval $[-L, L]$ with the noise level approximation L of Sec. 2.3. If so, this segment and the two preceding and two following segments are assumed to contain peaks. After application to all segments, all remaining non-peak segments are neglected in the following steps. Furthermore, this defines independent clusters of subsequent segments of g_B, in a sense that peak fitting can be performed in parallel, which can drastically reduce the computation time.

2.5. Peak model

The modeling of a peak in various chromatographic applications and more generally in the natural sciences is a long-standing problem. The underlying peak model is crucial for our approach to automate peak extraction. There are classical peak models such as Gaussian, Lorentzian, Voigt or the Exponentially Modified Gaussian (EMT) among others, see e.g. [23, 24]. However, these models are severely limited when certain effects such as tailing or fronting

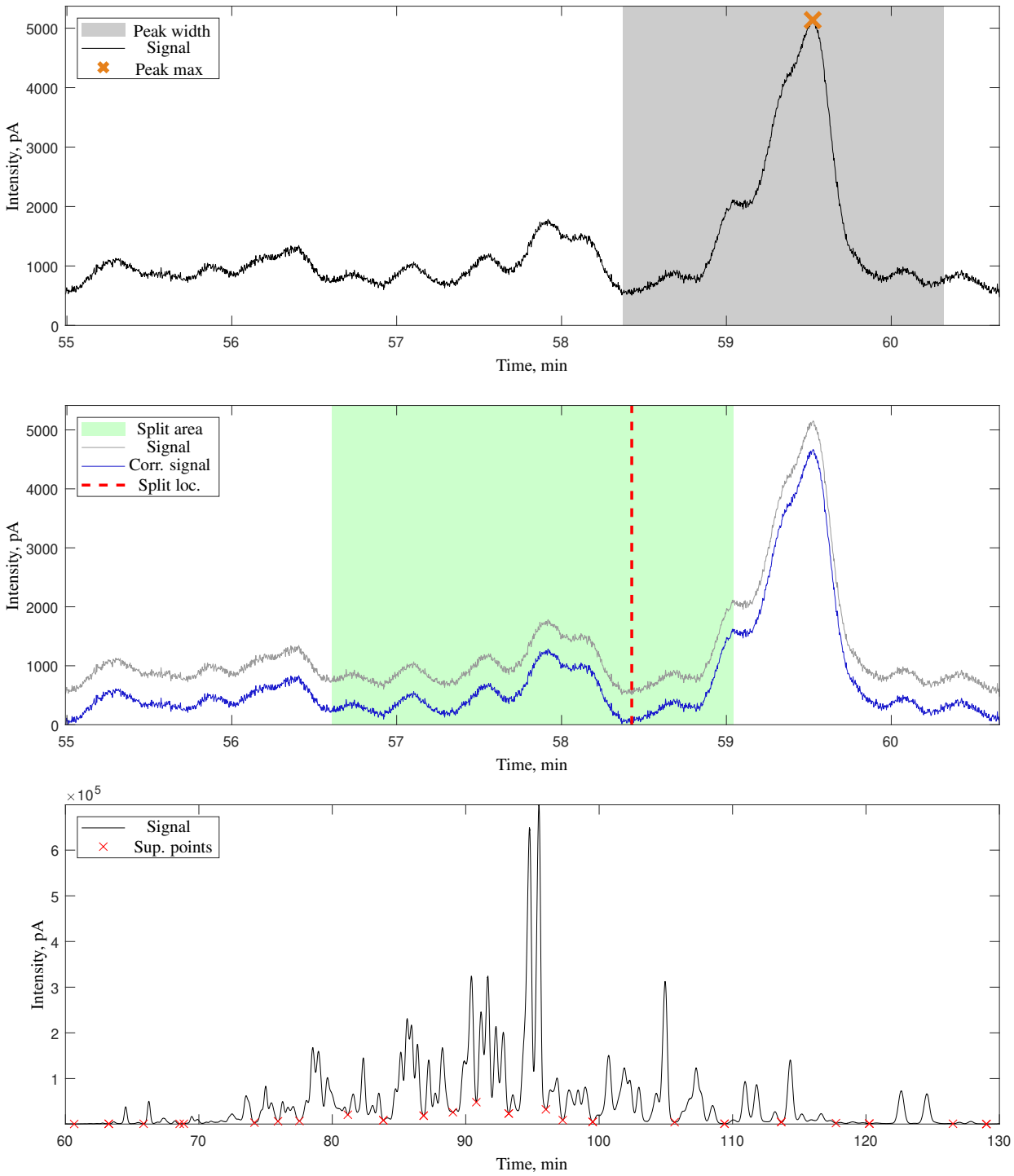

Figure 3: Visualization of the splitting stages to determine the baseline support points with the dataset A1 from Sec. 3.1. Top: The localization of the signal maximum (orange cross) and the subsequent approximation of the peak width (gray area) are shown. The approximated peak is considered narrow, because the ratio of its width to the total section length is small enough. This induces the splitting of the segment. Center: The determination of the final position of the segment split is shown. The gray line marks the original signal, while the blue line is corrected by subtracting a linear interpolant. The green area marks the centered β of the segment where splitting is allowed. Finally, the red dashed line marks the minimum of the blue signal within the green area and thus the final splitting location. Bottom: Plot of the resulting non-equidistant baseline support points of the portion of the signal with large peaks.

