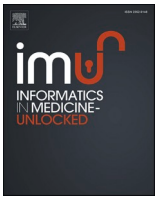
Informatics in Medicine Unlocked 34 (2022) 101113

Contents lists available at ScienceDirect

Informatics in Medicine Unlocked

journal homepage: www.elsevier.com/locate/imu



Comparative assessment of automated algorithms for the separation of one-dimensional Gaussian mixtures

Jörn Lötsch [a,b,*], Sebastian Malkusch [a], Alfred Ultsch [c]

[a] Goethe - University, Institute of Clinical Pharmacology, Theodor Stern Kai 7, 60590 Frankfurt am Main, Germany
[b] Fraunhofer Institute for Translational Medicine and Pharmacology ITMP, Theodor-Stern-Kai 7, 60596 Frankfurt am Main, Germany
[c] DataBionics Research Group, University of Marburg, Hans – Meerwein - Straße, 35032, Marburg, Germany

ARTICLE INFO

Keywords:
Data science
Machine learning
Data structure detection
Biomedical informatics

ABSTRACT

Motivation: Gaussian mixture models (GMMs) are probabilistic models commonly used in biomedical research to detect subgroup structures in data sets with one-dimensional information. Reliable model parameterization requires that the number of modes, i.e., states of the generating process, is known. However, this is rarely the case for empirically measured biomedical data. Several implementations are available that estimate GMM parameters differently. This work aims to provide a comparative evaluation of automated GMM fitting methods.

Results and conclusions: The performance of commonly used algorithms for automatic parameterization and mode number determination was compared with respect to reproducing the ground truth of generated data derived from multiple normal distributions. Four main variants of Gaussian mode number detection algorithms and five variants of GMM parameter estimation methods were tested in a combinatory scenario. The combination of best performing mode number determination algorithms and GMM parameter estimation methods was then tested on artificial and real-live data sets known to display a GMM structure. None of the tested methods correctly determined the underlying data structure consistently. The likelihood ratio test had the best performance in identifying the mode number associated with the best GMM fit of the data distribution while the Markov chain Monte Carlo (MCMC) algorithm was best for GMM parameter estimation while. The combination of the two methods of number determination algorithms and GMM parameter estimation was consistently among the best and overall outperformed the available implementations.

Implementation: An automated tool for the detection of GMM based structures in (biomedical) datasets was created based on the present results and made freely available in the R library "opGMMassessment" at https://cran.r-project.org/package=opGMMassessment.

1. Introduction

One-dimensional Gaussian mixtures are a common distribution model in medicine and psychology of data obtained from individuals belonging to different subgroups, e.g., patients versus control subjects or further substratifications. Clinical or psychological scores, on which diagnoses are based and therapeutic decisions are made, are often unidimensional variables obtained by querying or measuring a single item, e.g. pain intensity used to define pain requiring therapy [1], the blood glucose concentration used to diagnose diabetes, or the blood hemoglobin concentration used to diagnose anemia. Unidimensional variables can also be combined scores of a few to several items, such as the body mass index used to diagnose obesity, or typical outcomes of clinical or psychological questionnaires that measure, e. g. the degree of depression queried with the Beck's Depression Inventory [2], the risk of diabetes quantified with the "FINDRISK" score [3], or the degree of life impairment in fibromyalgia [4]. Further examples are sum scores of several different clinical tests and laboratory measurements such as the "MELD" score used to determine a patients' place on a liver transplant waiting list [5], the TDI score on which the diagnosis of normal, impaired, or absent olfactory function is based [6,7], and many others.

In most cases, these scores are translated into categories to which patients are then assigned. For example, the "MELD" score uses cut-offs at 15 to 19, 20 to 29, and $\geq$30 points to define mortality risk and urgency for liver transplantation [5], the olfactory "TDI" score uses cut-offs of 15.5 and 30.5 points to define diagnosis or anosmia, hyposmia, or

* Corresponding author. Goethe - University, Theodor - Stern - Kai 7, 60590 Frankfurt am Main, Germany.
E-mail address: j.loetsch@em.uni-frankfurt.de (J. Lötsch).

https://doi.org/10.1016/j.imu.2022.101113
Received 23 August 2022; Received in revised form 17 October 2022; Accepted 18 October 2022
Available online 22 October 2022

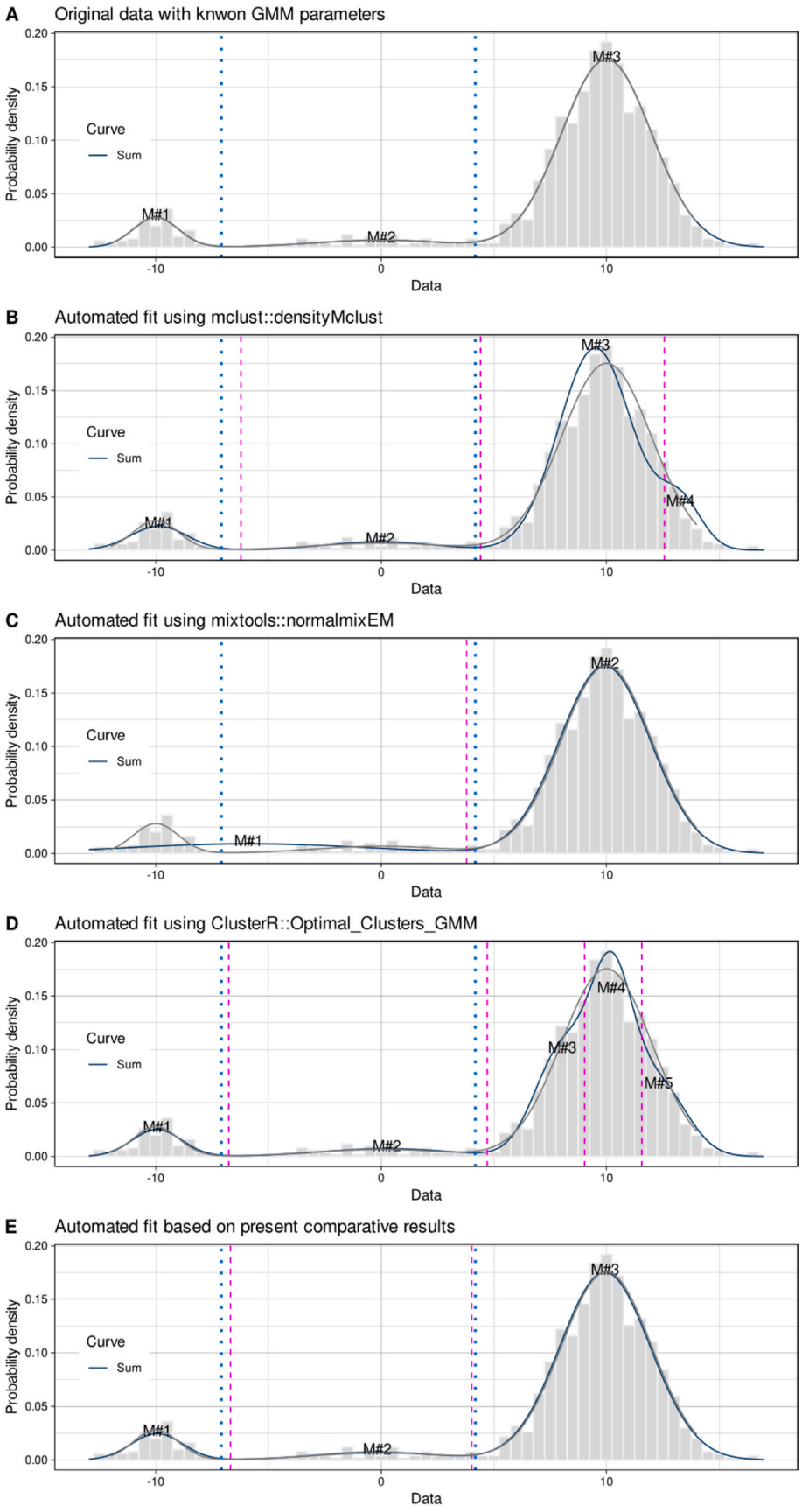


Fig. 1. Automated fit of a Gaussian mixture model to a three-modal Gaussian distributed data set. N = 1000 instances were drawn from i = 3 normal distributions with parameters means, m_i = [-10, 0, 10], standard deviations, s_i = [1–3], and weights, w_i = [0.07, 0.05, 0.88] (top panel). The data are included as "Mixture3" data set in the R package "opGMMassessment". The simulated data set was analyzed using algorithms that promise GMM evaluation "out of the box" without further statistical testing or parameter tuning, including the "densityMclust" method from the R package "mclust" (https://cran.r-project.org/package=mclust [20])), (iii) the method "normalmixEM" from the R package "mixtools" (https://cran.r-project.org/package=mixtools [21]), and the method "GMM" from the R package "ClusterR" (https://cran.r-project.org/package=ClusterR [19]). For comparison, the bottom panel shows the fit obtained when the methods that performed best in the present comparative assessments were used. The figure shows the density distribution of the data as a grey line and as a histogram. A GMM was fitted to the data (dark blue line), with the number of modes of M automatically estimated by the appropriate algorithm. Estimated Bayesian boundaries between Gaussians are shown as magenta vertical lines, and the true boundaries according to the underlying model are indicated as blue dotted vertical lines. The figure was created using the software package R (version 4.2.0 for Linux; https://CRAN.R-project.org/)[42] and the R package "ggplot2" (https://cran.r-project.org/package=ggplot2.(For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

normosmia, which define functionally absent, reduced, or normal olfactory function [8], or pain in rheumatoid arthritis is regarded as severe requiring therapy when rated >40 mm at a 100-mm visual analogue scale [9,10]. Breakpoints for subgroup assignment were often determined by experts and not defined from the data. However, breakpoints may not always be available, especially when measurements are used to categorize patients based on measurements for which no cut-off values are available from an established scoring system. Because patient subgroup detection is a common task in biomedical data analysis aimed at stratifying patients for more individualized therapies in "precision medicine," methods are needed to read subgroups from the data. Since the observed data in many cases result from sums of underlying processes, the assumption that the data follow a Gaussian distribution N(m, s) with parameters mean m and standard deviation s is justified by the central limit theorem [11]. This assumes that the data are generated by a process that uses this "likelihood function". In fact, the cutoff values resulting from the analyses of Gaussian mixture distributions seem to agree well with expert definitions of cutoff values, as illustrated by the example of pain scores in rheumatoid arthritis, where the data-driven approach of GMM modeling gave a cutoff value for subgroup separation at a visual analog scale score of 36 mm [12], which is close to the expert-defined threshold for severe pain of 40 mm mentioned above. Thus, unidimensional variables that serve as the basis for dividing patients into subgroups are ubiquitous in medical and related science and clinical practice.

