McCaw *et al. BMC Bioinformatics* 2022, **23**(1):208
https://doi.org/10.1186/s12859-022-04740-9

BMC Bioinformatics

RESEARCH Open Access

Fitting Gaussian mixture models on incomplete data

Zachary R. McCaw[1*], Hugues Aschard[2] and Hanna Julienne[2]

*Correspondence:
zmccaw@alumni.harvard.edu

[1] School of Public Health, Harvard T.H. Chan, 677 Huntington Ave, Boston, MA 02115, USA
[2] Department of Computational Biology, Institut Pasteur, Université de Paris, 25-28 Rue du Dr Roux, 75015 Paris, France

Abstract

Background: Bioinformatics investigators often gain insights by combining information across multiple and disparate data sets. Merging data from multiple sources frequently results in data sets that are incomplete or contain missing values. Although missing data are ubiquitous, existing implementations of Gaussian mixture models (GMMs) either cannot accommodate missing data, or do so by imposing simplifying assumptions that limit the applicability of the model. In the presence of missing data, a standard *ad hoc* practice is to perform complete case analysis or imputation prior to model fitting. Both approaches have serious drawbacks, potentially resulting in biased and unstable parameter estimates.

Results: Here we present missingness-aware Gaussian mixture models (`MGMM`), an `R` package for fitting GMMs in the presence of missing data. Unlike existing GMM implementations that can accommodate missing data, `MGMM` places no restrictions on the form of the covariance matrix. Using three case studies on real and simulated *'omics* data sets, we demonstrate that, when the underlying data distribution is near-to a GMM, `MGMM` is more effective at recovering the true cluster assignments than either the existing GMM implementations that accommodate missing data, or fitting a standard GMM after state of the art imputation. Moreover, `MGMM` provides an accurate assessment of cluster assignment uncertainty, even when the generative distribution is not a GMM.

Conclusion: Compared to state-of-the-art competitors, `MGMM` demonstrates a better ability to recover the true cluster assignments for a wide variety of data sets and a large range of missingness rates. `MGMM` provides the bioinformatics community with a powerful, easy-to-use, and statistically sound tool for performing clustering and density estimation in the presence of missing data. `MGMM` is publicly available as an `R` package on CRAN: https://CRAN.R-project.org/package=MGMM.

Keywords: Clustering, Missing data, Gaussian mixture models

Background

Gaussian mixture models (GMMs) provide a flexible approach to multivariate density estimation and probabilistic clustering [1]. Most implementations of GMMs in the `R` programming language, including `mclust` [2] and `mixtools` [3], require complete data. The few implementations that do allow for missing values, such as `MixAll` [4],

have limited applicability due to their restrictive simplifying assumptions. For example, `MixAll` assumes diagonal covariance matrices, which implies that the elements of the Gaussian vectors under consideration are mutually independent. In practice, both correlated and missing data are common. Our work was motivated by the problem of clustering summary statistics arising from genome-wide association studies (GWAS) of multiple correlated traits [5]. Missing data arose because not every genetic variant was tested for association with every trait.

Although commonly applied, standard approaches for addressing missing data prior to clustering, including complete case analysis and imputation, have serious drawbacks. By discarding information from observations that are only partially observed, complete case analysis makes inefficient use of the data. This leads to unstable estimates of model parameters and cluster assignments that are susceptible to significant changes if the missingness pattern of the input data changes slightly. On the other hand, mean or median imputation introduces bias by making the incomplete observations appear less variable, and by shrinking the incomplete observations towards the complete data. This can result in inaccurate posterior membership probabilities that place excess weight on clusters with less missing data. Although a method has been described for estimating GMMs from incomplete data [6], there are no existing implementations in `R`.

To fill this gap, we present `MGMM` [7], a computationally efficient `R` package for maximum likelihood estimation of GMMs in the presence of missing data. Our package is carefully implemented and documented for ease of use. In contrast to complete case analysis, our approach makes full use of the available data; and in contrast to clustering after imputation, our approach is unbiased for estimating the parameters of the generative GMM, accurately assesses the posterior membership probabilities, and correctly propagates estimation uncertainty. Moreover, our implementation places no restrictions on the model's covariance structures.