occur. Thus, our goal is to pragmatically approximate the actual peak and not necessarily to find a single peak model that is physically correct. The two main ideas for the reconstruction of complex peak shapes are to use a simple model that accounts for possible peak asymmetry, and to use superpositions of several of these simple models.

A simple and asymmetric peak model: The peak is divided at the center into its left and right parts, which are then considered separately in terms of their parameters. Only the peak height is shared between the two sides. Each flank is a convex combination of a Gaussian $g(x;\mu,\sigma)$ and a nearly linear curve $l(x;\mu,\sigma)$. The latter is obtained by smoothing the function

$$f(x;\mu,\sigma) = \begin{cases} 1 & |x-\mu| \leq \frac{1}{8}\sigma \\ 0 & |x-\mu| \geq \frac{17}{8}\sigma \\ \frac{17}{16} - \operatorname{sgn}(x)\frac{x}{2\sigma} & \text{else} \end{cases}$$

with a moving average with a window size of $0.05 \cdot \sigma$. Thus, the final peak model p with center μ is given by

$$p(x;\mu,h,\sigma_L,\sigma_R,\alpha_L,\alpha_R) = \begin{cases} h((1-\alpha_L)g(x;\mu,\sigma_L) + \alpha_L l(x;\mu,\sigma_L)) & x \leq \mu \\ h((1-\alpha_R)g(x;\mu,\sigma_R) + \alpha_R l(x;\mu,\sigma_R)) & x > \mu \end{cases},$$

see also Fig. 4. The linear part especially addresses the problem of *fronting* and *tailing*. This peak model only overcomes some of the limitations of such simple models, but still cannot approximate general peak shapes.

Superpositions: For experimental data, a peak is modeled as the sum of several subpeaks, each of which conforms to the proposed peak model, see Fig. 4. This allows a wide variety of peaks to be modeled while maintaining the relative simplicity of the underlying optimization problem, as each of the subpeaks can be fitted individually and then added to form the final peak.

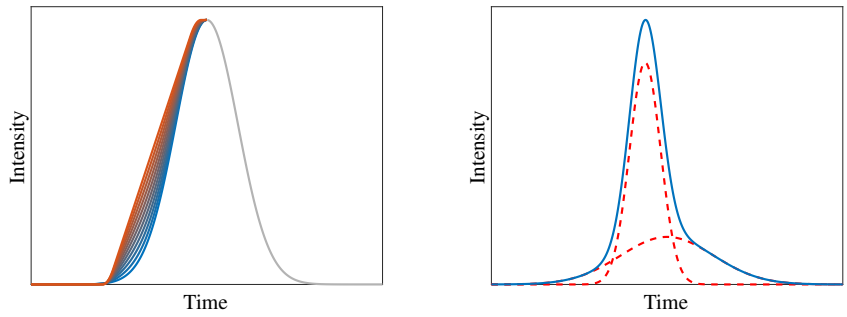

Figure 4: Left: A model of the left flank of a peak model, showing the almost linear curve (red), Gaussian (blue) and their convex combinations. Right: Two subpeaks approximating a more complex peak.

2.6. Peak detection, fitting and combination

Peak detection is a multidisciplinary problem and the approaches are transferable to different applications, e.g. a comparison of methods in mass spectrometry [25] includes methods for smoothing, baseline correction and peak finding criterion that are also useful in chromatography. In particular, derivatives are often used to detect peaks [26, 3, 6] in a variety of applications. Some practical approaches to automating peak detection are e.g. a rule based approach on derivatives [2], higher derivatives [4], intensity weighted variance [5], and probabilistic approach [27].

We have adopted a more classical residual approach on the cluster segments from Sec. 2.4. Starting with the complete cluster segment $r^{(0)}$ of g_B initial parameter estimates are determined for the largest peak. These parameters are then refined by optimization. The resulting peak profile p is subtracted from the cluster segment, updating the residual profile to $r^{(i+1)} = r^{(i)} - p$. The process is repeated until the maximum of the residual profile is less than δL, the relative model fit error $\|r^{(i+1)}\|_2^2 / \|r^{(0)}\|_2^2$ is below ε or other constraints are met, e.g. the maximum number of peaks or the maximum number of failed optimization attempts.