A Gaussian mixture model (GMM) is a probabilistic model that describes the probability p of observing an event x. It assumes that the underlying data was generated by weighted sum of a finite number M of normal distributions $N(x|m_i,s_i)$, often termed modes, with parameters mean m_i and standard deviation s_i. The relative contribution of each of these normal distributions to the mixture is defined by the weighting w_i, which are the prior probabilities of occurrence of x in one of the modes. Consequently, the sum of the weights of all M normal distributions add up to 1. A M-modal GMM is defined as $p(x) = \sum_{i=1}^{M} w_i N(x|m_i, s_i)$ with $\sum_{i=1}^{M} w_i = 1$. With GMM, an assignment to the modes (process types) can be calculated. For a given value of x the probability with which x can be assigned to one of the modes can be calculated using the theorem of Bayes [13].

Parameter optimization methods such as the expectation maximization (EM) algorithm are commonly used to fit GMMs to one-dimensional data. However, these methods do not guarantee correct results with respect to the true organization of the underlying a data set. Therefore, it is necessary to evaluate the accuracy of the commonly used GMM parameter estimation methods to assess the correctness of the results both in determining the number of modes in the data set and the correctness of the obtained GMM parameters. As an essential parameter of the data generation process, i.e., "nature", the number of different states corresponding to the different states of data generation must be determined. If the data generation process is not fully understood, the number of states = modes must be estimated from the data. Unfortunately, there is no general method for this task. Research reports use a variety of methods that are variants of a few main approaches. These are often chosen arbitrarily depending on the available software or based on examples in similar research areas. Given its increasing popularity, attempts have been published to simplify GMM analysis for biomedical field experts. In addition to automated algorithmic heuristics [14], interactive tools [15] based on the freely available programming language R [16] are available.

Most implementations provide the number of modes as part of the results along with the GMM parameters values and he classification of data set instances into the different mode classes. A first test of common GMM construction and fitting tools on data which was generated using a pre given GMM. The generating GMM consisted of three modes which well separated modes having means $m_i = [-10, 0, 10]$ and relatively small variances compared to the distances of the means, i.e., $s_i = [1–3]$). These modes are well separated. A reconstruction of the GMM from the data should therefore be an easy task. However, an experiment with common GMM tools shows that the parameters are estimated differently and often do not capture the structure of the data generating GMM (Fig. 1). The present comparative evaluation therefore addressed different implementations of algorithms for the separation of one-dimensional Gaussian mixtures, focusing on the correct identification of the mode number and GMM parameter values in artificial data whose true values were known. From these algorithms the winning methods were combined and applied to sample data sets.

2. Methods

The formation of subgroups based on data analysis crucially depends on the analysis of the distribution of these data. Therefore, methods for GMM parameter estimation and methods for mode number determinations were comparatively evaluated. After combining the best evaluated methods, the combination was evaluated on independent data sets.

2.1. Selection of algorithms for the separation of one-dimensional Gaussian mixtures

2.1.1. Algorithms for determining the GMM parameter values

A variety of fitting algorithms for Gaussian mixtures have been developed, and the methods are available as R packages as well as in other data science environments. The present analyses considered three main variants, including methods based (i) on the widely used expectation maximization (EM) algorithm [17] (ii) on evolutionary (genetic) algorithms and (ii) on the Markov chain Monte Carlo (MCMC) algorithm [18]. The methods are available in various R implementations as well as in other data science software packages.

2.1.1.1. EM based approaches. The EM algorithm is used to find (local) maximum likelihood parameters of a statistical model. As a common approach, the EM algorithm in different facets was used as (i) the "GMM" method implemented in the R library "ClusterR" (https://cran.r-project.org/package=ClusterR [19]), (ii) the EM-based "densityMclust" method from the R library "mclust" (https://cran.r-project.org/package=mclust [20]) and (iii) the "normalmixEM" method from the R library "mixtools" (https://cran.r-project.org/package=mixtools [21]).

2.1.1.2. Genetic algorithm-based approach. As an alternative to the EM-based GMM (iv) an evolutionary algorithm [22] was used. As described in detail previously [23], for GMM adaptation, a "population" of GMMs is processed through many iterations with multiple phases in which the GMMs are mutated, selected, and recombined. In the initialization step, a population of GMMs is created with randomly drawn parameter values. In the selection step, GMMs with high fitness values are selected and judged by χ^2 statistics between the observed and estimated distributions and by the overlap between neighboring modes. During the mutation step, random individuals (GMMs) are selected and their parameters are changed, followed by recombination of selected individual GMMs. The approach was adopted from the implementation in our R library "DistributionOptimization" (https://cran.r-project.org/package=DistributionOptimization [23]), which minimizes an overlap value based on the relative number of data points covered by each possible pair of Gaussian modes of the GMM, emphasizing maximum mode separation.

2.1.1.3. Markov chain Monte Carlo based approach. A further alternative to EM-based approaches was implemented as a Markov chain Monte Carlo (MCMC) algorithm [18]. Markov Chain Monte Carlo MCMC is a class of algorithms from the family of Bayesian statistics that are capable

of drawing random samples from any mathematically defined distributions, from one-dimensional normal distributions to complex, high-dimensional distributions [24,25]. It combines the Monte Carlo method of random sampling with the sequence-generating Markov chain. Monte Carlo methods draw a large number of independent samples from a target distribution. This allows the estimation of a desired quantity by integration over a larger number of independently drawn samples [26]. Markov chains, on the other hand, define a sequence of random values, where the current value is probabilistically coupled with its predecessor [27]. Therefore, a Markov chain allows predictions about future events based solely on its present state. Such a system is called memoryless. The property that describes how much a system is influenced by its past is called a Markov property. MCMC algorithms draw a sequence of autocorrelated samples from a known distribution, with the equilibrium of the sequence settling at the desired quantity. MCMC algorithms work particularly well on high-dimensional data sets when the only thing known of the distribution is its likelihood [28]. In the present experiments, the implementation of the MCMC algorithm for GMM fitting was taken as the "NMixMCMC" function of the R library "mixAK" (https://cran.r-project.org/package=mixAK [29]).

2.1.2. Algorithms for determining the number of modes

Several approaches exist to determine the optimal number of modes for one-dimensional data. The present analyses considered four main variants summarized elsewhere [30], including (i) GMM-based approaches with goodness-of-fit tests, (ii) kernel-based approaches with critical bandwidth tests, (iii) kurtosis measures with excess mass tests, and (iv) approaches based on the analysis of the within-group dispersion compared to a reference dispersion implemented as a so-called gap statistic [14]. The methods are available in various R implementations as well as in other data science software packages, occasionally as methods for determining cluster numbers in multidimensional data sets that can be adapted to determine the number of modes in the distribution of one-dimensional data.

2.1.2.1. GMM-based approaches. In GMM-based approaches to modal distribution, for a data set consisting of n instances, the possible number of Gaussian modes is M. The determination of the optimal number of modes M must be statistically supported. Options such as the mean squared error or its root, comparisons of the shape of the original and fitted distributions using Kolmogorov-Smirnov or comparable tests (for a summary, see, e.g., Ref. [31]) that do not penalize a greater complexity of the GMMs were not considered because they tend toward larger number of modes. Goodness-of-fit criteria used in the present analyses include (i) the Akaike information criterion (AIC) [32], (ii) the Bayesian information criterion (BIC) [13], and (iii) the likelihood ratio test. They were available in our R library "AdaptGauss" (https://cran.r-project.org/package=AdaptGauss [15]).

2.1.2.2. Critical bandwidth-based approaches. Tests of critical bandwidth apply use kernel smoothing to model the probability density function (PDF) of the data and analyze the kernel parameters. Small bandwidths tend to undersmooth data regions with low structure, with the number of modes eventually equaling the number of unique observations. Large bandwidths tend to oversmooth regions with high structure, eventually reducing the number of modes toward one. The critical bandwidth is the infimum of possible kernel widths accommodating k or more than k modes [33]. Several different variants of these test principles have been proposed [30]. In the present analyses, the variant proposed by Silverman from the R library "multimode" (https://cran.r-project.org/package=multimode [34]) was used.

2.1.2.3. Excess mass-based approaches. Tests of the excess mass [35,36] base on measurement of the kurtosis of a distribution and analyze the amount of probability mass not fitting a given statistical model that is usually the uniform distribution or the class of all unimodal distributions. A mode is present where an excess of probability mass is concentrated. Several different approaches have been prosed such as implementations of Hartigan and Hartigan [37], of Fisher and Marron [38], of Cheng and Hall [39] and of Ameijeiras-Alonso [30]. Some tests were available only to test unimodality versus multimodality without further specifying the optimal number of modes M > 1 and were therefore inappropriate for the present purpose. In the present analyses, the variant proposed by Fisher and Marron was taken from the R library "multimode".

2.1.2.4. Gap criterion-based approach. The gap criterion [14] is a statistical procedure that formalizes a heuristic to determine the optimal number of modes in clustered data. The idea of the approach is to standardize the cluster-wise pooled sums of pairwise distances between the events within M clusters (W_M^{data}) by comparing it to the cluster-wise pooled sums of pairwise distances within clusters of an equally sized data set randomly drawn from a reference distribution (W_M^{ref}) whereas the events of the reference data set are assigned to clusters by the same model. For a reference distribution a uniform distribution was proposed. For the case of one-dimensional data x_i, i=1,2, …n with n being the number of observed events the Euclidian distance between two events i and i' is given by $d_{i,i'}$.If the data is clustered into M modes, $D_r = \sum_{i,i' \in C_r} d_{i,i'}$ is the sum of distances between all events of cluster C_r and the pooled sums of pairwise distances within all M clusters calculates to $W_M = \sum_{r=1}^{M} \frac{D_r}{2n_r}$. The gap between the data drawn from the reference distribution and the original data is given by $g(M) = \frac{1}{B}\sum_{b=1}^{B} \log(W_{M,b}^{ref}) - \log(W_M^{data})$. Here, B denotes the number of times the experiment is being repeated with a different set of reference data drawn from the reference distribution. The optimal number of clusters M^{opt} is the smallest M that fulfills $g(M) \geq g(M+1) - S_{M+1}$. Here, S_M is a quantity that corrects for the simulation error. It is given by $S_M = \sigma_M\sqrt{1+\frac{1}{B}}$ with σ_M denoting the standard deviation of the cluster-wise pooled sums of pairwise distances within clusters of the B sampling repeats of the reference data. The gap-criterion was included using the "clusGap" implementation from the R library "cluster" (https://cran.r-project.org/package=cluster [40]), modified for parallel processing.