`MGMM` employs an expectation conditional maximization (ECM) algorithm [8], which accelerates estimation by breaking direct maximization of the EM objective function into a sequence of simpler conditional maximizations, each of which is available in closed form. While EM algorithms are regularly used for estimating GMMs, for example by both `mclust` and `mixtools`, those implementations only address missingness of the true cluster assignments, and not missingness of elements from the input vectors. In contrast, our ECM algorithm handles both missingness of the cluster assignments and of elements from the input data. We present a comprehensive benchmark, including three case studies, demonstrating that when the underlying distribution is well-approximated by a GMM, `MGMM` is better able to recover the true cluster assignments than `MixAll` or than standard GMM applied after state of the art imputation (e.g. multiple imputation by chained equations, MICE [9]). While we prioritized cluster assignments accuracy, our implementation also proves competitive in regard to running time for the missingness rates usually encountered in real data.

Methods

Model

This section provides an overview of the statistical model. For a detailed derivation and description of the ECM algorithm, see the Supporting Information.

Statistical model overview

Consider n independent vectors $\boldsymbol{y}_i = \text{vec}(Y_{i1}, \ldots, Y_{id})$ in $\mathbb{R}^d$, each arising from one of k distinct clusters. Although k is assumed known throughout this work, see "Methods" section of the Supporting Information for an approach to choosing k. Let $Z_{ij} = 1$ if the ith observation belongs to cluster j, and define the $k \times 1$ indicator vector $\boldsymbol{z}_i = \text{vec}(Z_{i1}, \ldots, Z_{ik})$. Conditional on membership to the jth cluster, $\boldsymbol{y}_i$ follows a multivariate normal distribution, with cluster-specific mean $\boldsymbol{\mu}_j$ and covariance $\boldsymbol{\Sigma}_j$. Let π_j denote the marginal probability of membership to the jth cluster. The observations can be viewed as arising from the following hierarchical model:

$$\begin{aligned} \boldsymbol{z}_i &\sim \text{Multinomial}(1, \boldsymbol{\pi}), \\ \boldsymbol{y}_i \big| (Z_{ij} = 1) &\sim N\left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right). \end{aligned} \tag{1}$$

Marginalized over the latent cluster assignment vector $\boldsymbol{z}_i$, each observation $\boldsymbol{y}_i$ follows a k component Gaussian mixture model (GMM):

$$f(\boldsymbol{y}_i) = \sum_{j=1}^{k} \pi_j f\left(\boldsymbol{y}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right). \tag{2}$$

To perform estimation in the presence of missingness, we derive the EM objective function $Q(\boldsymbol{\pi}, \boldsymbol{\theta} | \boldsymbol{\pi}^{(r)}, \boldsymbol{\theta}^{(r)})$, which is the expectation of the complete data log likelihood, given the observed data and current parameter estimates (see the Supporting Information for complete derivation). The EM objective is optimized using a sequence of three conditional maximizations. Let $\hat{\gamma}_{ij}^{(r)}$ denote the *responsibility* of the jth cluster for the ith observation, which is the current conditional probability of membership to that cluster, given the observed data. In the first step, the cluster means are updated using the responsibility-weighted average of the working outcome vectors $\hat{\boldsymbol{y}}_{ij}^{(r)}$. In the next step, the cluster covariances are updated using the responsibility-weighted average of the working residual outer products. In the final step, the cluster responsibilities and marginal membership probabilities are updated using the new means and covariances. This process iterates until the improvement in the EM objective drops below the specified tolerance. Unbiased estimation of the model parameters requires that missingnesss in the outcome vector occur at random [10]. This means that whether a particular element of the outcome vector is missing can depend on the values of those elements that are observed, but not upon the values of those elements that are missing. See section 1.3 of the Supporting Information for further discussion of the missing at random assumption.