For the optimization, we use a weighted combined objective function based on a tight peak range and broad peak range to address multiple peaks that are close to each other. Fitting only a single peak would typically lead to over- and underestimation at the locations of neighboring peaks. In the tight range, the model should fit the current $r^{(i)}$

exactly. In the broad range, it is sufficient if the model is smaller than $r^{(i)}$. The initial peak center position μ_0 is defined by the location of the maximum of the current r, the initial $\sigma_{L,0}$ and $\sigma_{R,0}$ are roughly approximated as 1.3 times the width of the peak at 60% height and the initial convex factors is set to $\alpha_L = \alpha_R = 0.3$. These values have low impact on the final result and only serve as a rough initial approximation for the solver. The tight peak range x_t is defined as $\mu_0 \pm \psi_{\text{tight}}\sigma_{L,0}$ and broad peak range x_b is defined as $\mu_0 \pm \psi_{\text{broad}}\sigma_{L,0}$. Then the minimization problem is given by the weighted combined objective function

$$\min_{\mu,h,\sigma_L,\sigma_R,\alpha_L,\alpha_R} \|r(x_t) - p(x_t)\|_2^2 + \varphi\|\min(r(x_b) - p(x_b), 0)\|_2^2.$$

We solve the optimization problem with the nonlinear least squares solver `lsqnonlin` in MATLAB, based on the trust-region-reflective algorithm [28].

Finally, we apply a rule-based approach to decide whether the fitted peaks should be combined into subsets describing more chemically meaningful peaks. For example, if significant areas of two neighboring peaks overlap, then they are combined. We refer the reader to the source code for more details of this post-processing step. A final simultaneous fitting of all peaks often results in slightly better quality, but also adds a significant amount of computational time.

3. Results

3.1. Validation

Validation of the AutoGCA was performed with over 150 proprietary GC datasets from the cooperation with Evonik Oxeno GmbH & Co. KG. In this paper, AutoGCA is explained and evaluated with the help of GC datasets of different distillation sections from the production of C9 and C13 oxo alcohols manufactured in continuous production processes at Evonik's largest production location in Marl. They are representative of the datasets used for the validation, see Fig. 5.

While datasets indicated with A and B are complex paraffin-olefin mixtures of C12 and C8 molecules respectively, datasets indicated with C are mixtures of oxygenates, mainly oxo alcohols, ethers, aldol condensates and acetals. These distillation sections are used as solvents or intermediates for the production of solvents, as blending components for diesel fuels or as a component of industrial defoamers. A major challenge in the analysis of these datasets is the detection and separation of the large amount of isomers which is well in the three-digit range.

A further chromatogram, indicated with D, is acquired from a GC measurement of an isomeric mixture of C9-aldehydes and n-octenes using a Petrocol® DH 150 (Supelco, Inc.) column, see the published study [29]. A liquid sample was collected after 240 min during the hydroformylation of thermodynamic mixture of linear octenes catalyzed by a diphosphite modified rhodium(I) hydrido complex of the type [HRh(CO)2(P∩P)] (P∩P: 4 in [29]). The hydroformylation experiment was conducted at 120 °C and 20 bar of synthesis gas (CO/H2 = 1:1) in toluene as a solvent with a rhodium concentration of 0.75 mM and a molar ratio of [P∩P]:[Rh] = 2. The initial concentration of the octene mixture was [n-octene] = 1.62 M. The overall yield of aldehydes was 39.0% with a n-regioselectivity of 69.8%. This dataset represents a much easier situation with a relatively flat baseline and mostly isolated peaks.

AutoGCA explains the area under the curve of the dataset consistently well, see Table 1.

Dataset	Number of detected peaks	Explained area, %	Time, seconds
A1	264	97.7	57
A2	364	99.2	89
B1	224	98.7	27
B2	229	98.7	35
C1	378	98.5	40
C2	397	98.3	36
D	76	99.8	13

Table 1: A summary of the results of AutoGCA on the chosen datasets.