2.1.2.5. Combined approach. Finally, a majority vote among several, including above-mentioned, methods for determining the number of modes, was included as provided by the R package "NbClust" (https://cran.r-project.org/package=NbClust [41]).

2.2. Experimentation

The programming work was performed in the R language [16] using the R software package [42], version 4.2.0 for Linux, which is available free of charge in the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/. Experiments were performed on 1–64 cores (threads) of an AMD Ryzen Threadripper 3970X (Advanced Micro Devices, Inc., Santa Clara, CA, USA) computer with 256 gigabytes of random-access memory (RAM) running on Ubuntu Linux 22.04 LTS (Canonical, London, UK)). Parallel processing was programmed using the implementation of the "parallel" R library provided with the R base environment [42].

2.2.1. Combination of algorithms for mode number detection and GMM parameter estimation

The goal of GMM fitting is to infer the parameters of the data generation process from the data generated by drawing a sample data set from multiple normal distributions. An automated, optimized assessment of the number and parameters of Gaussian mixtures in one-dimensional data should yield (i) GMMs with parameter values as

close as possible to the true underlying mixture, and (ii) the correct number of modes in the modally distributed data set. Both are interdependent but were initially addressed sequentially.

2.2.1.1. Selection of algorithms for estimating the parameters of a GMM. A simulation study was performed with the goal of selecting the most appropriate GMM fitting algorithm from the above options. Therefore, in three analyses, multimodal data were simulated, changing one of the simulation parameters *m*, *s*, or *w* at a time. GMMs with the true number of modes were then fitted to the data using different optimization algorithms and the resulting GMM parameters were compared to the simulation parameters to evaluate the performance of the algorithms.

2.2.1.2. Selection of algorithms for determining the number of mixtures. In a second simulation study, the reliability of different methods for determining the number of modes was evaluated. It should be noted that the actual number of modes for performance evaluation was determined based on the number of modes present in the generated datasets and not on the probability of drawing a sample from a particular mode used during the data generation process. This was done to account for the situation where a very low weight of a mode combined with a certain number of simulated instances (n = 1000 presently) may result in zero instances actually being drawn from that mode. Based on the results of the simulation experiments, promising combinations of the optimization and mode-determining algorithms were compiled.

2.2.2. Comparative evaluation of automated GMM assessment algorithms on independent datasets

The classifications in real data sets do not necessarily reflect the true class structure in the data. For example, unawareness of a subgroup structure beyond the known main structure of patients and controls, e.g., with subgroups within patients, would call into question the detection of more subgroups than expected, although the projection method may have worked correctly. This problem is well known in unsupervised data analysis and methods need therefore to be tested on datasets where hidden structures can be excluded as the process of data generation is fully known. Another example is clustering, where cluster methods occasionally fail and point either at subgroups not existing or disrupt a known group structure. Examples for this are provided, e.g., in Refs. [43, 44]. Nevertheless, unsupervised data analysis such as GMM or clustering are often a starting point for discovering additional subgroups, such as relevant subtypes of diseases that were previously considered as a single entity. In the problem addressed in the present paper, it is not the GMM calculation for known subgroups, which can be easily done based on the means, standard deviations and weights for each subgroup, but the aim is to detect such subgroups in unidimensional data by fitting GMMs to the data. It is therefore difficult to judge whether a discovered modal distribution is correct, making it difficult to compare automated GMM fitting methods on real-life data when the true structure of the underlying data generating process is not precisely known. However, an emphasis in this step of the experiments was that the data sets were created independently of the experiments at hand up to this set.

To evaluate the performance of the presently proposed GMM assessment approach on real data in comparison to alternative implementations that also promise "out of the box" GMM assessment without further statistical testing or parameter tuning, experiments were conducted. This assesses the clustering accuracy between the GMM-based classification according to the Bayesian boundaries using the optimized parameters of the models, and the available prior classification.

Alternatives to the present GMM assessment approach, called (i) "opGMMassessment", included (ii) the "densityMclust" method from the R package "mclust" (see above) without specifying the number of modes, (iii) the "normalmixEM" method from the R package "mixtools" with the parameters "arbmean" and "arbvar" set to "TRUE", triggering an automatic determination of appropriate initial values including the

Table 1
Deviations from the true GMM parameters obtained with different fit algorithms for their (medians and interquartile ranges).

Type of algorithm		Changing means			Changing SD			Changing weights			Sum of ranks
		m_i	s_i	w_i	m_i	s_i	w_i	m_i	s_i	w_i	
	True values	[-10,0,10] to [0,0,0]	[3,1,3]	[0.2,0.1, 0.7]	[-10,0,10]	[2,0.2,4] to [2,9,6,4]	[0.1,0.05,0.85]	[-10,0,10]	[1,2,3]	[0.33, 0.33, 0.33] to [0.000811333728939, 1125596625511 8 · 10^{-5}, 1-(w_1+w_2)]	
Median of differences to true values											
EM based	ClusterR:: GMM	0.82 (−0.24 - 1.79)	−0.15 (−0.58 - 0.68)	−0.01 (−0.16 - 0.18)	2.24 (1.21–6.99)	−0.71 (−0.84 - 0.9)	0.03 (−0.37 - 0.35)	2.21 (0.87–8.04)	0.01 (−0.55 - 1.73)	0 (−0.4 - 0.4)	0.82 (−0.24 - 1.79)
	densityMclust	1.16 (−0.03 - 2.81)	−0.02 (−0.36 - 1.73)	−0.04 (−0.35 - 0.34)	0.88 (0.35–8.85)	0.15 (−0.88 - 1.22)	0.01 (−0.45 - 0.44)	0.26 (0.02–7.65)	0.02 (−0.03 - 2.3)	0 (−0.2 - 0.2)	1.16 (−0.03 - 2.81)
	normalmixEM	0.4 (−0.07 - 1.81)	−0.03 (−0.53 - 1.05)	0 (−0.16 - 0.21)	0.33 (0.07–6.93)	0.02 (−0.4 - 0.52)	0 (−0.1 - 0.09)	0.02 (0–4.49)	−0.01 (−0.15 - 0.13)	0 (−0.01 - 0.01)	0.4 (−0.07 - 1.81)
Genetic	DO	0.17 (−0.84 - 0.99)	−0.28 (−0.62 - 0.82)	−0.03 (−0.11 - 0.13)	1.9 (0.36–5.3)	−0.45 (−0.8 - 0.15)	0.01 (−0.23 - 0.23)	0.8 (−0.1 - 5.2)	0.05 (−0.42 - 0.86)	0 (−0.01 - 0.05)	0.17 (−0.84 - 0.99)
MCMC	NMixMCMC	0.11 (0.03–0.84)	0.11 (0–0.73)	0 (−0.01 - 0.05)	0.22 (0.12–3.16)	0.04 (−0.21 - 0.19)	0 (−0.09 - 0.08)	0.05 (0–1.58)	0.04 (0–1.26)	0 (0–0)	0.11 (0.03–0.84)

Table 2
Ranking of fit algorithms for their correct capture of the true parameters of GMMs, using either the median of the differences to the true GMM parameters or their modes to accommodate the right-skewed distribution (Fig. XXX). Lower ranks indicate smaller differences, i.e., better precision in estimating the GMM parameters. Ranking has been done using the R command "rank" without further switches.

Type of algorithm		Changing means			Changing SD			Changing weights			Sum of ranks
		m_i	s_i	w_i	m_i	s_i	w_i	m_i	s_i	w_i	
Rank order of median of differences to true values											
EM based	ClusterR::GMM	4	2	3	5	4	4	5	5	5	37
	densityMclust	5	4	5	3	5	5	3	3	4	37
	normalmixEM	2	3	4	2	2	2	1	2	2	**20**
Genetic	DO	3	5	2	4	3	3	4	4	3	31
MCMC	NMixMCMC	1	1	1	1	1	1	2	1	1	**10**
Rank order of modes of differences to true values											
EM based	ClusterR::GMM	4	4	5	5	4	2	2	2	2.5	30.5
	densityMclust	5	5	4	3	5	4	5	4	5	40
	normalmixEM	2	1	1	2	1	1	1	1	2.5	**12.5**
Genetic	DO	3	3	3	4	3	5	4	3	2.5	30.5
MCMC	NMixMCMC	1	2	2	1	2	3	3	5	2.5	**21.5**

Table 3
Ranking of the criteria for determining the number of mixtures. Lower ranks indicate smaller differences between the determined number of modes and the actual number of modes, i.e., better estimation of the model underlying the data. Ranking has been done using the R command "rank" without further switches. AIC: Akaike information criterion; BIC: Bayesian information criterion, EM: expectation maximization algorithm, FM: Fisher and Marron method [38], GMM: Gaussian mixture modeling, LR: likelihood ratio test, MCMC: Markov chain Monte Carlo (MCMC) algorithm, SI: Silverman method [34].

Type of algorithm		Changing means		Changing SD		Changing weights		Sum of ranks
		MCMC	normalmixEM	MCMC	normalmixEM	MCMC	normalmixEM	
GMM-based with goodness-of-fit testing	AIC	2	2	2	3	1	3	**13**
	BIC	4	4	4	4	3	2	21
	LR	3	3	3	2	2	1	**14**
Critical bandwidth	SI	7	7	7	7	7	7	42
Excess mass	FM	6	6	6	6	6	6	36
Gap	GAP	5	5	5	5	5	5	30
Combined	NbClust	1	1	1	1	4	4	**12**

number of mixes, and (iv) the "GMM" method from the R package "ClusterR" in combination with "Optimal_Clusters_GMM" for mode number detection from the same library.