Imputation

Having fit the GMM in (2) via maximum likelihood, the missing values $\boldsymbol{y}_i^{\text{miss}}$ of each observation $\boldsymbol{y}_i$ may subsequently be imputed. Note that, in contrast to the imputation *before* estimation procedure commonly used to address missing input data, `MGMM` performs imputation only *after* estimation. In this way, imputation has no effect on the final maximum likelihood estimates $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})$. To perform a deterministic single imputation, as is done by the `FitGMM` function, $\hat{\boldsymbol{y}}_i^{\text{miss}}$ may be set to its posterior expectation given $\boldsymbol{y}_i^{\text{obs}}$:

$$\begin{aligned}\hat{\boldsymbol{y}}_i^{\text{miss}} &\equiv \mathbb{E}\left(\boldsymbol{y}_i^{\text{miss}} | \boldsymbol{y}_i^{\text{obs}}; \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}\right) \\ &= \mathbb{E}\left\{\mathbb{E}\left(\boldsymbol{y}_i^{\text{miss}} | z_{ij} = 1, \boldsymbol{y}_i^{\text{obs}}; \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}\right)\right\} \\ &= \sum_{j=1}^{k} \hat{\boldsymbol{y}}_{ij}^{\text{miss}} \hat{\gamma}_{ij}.\end{aligned} \tag{3}$$

Here $\hat{\gamma}_{ij}$ is the final responsibility of cluster j for observation i, and:

$$\hat{\boldsymbol{y}}_{ij}^{\text{miss}} = \hat{\boldsymbol{\mu}}_{\text{miss},j} + \hat{\boldsymbol{\Sigma}}_{\text{miss,obs},j} \hat{\boldsymbol{\Sigma}}_{\text{obs},j}^{-1} \left(\boldsymbol{y}_i^{\text{obs}} - \hat{\boldsymbol{\mu}}_{\text{obs},j}\right).$$

While single imputation to the posterior expectation is useful for visualization, drawing inferences from singly-imputed data is generally invalid [10]. For subsequent inference, a multiple imputation procedure is necessary, wherein multiple stochastic imputations of the input data are generated, analyzed in parallel, and the resulting estimates combined. To generate a single stochastic imputation of $\boldsymbol{y}_i^{\text{miss}}$, as is done by the `GenImputation` function, the latent cluster membership $\boldsymbol{z}_i$ is first drawn from a multinomial distribution over $\hat{\boldsymbol{\gamma}}_i = \mathbb{P}(\boldsymbol{z}_i | \boldsymbol{y}_i^{\text{obs}}; \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}})$:

$$\boldsymbol{z}_i \sim \text{Multinomial}(1, \hat{\boldsymbol{\gamma}}_i).$$

Given the cluster assignment, $Z_{ij} = 1$, the missing elements $\boldsymbol{y}_i^{\text{miss}}$ are drawn from a normal distribution with mean:

$$\mathbb{E}\left(\boldsymbol{y}_i^{\text{miss}} | \boldsymbol{y}_i^{\text{obs}}, Z_{ij} = 1; \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}\right) = \hat{\boldsymbol{\mu}}_j^{\text{miss}} + \hat{\boldsymbol{\Sigma}}_{\text{miss,obs},j} \hat{\boldsymbol{\Sigma}}_{\text{obs,obs},j} \left(\boldsymbol{y}_i^{\text{obs}} - \hat{\boldsymbol{\mu}}_{\text{obs},j}\right)$$

and covariance:

$$\mathbb{V}\left(\boldsymbol{y}_i^{\text{miss}} | \boldsymbol{y}_i^{\text{obs}}, Z_{ij} = 1; \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}\right) = \hat{\boldsymbol{\Sigma}}_{\text{miss,miss},j} - \hat{\boldsymbol{\Sigma}}_{\text{miss,obs},j} \hat{\boldsymbol{\Sigma}}_{\text{obs,obs},j}^{-1} \hat{\boldsymbol{\Sigma}}_{\text{obs,miss},j}$$

An exposition of how to use multiple stochastic imputations for inference is presented in the Supporting Information.