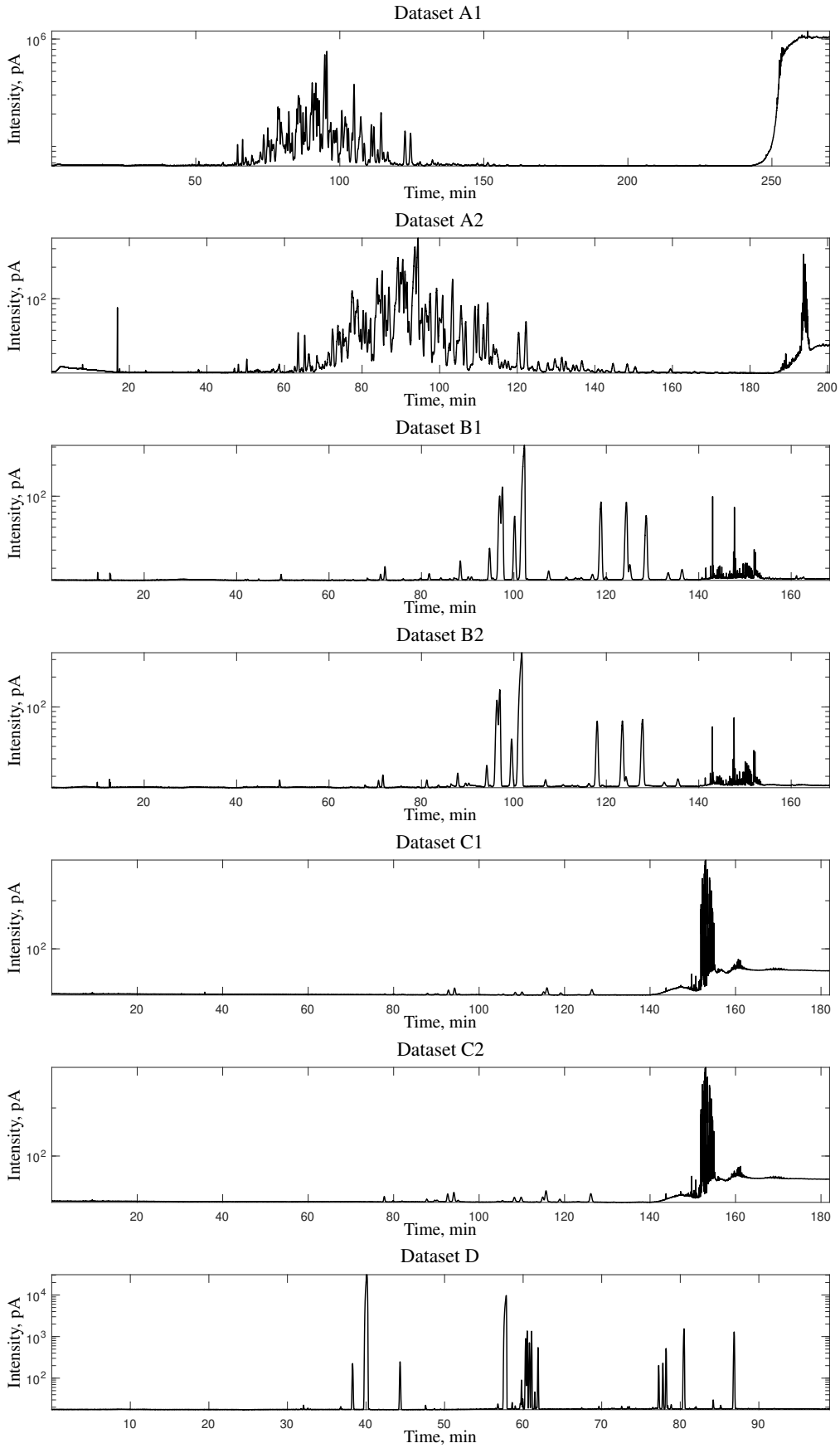

Figure 5: An overview of the various datasets from Sec.3.1 with semi-logarithmic plots.

The following sections contain an in depth analysis of AutoGCA with the help of dataset A1. This dataset is representative for some of the challenging situations that were encountered in the development of AutoGCA, for example, the dataset contains overlapping peaks including peaks on shoulders of other peaks and there is a strong baseline drift in the last part of the dataset. The total computation time of the AutoGCA algorithm was 57.1 seconds on a desktop computer with a 13th generation Intel Core i9 processor. Most of this time was used for peak fitting, which took 55.5 seconds. Six parallel MATLAB workers from the Parallel Computing Toolbox were used to reduce the computation time. The total computation time without parallelization was 92.2 seconds. The result of AutoGCA is both an integral table and a more detailed description of the peaks, see Fig. 6.