2.3. Implementation

The methods used in the present comparative assessments were assembled in an R package "opGMMassessment", freely available at https://cran.r-project.org/package=opGMMassessment. The GMM evaluation can be called with opGMMassessment(Data, FitAlg = "MCMC", Criterion = "LR", MaxModes = 8, MaxCores = getOption("mc.cores", 2L), PlotIt = FALSE, KS = TRUE, Seed). For the fitting algorithm ("FitAlg"), "Markov chain Monte Carlo" was selected as the default, and for the mode number detection method ("Criterion"), the likelihood ratio test ("LR") was selected as the default, according to the results of the present comparative evaluations (Table 1 and Table 3). The parameter "PlotIt" creates a GMM plot, and "KS" provides a Kolmogorov-Smirnov test [45] of the final fit compared to the original data. The library imports functions for GMM fitting and mode number determination from the above R libraries. The M modes of the GMM are fitted simultaneously using parallel processing unless (vii) the parameter "MaxCores" is set at a value of 1. Parallel computing is implemented from the "parallel" library provided with the R base environment, which provided faster operation on Linux systems than packages "doParallel" (https://cran.r-project.org/package=doParallel [46]) and "foreach" (https://cran.r-project.org/package=foreach [47]), which needed to be used on systems running on Windows™ (Microsoft Corporation, Redmond, WA, USA). More detailed hyperparameter settings are beyond the scope of this report and are available through the R library help function.

3. Results

3.1. Simulation studies

The simulation study was designed to compare various methods for either GMM parameter estimation or mode number determination. One-dimensional data sets with 1000 instances were drawn from three normal distributions with different probabilities resulting in a three modal data set (M = 3 modes). Each of the three normal distributions is characterized by its mean value m_i and standard deviation s_i. and a weighting parameter w_i. The simulated data sets are therefore based on 3*M-1=8 simulation parameters ($w_3 = 1\text{-}w_1\text{-}w_2$). The parameters of five different fitting algorithms were optimized on the simulated data and compared to the simulation parameters to evaluate the performances of the different algorithms.

In three simulation studies data sets were generated by Monte Carlo simulations with modified simulation parameters. Here, only one of the three simulation parameters was varied during each of the three simulation studies, while the other two were kept fix. In detail, the following scenarios were investigated:

1. **Variation of the mean values:** The values of m_i were varied from $m_{1\text{-}3} = [-10, 0, 10]$ in 48 steps to [0, 0, 0] by decreasing or increasing, respectively, m_1 and m_3 in equal steps. The values of s_i and w_i were kept fix at $s_{1\text{-}3} = [3,1,3]$ and $w_{1\text{-}3} = [0.2, 0.1, 0.7]$ (Fig. 2 A).
2. **Variation of the standard deviation:** The values of s_i were varied from $s_{1\text{-}3} = [2, 0.2, 4]$ in 48 steps to [2, 9.6, 4] by increasing s_2 by 0.2 during each step. The values of m_i and w_i were kept fix at $m_{1\text{-}3} = [-10, 0, 10]$ and $w_{1\text{-}3} = [0.1,0.05,0.85]$ (Fig. 2C).
3. **Variation of weights:** The values of w_i were varied from equal weights $w_{1\text{-}3} = [0.33, 0.33, 1\text{-}(w_1+w_2]$. The values of m_i and s_i were

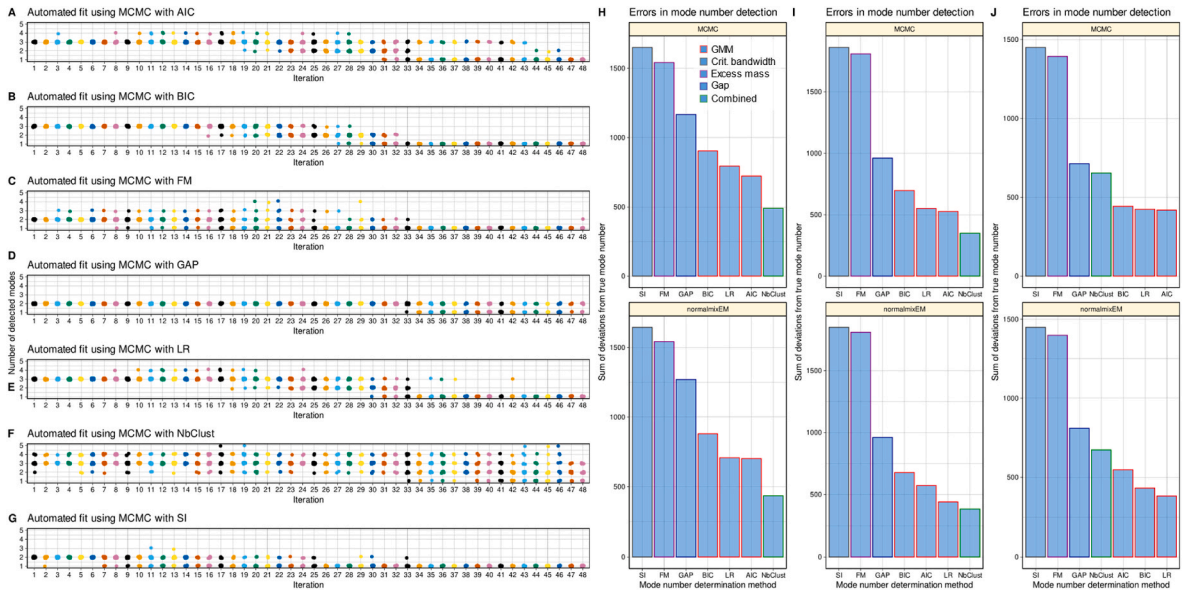


Fig. 2. Comparative evaluation of methods for the determination of the number of modes in the mixture. Mode number detection was performed in 10 repetitions each using seven different methods, including goodness-of-fit-based selection criteria using AIC, BIC, or the likelihood ratio test, critical bandwidth-based methods implemented as the Silverman criterion, excess mass-based methods implemented as the Fisher and Marron criteria, the GAP criterion, and majority voting of several criteria from the R package "NbClust" (https://cran.r-project.org/package=NbClust [41]). Experiments were performed with 48 iterations of each data set, changing either the means, standard deviations, or weights of the three normal distributions that underlay the data generating process (parameter values in Results section). The number of modes detected was compared to the true number of modes read from the data generating process. The number of modes is presented as dot plots, with the dots jittered to allow identification of all 10 values. The colors are arbitrary and are used only to improve the association of the results with the iteration count. They were taken from the "colorblind_pal" palette" provided with the R library "ggthemes" (https://cran.r-project.org/package=ggthemes [62]). **A** and **G**: Dot plots of the actual number of modes in the experimental scenario, where the mean values change in 48 equal steps from m_i = [-10,0,10] to m_i = [0,0,0], the MCMC-based GMM fitting method. The true number of modes was always M = 3. **H:** Bar plots of the sum of the absolute deviations of the detected number of modes from the true number of modes, sorted in descending order for the different mode detection methods. The upper panel shows the result of fitting the GMM with the MCMC-based method, and the lower panel shows the result of fitting the GMM with the modified EM-based method. **I:** Bar graphs of the errors in the experimental scenario where the standard deviation was changed stepwise. **J:** Bar plots of the errors in the experimental scenario where the widths were changed stepwise. The figure was created using the software package R (version 4.2.0 for Linux; https://CRAN.R-project.org/)[42] and the R package "ggplot2" (https://cran.r-project.org/package=ggplot2 [61]). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

kept fix at $m_{1\text{-}3}$ = [-10, 0, 10] and s_i at $s_{1\text{-}3}$ = [1,2,3], [w_i,w_i,1-(w_1+w_2)] changing w_1 from 0.33 to $w_f = 0.000811333728939$ and changing w2 from 0.33 to $w_f = 9.11255966255118 \cdot 10^{-5}$, both in 48 steps (Fig. 2 E).

Each of the experiments was conducted in 20 samples of size n = 1000 drawn from the 48 different distributions, each created as described above, resulting in a total of 960 runs per scenario for the respective 48 parameter sets. Different values for the seed parameter were set for each replicate run.

3.1.1. *Comparative evaluation of the methods for GMM parameter estimation*

The fitting algorithms were comparatively evaluated for their ability to determine from a one-dimensional data set the simulation parameters that had been used for data-generating. In these evaluations, methods from each of the three families mentioned above were included, i.e., methods based (i) on the EM algorithm (methods "GMM" from the R library "ClusterR", "densityMclust" from the R library "mclust", and "normalmixEM" from the R library "mixtools"), an evolutionary (genetic) algorithm ("DistributionOptimization") and on the MCMC algorithm ("NMixMCMC" from the R library "mixAK") were tested.

The results of these experiments showed that one of the implementations of the EM algorithm, i.e., the modified EM method provided as "normalmixEM" in the R library "mixtools" and the MCMC-based GMM fitting algorithm "NMixMCMC" implemented in the R library "mixAK", estimated GMM parameters with the least differences from the true parameter values used for the data simulations (Table 1 and Fig. 3). This can be seen by the ranking of the absolute differences between the estimated values of m_i, s_i, and w_i and the true values used to create the respective datasets (Table 2), with lower ranks indicating smaller differences, and summing these ranks for each fitting algorithm yielded the lowest ranks. "NormalmixEM" and "NMixMCMC" showed the best performance indicators, both when using the either the median of the differences or the mode as criteria.

3.1.2. *Comparative evaluation of methods for the determination of the mode number*

Based on above results, "NormalmixEM" and "NMixMCMC" were selected for the evaluations performed to select the best method for determining the number of modes. They were used for the same data sets as above in the three scenarios, but with the task of determining the number of modes. In these evaluations, methods from each of the five types of approaches to mode number detection were included, i.e., GMM-based approaches where the number of modes was determined via the best fit, assessing goodness-of-fit with including (i - iii) AIC, BIC, and likelihood ratio (LR) tests, critical bandwidth-based methods including (iv) the Silverman criterion, excess mass-based methods including (v) the Fisher and Marron criterion, and (vi) the GAP criterion were evaluated. A collection of cluster number detection methods (vii) was added from the "NbClust" R package, which uses above or similar methods for cluster number determination and makes the final decision as a majority vote among them.

Based on the results of these experiments on the three scenarios mentioned above, the algorithms were ranked according to their ability to determine the actual number of modes in the simulated data sets (Tables 3 and i.e., $\sum_{i,j} \left| M_{determined,i,j} - M_{true,i,j} \right|$ for i = 2 fitting algorithms and j = 7 criteria for the determination of M). The lowest ranks,

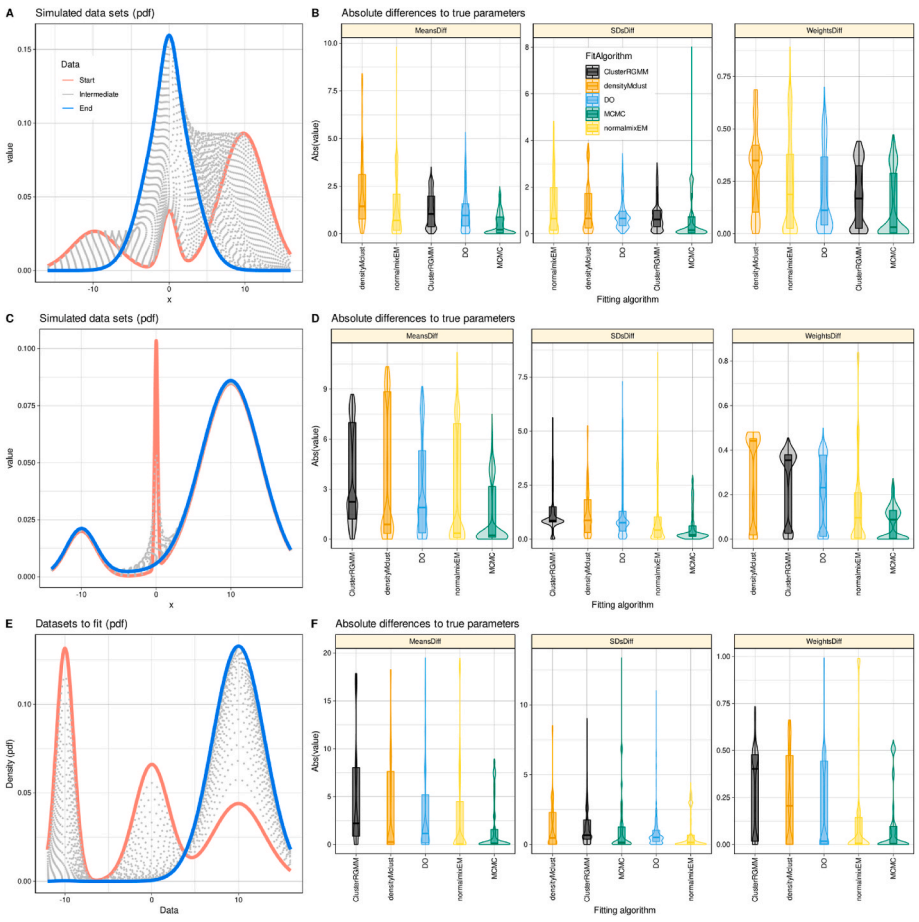


Fig. 3. Comparative evaluation of methods for fitting Gaussian mixture models to multimodally distributed data. GMM fitting was performed in 10 repetitions each using five different methods, including the "GMM" method implemented in the R library "ClusterR", the genetic "DistributionOptimization" algorithm (DO), the EM-based "densityMclust" method from the R library "mclust", the "normalmixEM" method from the R library "mixtools", and the MCMC based "NMixMCMC" algorithm from the R library "mixAK". Experiments were performed with 48 iterations of each data set, changing either the means, standard deviations, or weights of a Gaussian model with M = 3 modes (original parameter values in Results section). The obtained parameters of the GMM (means, standard deviations, weights) were compared with the values used for data set generation. Three scenarios were assessed, with either changing means (panel **A**), standard deviation (panel **C**) or weights (panel **E**) of the GMM. Panels **B**, **D** and **F** show the differences of the obtained GMM parameters to the original parameters in descending order of magnitude. The data are shown as violin plots, overlaid with box plots where the boxes were constructed using the minimum, quartiles, median (solid line inside the box) and maximum of these values. The whiskers add 1.5 times the interquartile range (IQR) to the 75th percentile or subtract 1.5 times the IQR from the 25th percentile. The figure was created using the software package R (version 4.2.0 for Linux; https://CRAN.R-project.org/)[42] and the R package "ggplot2" (https://cran.r-project.org/package=ggplot2 [61]).

indicating the lowest error rates, were obtained when using majority voting under a collection of mode number methods such as those provided by the "NbClust" command of the R package of the same name, or when using GMM-based approaches to determine the mode number and assessing the goodness of fit by the likelihood ratio test or the AIC. By contrast, the critical bandwidth, excess mass, or GAP criteria led to more errors, up to markedly poor results with the Silverman or Fisher and Marron criteria, which always indicated unimodality, for example, in the experiments with successive changes in the weights of the GMM (Fig. 2).

3.1.3. *Combination of mode number determination and GMM parameter estimation methods*

In the above results, the likelihood ratio test was among the top-performing tests for mode number de-termination, while the Markov chain Monte Carlo (MCMC) algorithm was among the top-performing tests for GMM parameter estimation. The combination of the two methods was applied to one-dimensional sample data sets for which a modal distribution was known or could be reasonably assumed from the topical context of their creation.

3.2. *Application on independent data sets*

In the above results, the likelihood ratio test was among the top-performing tests for mode number de-termination, while the Markov chain Monte Carlo (MCMC) algorithm was among the top-performing tests for GMM parameter estimation. The combination of the two methods was applied to one-dimensional sample data sets for which a modal distribution was known or could be reasonably assumed from the topical context of their creation. Comparative evaluations were performed involving R implementations of algorithms promising automatic GMM assessment, i.e., the present combination assembled in the above-mentioned R package "opGMMassessment", and additionally the method "densityMclust" from the R package "mclust", the method "normalmixEM" from the R package "mixtools", and the method "GMM" from the R package "ClusterR". Since "out-of-the-box" results were desired, all methods were run with the respective default hyperparameter settings. The experiments were performed in 20 replicates as above and setting the maximum number of modes at M + 3.

Comparative evaluations were performed involving R implementations of algorithms promising automatic GMM assessment, i.e., the present combination assembled in the above-mentioned R package "opGMMassessment", and additionally the method "densityMclust" from the R package "mclust", the method "normalmixEM" from the R package "mixtools", and the method "GMM" from the R package "ClusterR". Since "out-of-the-box" results were desired, all methods were run with the respective default hyperparameter settings. The experiments were performed in 20 replicates as above and setting the maximum number of modes at M + 3.

3.2.1. *One-dimensional three-modal data set from a machine-learning textbook*

A first example data set was taken from a textbook on machine learning [48] where it served as an example for GMM modeling with Python (e.g., https://www.astroml.org/book_figures/chapter4/fig_GMM_1D.html). The GMM is defined as $m_{1\text{-}3} = [-1, 0, 3]$, $s_{1\text{-}3} = [1.5, 1, 0.5]$ and $w_{1\text{-}3} = [0.35, 0.5, 0.15]$. N = 2000 data points were generated for the present tests. The combination of Markov chain Monte Carlo fitting with a likelihood ratio test proposed here and the mclust:densityMclust method were able to detect that the data were trimodally