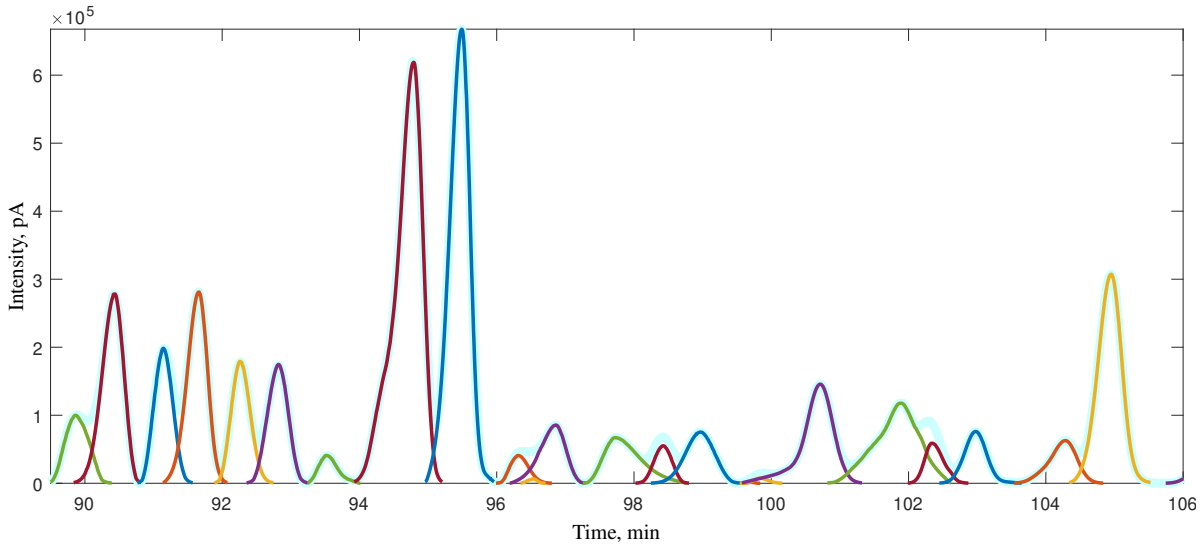

Figure 6: Some of the peaks resulting from an AutoGCA analysis of the dataset A1 from Sec. 3.1 shown in different colors. The wide cyan line in the background shows the baseline corrected raw data.

3.2. AutoGCA parameters and sensitivity analysis

All AutoGCA parameters from Sec. 2 are listed below with the optimal values.

Parameter	Default value	Description
α	0.43	peak width ratio parameter for baseline correction,
β	33%	centered range for subdivision in baseline correction,
γ_{base}	22	minimum number of data points for subdivision in baseline correction,
γ_{cluster}	300	number of equidistant segments for clustering,
γ_{downsc}	400	number of data points per minute for downscaling,
γ_{noise}	501	number of data points in a window for noise level approximation,
δ	3	coefficient for the maximum of the residual profile,
ε	0.001	coefficient for the relative model fit error,
ζ_d	3	degree for the Savitzky-Golay filter in baseline correction,
ζ_w	40	window width for the Savitzky-Golay filter in baseline correction,
η	0.1	parameter to ensure that baseline is mostly below the signal,
λ	1.3	coefficient for the approximation of the noise level,
τ	80	number of countertrends for the peak width approximation in baseline correction,
φ	10	preference factor of broad peak range in optimization,
ψ_{tight}	0.06	distance factor for the tight peak range in optimization,
ψ_{broad}	10	distance factor for the broad peak range in optimization.

AutoGCA parameters have been determined heuristically by analyzing various GC profiles and are fine-tuned to the problems that motivated AutoGCA. Finally, a sensitivity analysis has been performed, by repeating the calculation 12 times for each parameter with a different value of the specific parameter and comparing the number of the detected peaks and the explained area.

The downsampling parameter γ_{downsc} is central to the algorithm and some parameters ($\gamma_{\text{base}}, \gamma_{\text{cluster}}, \gamma_{\text{noise}}, \zeta_w, \tau$) strongly depend on it. By fixing the number of data points per minute, other parameters can also depend on absolute number of data points. We found across different datasets that 400 data points per minute of retention time adequately compresses dataset without losing the relevant information on peaks. Small changes in $\eta, \gamma_{\text{base}}, \gamma_{\text{cluster}}, \gamma_{\text{noise}}, \zeta_w$, and τ have a comparatively small influence on the final result. For α, values near 0.5 and for β values of $25\% - 50\%$ seem to work well. Noise level detection is controlled by λ, which has a large influence on what is recognized as a peak. Similarly, δ mainly influences the number of found peaks with larger values resulting in more peaks, whereas ε directly translates to the final lack of fit.

Since the source code is available online, adjustments can be made to adapt the method to different scenarios.

3.3. Comparison of AutoGCA and manual GC analysis

For comparison, this dataset A1 was manually evaluated using Agilent OpenLab CDS by two experienced chemical analysts, designated User A and User B. The manual evaluation of the dataset took approximately 10 minutes for each analyst, and only the time interval [48, 170] minutes of the data set was evaluated. The peaks in Table 2 refer only to peaks in this time interval and only the peak locations in the resulting integration tables were compared. The matching of peaks based on their location between the different results of AutoGCA, User A and User B was performed with a retention time tolerance of 0.07 minutes. Only the two closest peaks between two evaluations were matched and no peak area information was used.

Interestingly, the results of the two manual evaluations do not agree very well. This can be explained by the challenging nature of the dataset. For example, the peak at 94.8 minutes (light blue in Fig. 6) was recognized as a single peak by AutoGCA and User B, but User A split it into two separate peaks. The results in Table 2 show that AutoGCA is in reasonably good agreement with the users and the discrepancies are comparable with the difference in evaluations between both analysts. The results for small peaks have the most discrepancies with the manual evaluation.