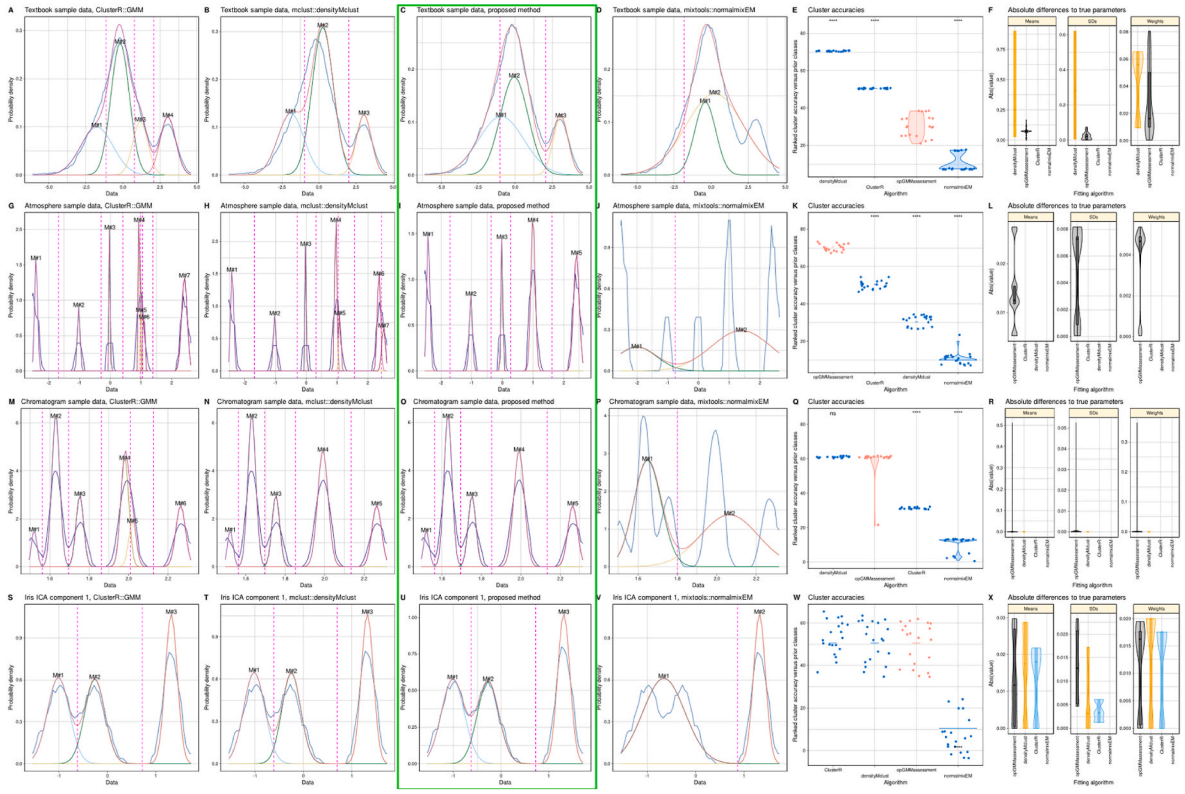


Fig. 4. Fits and classifications obtained with four alternative automatic algorithms applied on different data sets. Data sets are shown in the for rows, from to bottom consisting of (i) a one-dimensional three-modal data set with n = 2000 points generated according to an example given in a **textbook** on machine learning [48], with $m_{1\text{-}3} = [-1, 0, 3]$, $s_{1\text{-}3} = [1.5, 1, 0.5]$ and $w_{1\text{-}3} = [0.35, 0.5, 0.15]$ (**panels A – F**), on a one-dimensional five-modal data set prosed in the "**Atmosphere**" journal [49], with n = 100 data points drawn from a mixture with GMM parameters $m_{1\text{-}5} = [-2.4, -1.0, 0, 1.0, 2.4]$, $s_{1\text{-}5} = [0.05, 0.07, 0.02, 0.06, 0.1]$ and $w_{1\text{-}5} = [0.2, 0.1, 0.1, 0.3, 0.3]$ (**panels G – L**), on a data set created from a **chromatogram** (Fig. 5) with five different lysophosphatidic acids (LPA 16:0, 18:0, 18:3, 20:0, and 20:4) (**panels M – R**), and to the first independent component obtained from the **Iris** flower data set [50,51] (**panels S – X**). Automatic GMM fitting and subsequent class assignment of instances after calculating Bayesian boundaries from the obtained GMM parameter values was performed in 20 replications using different values of seed. The alternatives tested included the present proposal "opGMMassessment" (framed in green), "densityMclust" from the R package "mclust", "normalmixEM" from the R package "mixtools" and "GMM" from the R package "ClusterR". The default parameter settings of the respective R packages were used. In the first four panels in each row from the left (i.e., panels A, B, C, D, G, H., I, J, M, N, O, P, S, T, U, V, see also above), the empirical distribution of the data, estimated using the Pareto density estimation (PDE [63]; black lines), is shown along with the GMM fits (red line) and the single Gaussians (differently colored lines). Bayesian boundaries between Gaussians are shown as magenta vertical dashed lines. In the second lest panels to the right (i.e., panels E, K, Q, W, see also above), ranks of classification accuracies against the prior classification are shown as individual dots, overlaid with a violin plot. Algorithms are presented in descending order of ranks achieved; higher ranks indicate better performance in assigning an instance to its original class. The statistical significance of differences to the presently proposed "opGMMassessment" method, assessed by means of Wilcoxon-Mann-Whitney U tests [64, 65], is indicated at the top line as stars: *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$. In the right panels in each row (i.e., panels F, L, R, X, see also above) the absolute differences of the obtained GMM parameters to the original parameters in descending order of magnitude. The data are shown as violin plots, overlaid with box plots where the boxes were constructed using the minimum, quartiles, median (solid line inside the box) and maximum of these values. The whiskers add 1.5 times the interquartile range (IQR) to the 75th percentile or subtract 1.5 times the IQR from the 25th percentile. The figure was created using the software package R (version 4.2.0 for Linux; https://CRAN.R-project.org/) [42] and the R package "ggplot2" (https://cran.r-project.org/package=ggplot2 [61]). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

distributed (Fig. 4), the latter providing higher cluster assignment accuracy, but the former providing GMM parameters closer to the truth. The mclust:densityMclust method occasionally provided better cluster accuracies, likely due to the low weight of a third cluster; however, it could not detect trimodality and thus failed at the task, as did ClusterR: GMM by detecting four modes.

3.2.2. *One-dimensional five-modal data set from "Atmosphere" journal*

In a report on the development of a cluster sampling filter for geoscience data [49], a prior distribution approximated by a Gaussian mixture model (GMM) was used as a starting point for further elaboration of the method. This independently constructed Gaussian mixture was used here to test the automatic GMM fitting tools under the name "Atmosphere data," which is derived from the name of the journal in which the report was published. Specifically, the authors of the cited paper had defined a five-modal problem for their experiments (equation 26 in https://www.mdpi.com/2073-4433/9/6/213/htm) with GMM parameters $m_{1\text{-}5} = [-2.4, -1.0, 0, 1.0, 2.4]$, $s_{1\text{-}5} = [0.05, 0.07, 0.02, 0.06, 0.1]$ and $w_{1\text{-}5} = [0.2, 0.1, 0.1, 0.3, 0.3]$. As by the authors of the cited paper, n = 100 data set instances were generated. The five-modal distribution of this dataset was detected only by the currently proposed combination of Markov chain Monte Carlo fitting with a likelihood ratio test to detect the modal number (Fig. 4). The worst performance was provided by mixtools:normalmixEM, which suggested a bimodal distribution.

3.2.3. *Chromatogram one-dimensional five-modal data set*

Chromatographic separation of substance mixtures is carried out prior to the concentration determination of the individual components. The substance mixture passed through a stationary phase that is traversed at a different speed by the individual components. The resulting chromatogram is a representation of the frequency with which

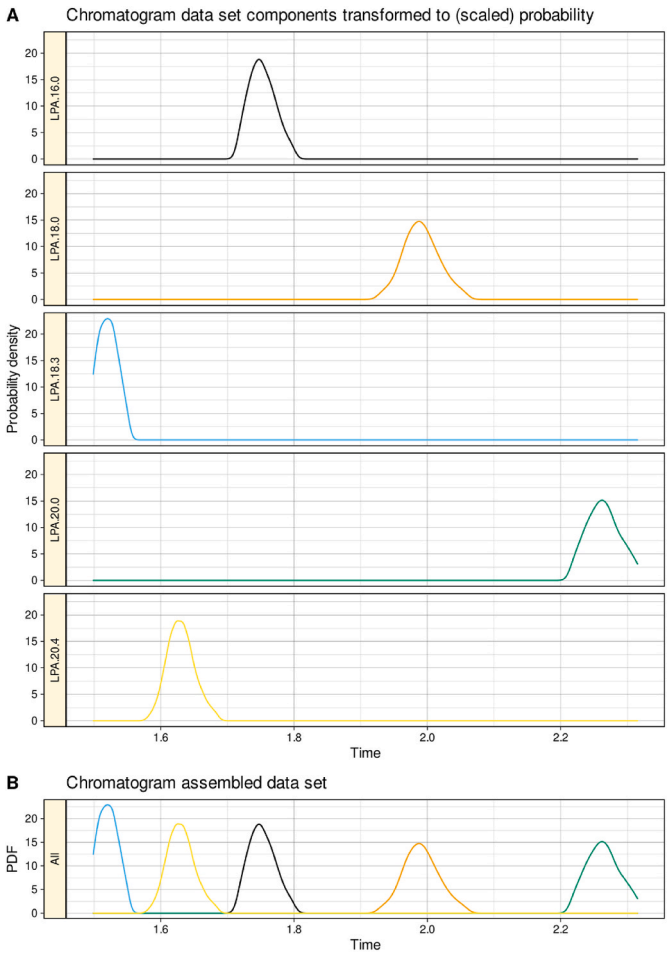


Fig. 5. Chromatogram data set of five different lysophosphatidic acids (LPA 16:0, 18:0, 18:3, 20:0, and 20:4). This substance mixture was prepared for calibration of laboratory assays in lipidomics. Hit information per lipid was available separately, which provided the modes information. A one-dimensional data set was generated from the time and peak height information by repeating each time point as many times as the peak height, weighted by a factor of 0.0001 to limit the size of the data set (n = 1166). The dataset is available as "Chromatogram" sample data in the above R library "opGMMassessment".

individual components hit a detector at a given time. For the present experiments, a chromatogram with five different lysophosphatidic acids (LPA 16:0, 18:0, 18:3, 20:0, and 20:4) was taken from the local in-house analytics laboratory. This substance mixture was prepared for calibration of laboratory assays in lipidomics. Hit information per lipid was available separately, which provided the modes information. A one-dimensional data set was generated (Fig. 4) from the time and peak height (counts per second) information by repeating each time point as many times as the peak height, weighted by a factor of 0.0001 to limit the size of the data set (n = 1166), and after setting a high-pass filter to peak height = 50,000. The dataset is available as "Chromatogram" sample data in the above R library "opGMMassessment". Correct GMM estimates in terms of mode number detection were obtained only with the combination of Markov chain Monte Carlo fitting with a likelihood ratio test proposed here and with the mclust:densityMclust, the latter providing slightly better parameter estimates (Fig. 4).

3.2.4. *One-dimensional three-modal data derived from iris flower measurements*

The widely known Iris flower data set [50,51] contains measurements in centimeters of four variables, sepal length and width or petal length and width, acquired from 50 flowers of each of three species, *Iris setosa*, *versicolor*, and *virginica*. The class information is mainly reflected in this multidimensional data space, where it was used to develop the method of linear discriminant analysis [50]. To obtain a one-dimensional three-class data set consistent with the species classification, independent component analysis (ICA) [52] was performed on the iris data set. The resulting first component, IC1, showed the desired three-modal distribution reflecting the flower species (Fig. 4). The M = 3 modes were recognized by all GMM algorithms except mixtools: normalmixEM.

4. Discussion

GMMs are often used to analyze structures in biomedical data, as evidenced by their increasing mentions in the PubMed database. That is, a search for "Gaussian mixture" at https://pubmed.ncbi.nlm.nih.gov on May 27, 2022, yielded 1757 hits, with the earliest article dating from 1983 [53]. Mentions in PubMed steadily increased over the past two decades. GMMs are commonly used for subgroup detection in biomedical data, including in one-dimensional variables, with subsequent subject assignment [54] or for classification of biological signals in the generation of diagnostic markers [55]. Many clinical diagnoses are based on cutoff values in one-dimensional variables. Examples are an 11-point numerical rating Scale (NRS) for pain intensity that triggers therapeutic interventions when NRS >4 [56], a 100-mm visual analogue scale for assessment of pain in rheumatoid arthritis where scores >40 mm indicate persistent pain as a clinically accepted cutoff [9], or a diabetes risk score [3] called "FINDRISK" [57], from which five categories of diabetes risk are derived, from "low risk" of 1% of developing diabetes in the next 10 years at values < 7 to, "slightly increased" of 4% diabetes risk at values [7, …,11], "moderate risk" of 17% risk of diabetes at scores [12, …,14], "high risk" of 33% risk of diabetes at scores [15, …,20], and "very high" risk of 50% of developing diabetes at scores >20. Many other similarly constructed scores that are either unidimensional in that they contain only one measurement or are scores composed of multiple measurements of different items that are eventually reduced to a single dimension.

GMM analyses often constitute only a small part of data analysis, addressing subgroup structure in simple signals such as the rating of a clinical symptom on a visual analog scale (for further example, see motivation section in this report), while the main interest is in subsequent analyses that relate more complex information to the identified subgroup structure. The correctness of the target of classifier tuning, i.e., the subgroup structure in a data set, is often taken for granted. This is facilitated by the apparent simplicity of a GMM. The validity of subsequent analyses, however, depends critically on the validity of the GMM result. For research environments that rely on pre-packaged software solutions without comparing approaches or tuning hyperparameters, as is often standard in clinical biomedical laboratories, it is critical to obtain a reliable estimate of the underlying GMM in an automated manner. The combination of methods proposed here most often yielded the correct number of mixtures, i.e., it is least likely to miss subgroups or identify incorrect subgroups in one-dimensional data. Since subgroup identification is often the preliminary unsupervised part of a data analysis where a class structure is created as a target for subsequent supervised analysis (e.g., Refs. [12,58]), reliability on this point is critical. A visual review of the results is strongly recommended and therefore implemented in the presented R package "opGMMassessment".

4.1. *Limitations*

The present experiments aimed at an automated out-of-the-box tool for GMM estimation in R. No new methods were developed for mode number detection in GMM parameter estimation, but the goal was to combine available methods based on extensive comparative testing. The results and implementations are limited to the implementations in the R

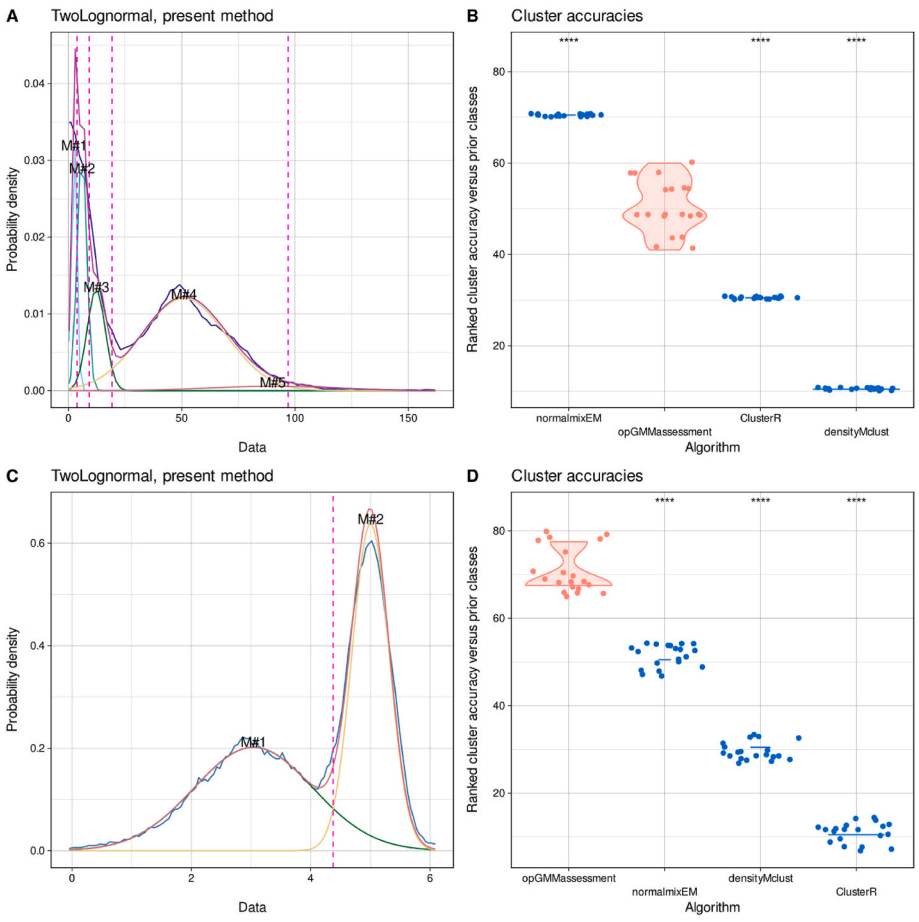


Fig. 6. Demonstration of the need for appropriate data transformation prior to GMM fitting. A one-dimensional log-normal distributed data set with n = 1000 points for each mode and means $m_{1,2} = [2,4]$ and standard deviations $s_{1,2} = [1, 0.3]$. This lognormal data could only be reliably fitted with correct detection of the M = 2 modes by the presently proposed method when it was log-transformed: **A and C:** Empirical distribution of the data estimated using the Pareto density estimation (PDE [63]; black lines), along with the GMM fits (red line) and the single Gaussians (differently colored lines). Bayesian boundaries between Gaussians are shown as magenta vertical dashed lines before and after log transformation (panel A and C, respectively). **B and D:** Ranks of classification accuracies against the prior classification are shown as individual dots, overlaid with a violin plot. Algorithms are presented in descending order of ranks achieved; higher ranks indicate better performance in assigning an instance to its original class. The statistical significance of differences to the presently proposed "opGMMassessment" method, assessed by means of Wilcoxon-Mann-Whitney U tests [64, 65], is indicated at the top line as stars: *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$. Automatic GMM fitting and subsequent class assignment of instances after calculating Bayesian boundaries from the obtained GMM parameter values was performed in 20 replications using different values of seed. The alternatives tested included the present proposal "opGMMassessment", "densityMclust" from the R package "mclust", "normalmixEM" from the R package "mixtools" and "GMM" from the R package "ClusterR". The default parameter settings of the respective R packages were used. The figure was created using the software package R (version 4.2.0 for Linux; https://CRAN.R-project.org/)[42] and the R package "ggplot2" (https://cran.r-project.org/package=ggplot2 [61]). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

programming language. Implementations of the methods used available in other popular data science programming languages may achieve different performances and final scores and must therefore be tested separately before implementing an analogous tool in another programming language.

It should also be noted that for empirical data with skewed distributions such as exponential or lognormal distributions, the present combination of methods and the associated implementation in the R programming language do not replace general data preprocessing requirements, such as data transformation or outlier removal. This is demonstrated on a simple artificial data set consisting of a mixture of two lognormal distributed modes with $m_{1,2} = [2,4]$, $s_{1,2} = [1, 0.3]$ and $w_{1,2} = [0.5, 0.5]$ (Fig. 6). Without appropriate transformation, the bimodal distribution was recognized by mixtools:normalmixEM but not by other fitting tools. After transformation, regardless of whether the Box-Cox transformation [59] was used or a log transformation was applied the presently proposed method gave the best results. The mixtools:normalmixEM algorithm showed a tendency to favor a smaller number of modes in the experiments in this report. Its correct result on this skewed data may be more due to this behavior than reflecting a true advantage.

Among the methods used to determine the number of mixtures in the GMM, representative members were selected for each major method family. However, this was not complete as underlined by the large number of methods included in the R library "NbClust". In addition, some of the methods mentioned in the theoretical part of the method description were not included because they either did not run on univariate data, which precluded their use for the present purpose, or the calculations were stopped after hours without any progress in the calculations as observed with the implementation of the Ameijeiras-Alonso [30] in the "multimode" R library despite that being the default of that package. However, the good ranking of the composite approach for determining the number of mixes provided in the R library "NbClust" underscores that majority voting is probably a reasonable approach here.

The comparatively more reliable GMM adaptation in the present software implementation came already at the cost of computational overhead. Although systematic benchmarking was not intended in the present experiments, the observation of RAM usage of up to 200 gigabytes during the experiments suggests a cost of parallel processing implementation. However, this was necessary considering that the three experiments in the simulation study each lasted 6 h when parallel computing on 60 cores. It should also be noted that the application of GMM for substructure determination assumes that the generative process of the underlying data set is based on sampling from a superposition of multiple normal distributions. If this prior assumption is not met by the data GMM-based analyses may deliver a false result. However, for RNAseq data, whose underlying generative process is most likely not a multimodal normal distribution, it has already been shown that GMM analysis can still reconstruct the true data structure to an acceptable extent [60].

5. Conclusions

An automated approach for fitting Gaussian mixtures to one-dimensional data was developed, with the goal of providing reliable information about the number of modes and GMM parameters "out of the box". After comparative analyses, the best-performing GMM fitting algorithms were combined with criteria for determining the mode number that provided the closest results to the known mode number. The performance of this combination was compared with common alternatives for "out of the box" assessment of GMM parameters. However, it should be noted that none of the methods always indicated the underlying class structure correctly; the presented method was nevertheless most consistently placed among the top-ranked approaches. The results of the assessments led to the technical contribution of an R package freely available at https://cran.r-project.org/package=opGMMassessment. It can be advised for fitting GMM in real-world data where a Gaussian distribution can be expected and the fitting is performed in an automated manner, as is often the standard in biomedical research environments.

Ethics approval and consent to participate

Not applicable. Data have been taken from publicly available sources.

Consent for publication

Not applicable.

Availability of data and material

The "opGMMassessment" R package is freely available at https://cran.r-project.org/package=opGMMassessment. It contains the chromatogram data set while other data sets used are freely available from the sources referenced in this report or their generates is described in this report.

Funding

This work has been funded by the Landesoffensive zur Entwicklung wissenschaftlich-ökonomischer Exzellenz (LOEWE), LOEWE-Zentrum für Translationale Medizin und Pharmakologie (JL), through the project "Reproducible cleaning of biomedical laboratory data using methods of visualization, error correction and transformation implemented as interactive R-notebooks" (JL). JL was supported by the Deutsche Forschungsgemeinschaft (DFG LO 612/16-1).