Rate of detection for peaks

Peak prediction by *(who found the peaks)*	Peak reference by *(compared against peaks found by)*	Peak size			
		All	Small	Medium	Large
AutoGCA	User A	81.4%	62.3%	86.0%	95.0%
AutoGCA	User B	85.0%	72.7%	88.4%	90.0%
User A	AutoGCA	89.1%	88.9%	74.0%	79.2%
User B	AutoGCA	75.3%	66.7%	71.7%	85.7%
User A	User B	90.6%	87.0%	90.0%	87.5%
User B	User A	73.4%	56.0%	84.9%	100.0%
AutoGCA	Both by User A and by User B	77.1%	57.1%	86.0%	90.0%
Both by User A and by User B	AutoGCA	93.1%	93.6%	82.2%	85.7%
AutoGCA	All peaks by users	89.3%	77.9%	88.4%	95.0%

Table 2: The rate of peaks from the prediction evaluation that were also found in the reference evaluation. Only the position of the peak center was taken into account. "Both by User A and by User B" refers only to those peaks which both users found in agreement. "All peaks by users" refers to peaks found by User A, User B or both. All peaks were heuristically divided into three groups according to their area, with medium peaks ranging from 4000 pA·min to 45000 pA·min.

In Fig. 7 the evaluations are compared in more detail by showing the matched peaks in black and peaks unique to only one of the evaluations in a different color. It can be seen that some discrepancies can be explained by different splitting of peaks. Sometimes the peak centers were too far apart to be considered the same peak. Also, AutoGCA uses a different baseline than the one used in the manual evaluation, which explains some differences in the peak areas.

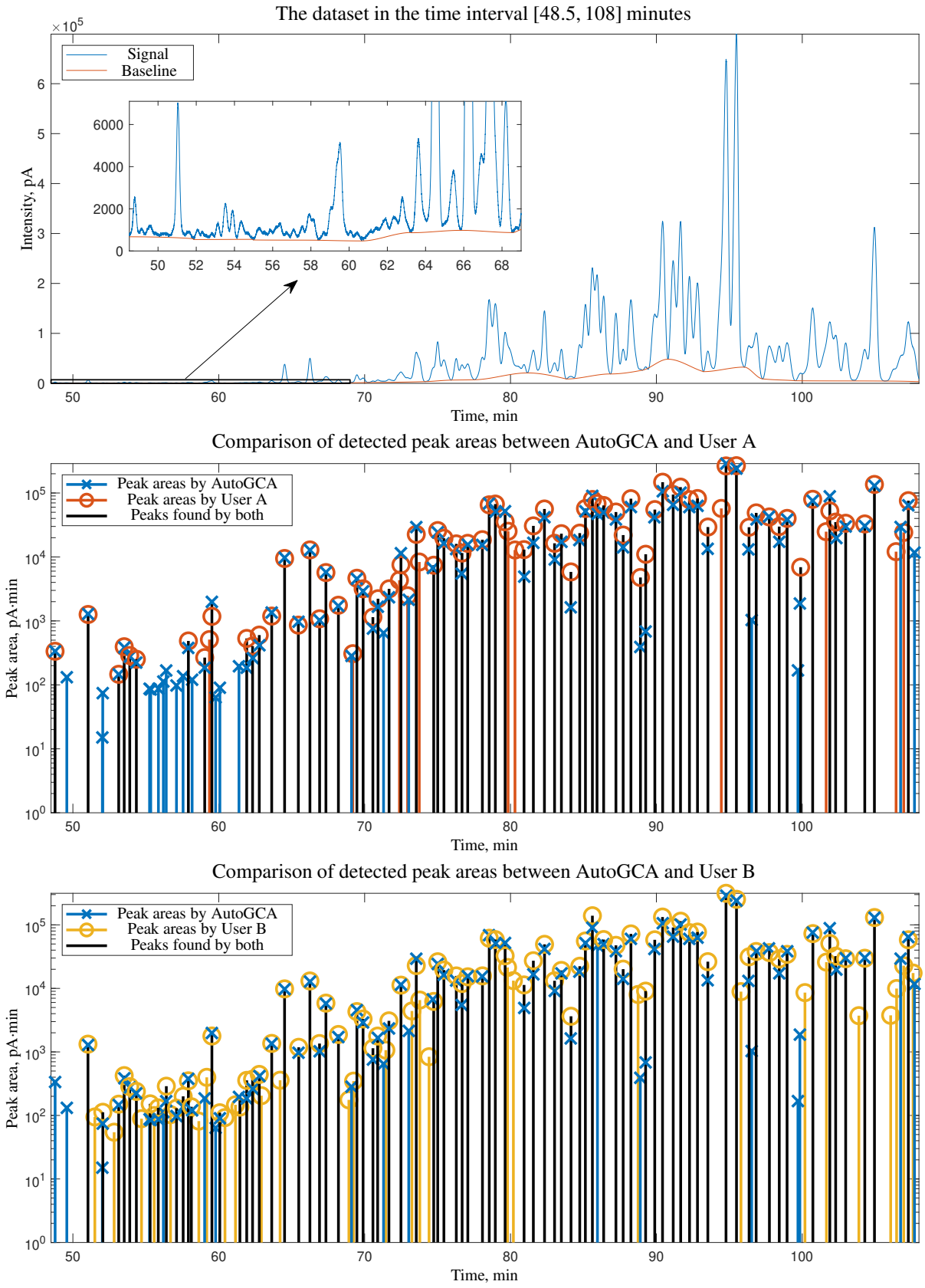

Figure 7: A comparison of the results: The dataset A1 from Sec. 3.1 (top) as well as the detected peaks and their areas.

4. Conclusion

The analysis of gas chromatogram peaks by personnel in industrial processes is often a time-consuming task. Especially when this analysis is a recurring task, an automated procedure seems to be welcome. This report covers several areas of study and presents novel methods for automating the analysis of GC data. In particular, automated peak integration of a challenging GC dataset can be achieved with similar precision to a time-consuming manual evaluation. Furthermore, an automated analysis always produces the same results, but a manual evaluation can vary from person to person. While automation of each of the required steps has been researched individually, a combined and practical automated approach is lacking in traditional software solutions. The proposed AutoGCA method has been successfully applied in industrial process analysis at Evonik Oxeno GmbH & Co. KG. We hope that this report demonstrates the potential time savings that can be achieved by automating routine procedures in analytical chemistry and serves to motivate the development of an in-depth integrated industrial solution based on the ideas of AutoGCA.

References

[1] F.H.G. Zucatelli and R.K. Nogueira. Tea component analysis: From chromatography raw data to a fully automated report. *Chromatographia*, 85:193–211, 2022. doi: 10.1007/s10337-021-04118-8.

[2] S.J. Dixon, R.G. Brereton, H.A. Soini, M.V. Novotny, and D.J. Penn. An automated method for peak detection and matching in large gas chromatography-mass spectrometry data sets. *J. Chemom.*, 20:325–340, 2006. doi: 10.1002/cem.1005.

[3] J.L. Excoffier and G. Guiochon. Automatic peak detection in chromatography. *Chromatographia*, 15:543–545, 1982. doi: 10.1007/BF02280372.

[4] G. Vivó-Truyols, J.R. Torres-Lapasió, A.M. van Nederkassel, Y. Vander Heyden, and D.L. Massart. Automatic program for peak detection and deconvolution of multi-overlapped chromatographic signals: Part I: Peak detection. *J. Chromatogr. A*, 1096(1-2):133–45, 2005. doi: 10.1016/j.chroma.2005.03.092.

[5] K.H. Jarman, D.S. Daly, K.K. Anderson, and K.L. Wahl. A new approach to automated peak detection. *Chemom. Intell. Lab. Syst.*, 69(1-2): 61–76, 2003. doi: 10.1016/S0169-7439(03)00113-8.

[6] Y.-J. Yu, Q.-L. Xia, S. Wang, B. Wang, F.-W. Xie, X.-B. Zhang, Y.-M. Ma, and H.-L. Wu. Chemometric strategy for automatic chromatographic peak detection and background drift correction in chromatographic data. *J. Chromatogr. A*, 1359:262–270, 2014. doi: 10.1016/j.chroma.2014.07.053.

[7] K. Gkountanas, I. Dagla, E. Gikas, A. Malenović, and Y. Dotsikas. Baseline correction for hplc chromatograms by using free open-source software. *Analytica*, 4(1):45–53, 2023. doi: 10.3390/analytica4010005.

[8] T.S. Bos, W.C. Knol, S.R.A. Molenaar, L.E. Niezen, P.J. Schoenmakers, G.W. Somsen, and B.W.J. Pirok. Recent applications of chemometrics in one- and two-dimensional chromatography. *J. Sep. Sci.*, 43:1678–1727, 2020. doi: 10.1002/jssc.202000011.

[9] B.W.J. Pirok, S.R.A Molenaar, L.S. Roca, and P.J. Schoenmakers. Peak-tracking algorithm for use in automated interpretive method-development tools in liquid chromatography. *Anal. Chem.*, 90(23):14011–14019, 2018. doi: 10.1021/acs.analchem.8b03929.

[10] S.R.A. Molenaar, J.H.M. Mommers, D.R. Stoll, S. Ngxangxa, A.J. de Villiers, P.J. Schoenmakers, and B.W.J. Pirok. Algorithm for tracking peaks amongst numerous datasets in comprehensive two-dimensional chromatography to enhance data analysis and interpretation. *J. Chromatogr. A*, 1705:464223, 2023. doi: 10.1016/j.chroma.2023.464223.