Author contributions

JL – Data analysis, conceptualization and implementation of the algorithms, software implementation, interpretation of the results, writing of the manuscript, creation of the figures, writing the supplementary information, revision of the manuscript.

SM – Writing of the manuscript, critical revision of the manuscript for important intellectual content, writing the supplementary information, testing of the software implementation, revision of the manuscript.

AU - critical revision of the manuscript for important intellectual content, writing of the manuscript, contributing to the selection of sample data sets.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Not applicable.

References

[1] Finnerup NB, Haroutounian S, Kamerman P, Baron R, Bennett DLH, Bouhassira D, et al. Neuropathic pain: an updated grading system for research and clinical practice. Pain 2016;157(8):1599–606.
[2] Beck AT, Ward CM, Mendelson M, Mock JE, Erbaugh JK. An inventory for measuring depression. Arch Gen Psychiatr 1961;4:561–71.
[3] Lindström J, Tuomilehto J. The diabetes risk score: a practical tool to predict type 2 diabetes risk. Diabetes Care 2003;26(3):725–31.
[4] Burckhardt CS, Clark SR, Bennett RM. The fibromyalgia impact questionnaire: development and validation. J Rheumatol 1991;18(5):728–33.
[5] Roth JA, Chrobak C, Schädelin S, Hug BL. MELD score as a predictor of mortality, length of hospital stay, and disease burden: a single-center retrospective study in 39,323 inpatients. Medicine (Baltim) 2017;96(24):e7155.
[6] Kobal G, Hummel T, Sekinger B, Barz S, Roscher S, Wolf SR. Sniffin' sticks": screening of olfactory performance. Rhinology 1996;34:222–6.
[7] Hummel T, Sekinger B, Wolf SR, Pauli E, Kobal G. 'Sniffin' sticks': olfactory performance assessed by the combined testing of odor identification, odor discrimination and olfactory threshold. Chem Senses 1997;22(1):39–52.
[8] Oleszkiewicz A, Schriever VA, Croy I, Hahner A, Hummel T. Updated Sniffin' Sticks normative data based on an extended sample of 9139 subjects. Eur Arch Oto-Rhino-Laryngol 2019;276(3):719–28.
[9] Tubach F, Ravaud P, Martin-Mola E, Awada H, Bellamy N, Bombardier C, et al. Minimum clinically important improvement and patient acceptable symptom state in pain and function in rheumatoid arthritis, ankylosing spondylitis, chronic back pain, hand osteoarthritis, and hip and knee osteoarthritis: results from a prospective multinational study. Arthritis Care Res 2012;64(11):1699–707.
[10] Uhlig T, Kvien TK, Glennas A, Smedstad LM, Forre O. The incidence and severity of rheumatoid arthritis, results from a county register in Oslo, Norway. J Rheumatol 1998;25(6):1078–84.
[11] Fischer H. A history of the central limit theorem: from classical to modern probability theory. In: Fischer H, editor. New York, NY: Springer New York; 2011. p. 1–16. 2011//.
[12] Lotsch J, Alfredsson L, Lampa J. Machine-learning based knowledge discovery in rheumatoid arthritis related registry data to identify predictors of persistent pain. Pain; 2019.
[13] Bayes M, Price M. An essay towards solving a problem in the doctrine of chances. By the late rev. Mr. Bayes, F. R. S. Communicated by mr. Price, in a letter to john canton, A. M. F. R. S. Phil Trans 1763;53:370–418.
[14] Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a data set via the gap statistic. J Roy Stat Soc B 2001;63(2):411–23.
[15] Ultsch A, Thrun MC, Hansen-Goos O, Lötsch J. Identification of molecular fingerprints in human heat pain thresholds by use of an interactive mixture model R toolbox (AdaptGauss). Int J Mol Sci 2015;16(10):25897–911.
[16] Ihaka R, Gentleman RR. A language for data analysis and graphics. J Comput Graph Stat 1996;5(3):299–314.
[17] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J Roy Stat Soc B 1977;39(1):1–38.
[18] Frühwirth-Schnatter S. Finite mixture and Markov switching models. Berlin: Springer; 2006.
[19] Mouselimis L. ClusterR: Gaussian mixture models, K-means, mini-batch-kmeans, K-medoids and affinity propagation clustering. 2020.
[20] Scrucca L, Fop M, Murphy TB, Raftery AE. Mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. Rice J 2016;8(1):205–33.
[21] Benaglia T, Chauveau D, Hunter DR, Young DS. Mixtools: an R package for analyzing mixture models. J Stat Software 2010;1(Issue 6). 2009.
[22] Ashlock D. Evolutionary computation for modeling and optimization. Springer Science & Business Media; 2006.
[23] Lerch F, Ultsch A, Lotsch J. Distribution Optimization: an evolutionary algorithm to separate Gaussian mixtures. Sci Rep 2020;10(1):648.
[24] Gilks WR, Richardson S, Spiegelhalter D. Markov chain Monte Carlo in practice. Taylor & Francis; 1995.
[25] Peters G. Markov chain Monte Carlo: stochastic simulation for Bayesian inference (second ed.). Dani gamerman and hedibert F. Lopes, chapman & Hall/CRC, boca raton, FL, 2006. No. of pages: xvii +323. Price: $69.95. ISBN10: 1-58488-587-4, ISBN13: 978-1-58488-587-0. Statistics in Medicine. 2008;vol. 27(16):3213-3214.
[26] Harrison RL. Introduction to Monte Carlo simulation. AIP Conf Proc 2010;1204(1):17–21.
[27] Eddy SR. What is a hidden Markov model? Nat Biotechnol 2004;22(10):1315–6.
[28] van Ravenzwaaij D, Cassey P, Brown SD. A simple introduction to Markov chain Monte–Carlo sampling. Psychonomic Bull Rev 2018;25(1):143–54.
[29] Komárek A, Komárková L. Capabilities of R Package mixAK for clustering based on multivariate continuous and discrete longitudinal data. J Stat Software 2014;1 (Issue 12). 2014.
[30] Ameijeiras-Alonso J, Crujeiras RM, Rodríguez-Casal A. Mode testing, critical bandwidth and excess mass. Test 2019;28(3):900–19.
[31] Lötsch J, Malkusch S, Ultsch A. Optimal distribution-preserving downsampling of large biomedical data sets (opdisDownsampling). PLoS One 2021;16(8):e0255838.
[32] Akaike H. A new look at the statistical model identification. IEEE Trans Automat Control 1974;19(6):716–23.

[33] Silverman BW. Using kernel density estimates to investigate multimodality. J Roy Stat Soc B 1981;43(1):97–9.
[34] Ameijeiras-Alonso J, Crujeiras RM, Rodríguez-Casal A. Multimode: an R package for mode assessment. arXiv preprint arXiv: 180300472. 2018.
[35] Müller DW, Sawitzki G. Excess mass estimates and tests for multimodality. J Am Stat Assoc 1991;86(415):738–46.
[36] Polonik W. Measuring mass concentrations and estimating density contour clusters-an excess mass approach. Ann Stat 1995;23(3):855–81.
[37] Hartigan JA, Hartigan PM. The dip test of unimodality. Ann Stat 1985;13:70–84.
[38] Fisher NI, Marron JS. Mode Testing via the Excess Mass Estimate 2001;88(2):499–517.
[39] Cheng M-Y, Hall P. Calibrating the excess mass and dip tests of modality. J Roy Stat Soc B 1998;60(3):579–89.
[40] Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K. Cluster: cluster Analysis basics and extensions. 2017.
[41] Charrad M, Ghazzali N, Boiteau V, Niknafs A. NbClust: an R package for determining the relevant number of clusters in a data set. J. Stat. Software. Artic. 2014;61(6):1–36.
[42] R Development Core Team. R. A language and environment for statistical computing. 2008.
[43] Lötsch J, Ultsch A. Current projection methods-induced biases at subgroup detection for machine-learning based data-analysis of biomedical data. Int J Mol Sci 2019;21(1).
[44] Ultsch A, Lötsch J. Machine-learned cluster identification in high-dimensional data. J Biomed Inf 2017;66:95–104.
[45] Smirnov N. Table for Estimating the Goodness of Fit of Empirical Distributions 1948;(2):279–81.
[46] Weston S. doParallel: foreach parallel adaptor for the 'parallel' package. Microsoft Corporation; 2020.
[47] Weston S. Foreach: provides foreach looping construct. Microsoft Corporation; 2020.
[48] Ivezić Ze. Statistics, data mining, and machine learning in astronomy : a practical Python guide for the analysis of survey data. Princeton U.P.; 2014.
[49] Attia A, Moosavi A, Sandu A. Cluster sampling filters for non-Gaussian data assimilation. Atmosphere 2018;9(6).
[50] Fisher RA. The use of multiple measurements in taxonomic problems. Ann. Eugen. 1936;7(2):179–88.
[51] Anderson E. The irises of the Gaspé peninsula. Bull. Am. Iris. Soc. 1935;59:2–5.
[52] Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. Neural Network 2000;13(4):411–30.
[53] Fukunaga K, Flick TE. Estimation of the parameters of a Gaussian mixture using the method of moments. IEEE Trans Pattern Anal Mach Intell 1983;5(4):410–6.
[54] Heidegger T, Hansen-Goos O, Batlaeva O, Ziemann U, Lötsch J. A data-driven approach to responder subgroup identification after paired continuous theta burst stimulation. Front Hum Neurosci 2017;4(11):382.
[55] Costa T, Boccignone G, Ferraro M. Gaussian mixture model of heart rate variability. PLoS One 2012;7(5):e37731.
[56] Wolfe F, Michaud K. Assessment of pain in rheumatoid arthritis: minimal clinically significant difference, predictors, and the effect of anti-tumor necrosis factor therapy. J Rheumatol 2007;34(8):1674–83.
[57] Schwarz P. FINDRISK – test für Diabetesrisiko. Diabetologe 2020;16(5):524–6.
[58] Lötsch J, Geisslinger G, Heinemann S, Lerch F, Oertel BG, Ultsch A. Quantitative sensory testing response patterns to capsaicin- and UV-B-induced local skin hypersensitization in healthy subjects: a machine-learned analysis. Pain 2017;159(1):11–24.
[59] Box GE, Cox DR. An analysis of transformations. Journal of the Royal Statistical Society Series B (Methodological); 1964. p. 211–52.