[11] A. Ianeselli, E. Longo, S. Poggesi, M. Montali, and E. Boselli. A complete analysis pipeline for the processing, alignment and quantification of hplc–uv wine chromatograms. *Chromatographia*, 87:159–166, 2024. doi: 10.1007/s10337-023-04301-z.

[12] A.B. Risum and R. Bro. Using deep learning to evaluate peaks in chromatographic data. *Talanta*, 204:255–260, 2019. doi: 10.1016/j.talanta.2019.05.053.

[13] D. Eilertz, M. Mitterer, and J.M. Buescher. automrm: An r package for fully automatic lc-qqq-ms data preprocessing powered by machine learning. *Anal. Chem.*, 94(16):6163–6171, 2022. doi: 10.1021/acs.analchem.1c05224.

[14] A.D. Melnikov, Y.P. Tsentalovich, and V.V. Yanshole. Deep learning for the precise peak detection in high-resolution lc–ms data. *Anal. Chem.*, 92(1):588–592, 2020. doi: 10.1021/acs.analchem.9b04811.

[15] C. Bueschl, M. Doppler, E. Varga, B. Seidl, M. Flasch, B. Warth, and J. Zanghellini. Peakbot: machine-learning-based chromatographic peak picking. *Bioinformatics*, 38(13):3422–3428, 2022. doi: 10.1093/bioinformatics/btac344.

[16] J. Obořil, C.P. Haas, M. Lübbesmeyer, R. Nicholls, T. Gressling, K.F. Jensen, G. Volpin, and J. Hillenbrand. Automated processing of chromatograms: a comprehensive python package with a gui for intelligent peak identification and deconvolution in chemical reaction analysis. *Digital Discovery*, 3(10):2041–2051, 2024. doi: 10.1039/d4dd00214h.

[17] X. Zhao, R. Aridi, J. Hume, S. Subbiah, X. Wu, H. Chung, Y. Qin, and Y.B. Gianchandani. Automatic peak detection algorithm based on continuous wavelet transform for complex chromatograms from multi-detector micro-scale gas chromatographs. *J. Chromatogr. A*, 1714: 464582, 2024. doi: 10.1016/j.chroma.2023.464582.

[18] X. Ning, I.W. Selesnick, and L. Duval. Chromatogram baseline estimation and denoising using sparsity (BEADS). *Chemom. Intell. Lab. Syst.*, 139:156–167, 2014. doi: 10.1016/j.chemolab.2014.09.014.

[19] J.D. Wilson and C.A.J. McInnes. The elimination of errors due to baseline drift in the measurement of peak areas in gas chromatography. *J. Chromatogr. A*, 19:486–494, 1965. doi: 10.1016/S0021-9673(01)99489-0.

[20] J. Schneidewind and H. Olickel. Improving data analysis in chemistry and biology through versatile baseline correction. *Chem. Methods*, 1 (2):89–100, 2021. doi: 10.1002/cmtd.202000027.

[21] Ł. Komsta. Comparison of several methods of chromatographic baseline removal with a new approach based on quantile regression. *Chromatographia*, 73:721–731, 2011. doi: 10.1007/s10337-011-1962-1.

[22] L.E. Niezen, P.J. Schoenmakers, and B.W.J. Pirok. Critical comparison of background correction algorithms used in chromatography. *Anal. Chim. Acta*, 1201:339605, 2022. doi: 10.1016/j.aca.2022.339605.

[23] J.P. Foley. Equations for chromatographic peak modeling and calculation of peak area. *Anal. Chem.*, 59(15):1984–1987, 1987. doi: 10.1021/ac00142a019.

[24] J.J. Baeza-Baeza and M.C. García-Alvarez-Coque. Characterization of chromatographic peaks using the linearly modified gaussian model. Comparison with the bi-Gaussian and the Foley and Dorsey approaches. *J. Chromatogr. A*, 1515:129–137, 2017. doi: 10.1016/j.chroma.2017.07.087.

[25] C. Yang, Z. He, and W. Yu. Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis. *BMC Bioinf.*, 10:4, 2009. doi: 10.1186/1471-2105-10-4.

[26] F. Holler, D.H. Burns, and J.B. Callis. Direct use of second derivatives in curve-fitting procedures. *Appl. Spectrosc.*, 43(5):877–882, 1989. doi: 10.1366/0003702894202292.

[27] M. Lopatka, G. Vivó-Truyols, and M.J. Sjerps. Probabilistic peak detection for first-order chromatographic data. *Anal. Chim. Acta*, 817:9–16, 2014. doi: 10.1016/j.aca.2014.02.015.

[28] T. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.*, 6(2):418–445, 1996. doi: 10.1137/0806023.

[29] D. Selent, R. Franke, C. Kubis, A. Spannenberg, W. Baumann, B. Kreidler, and A. Börner. A new diphosphite promoting highly regioselective rhodium-catalyzed hydroformylation. *Organometallics*, 30(17):4509–4514, 2011. doi: 10.1021/om2000508.