Benchmarking method

All analyses were performed in R 3.5.0 `R` [11]. We designed a benchmarking procedure to compare the performance of `MGMM` against imputation followed by standard GMM (also implemented by `MGMM`) and another package that allows for missing values (`MixAll`). The imputation methods included in the benchmark were: naive mean and median imputation; k-nearest neighbors imputation, as implemented by the `VIM` package [12]; multiple imputation by chained equations, as implemented by the `MICE` package [9]; and random forest imputation, as implemented by the `missforest` package [13]. We defined clustering performance as the capacity of the algorithm to recover the true cluster assignments when applied to example data sets. We assessed the quality of the clustering by calculating the adjusted rand index (ARI) between the recovered and true class assignments. The running time was defined as the time necessary to obtain cluster assignation starting with the data set with missing values. We applied the benchmarking procedure to four case studies: a simulated four

component mixture of bivariate Gaussians, a cancer patient RNA-seq data set, simulated genome-wide association studies (GWAS) summary statistics, and summary statistics from GWAS for cardiovascular disease risk factors [14].

Missingness

For n observations on d dimensional data, a fraction of missing values m was introduced completely at random by setting $\lceil (m \times n \times d) \rceil$ elements of the data set to `NA`.

Evaluation metric

The quality of clustering was evaluated using the ARI [15, 16]. Briefly, the Rand Index (RI) is a measure of similarity that assesses the agreement between two partitions of a collection of n objects. All possible pairs of objects are examined, and the proportion of pairs that are either 1. in the same cluster or 2. in different clusters according to both partitions is calculated. ARI is a variation of the RI that is adjusted for chance, and is permutation invariant. A value near zero suggests the agreement between the two partitions is no better than expected by chance, while a value of one occurs when the two partitions are identical. We define the quality of clustering as the ARI between the reference or true clustering, established in the data set description, and the clustering performed in the presence of missingness.

Benchmarking procedure

We designed the benchmarking procedure outlined in Fig. 1 and described in Algorithm 1 to compare the performance of `MGMM` with imputation followed by standard GMM.

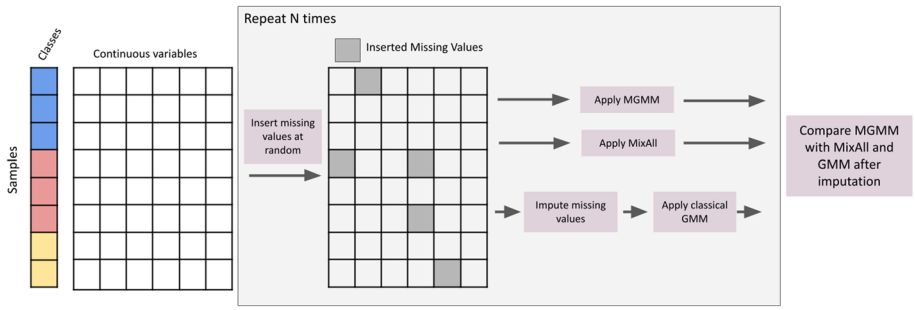

Fig. 1 Benchmark procedure schematic. The input data are continuous vectors with known class assignments. Missing values are introduced completely at random. GMMs were then fit to the incomplete data in several ways: 1. by using `MGMM`, which allows for missing values and arbitrary covariance structures; 2. by using `MixAll`, which allows for missing values but assumes a diagonal covariance structure; 3. by imputing the missing values, then fitting a standard GMM. The GMMs were evaluated based on the adjusted Rand index between the predicted and true cluster assignments. This procedure was repeated $N =$ times

Algorithm 1 Benchmarking Procedure for `MGMM`

Require: Complete data $\mathcal{D}$ with known cluster assignments.

for m in `missing_rate_range` **do**

1 Randomly introduce `NAs` to form the *incomplete data* $\mathcal{D}_m$.

2 Apply `MGMM` to $\mathcal{D}_m$, and compute the ARI.

3 Apply `MixAll` to $\mathcal{D}_m$, and compute the ARI.

for j in `imputation_methods` **do**

1 Create $\mathcal{D}_j$ by completing $\mathcal{D}_m$ using imputation method j.

2 Apply standard GMM to $\mathcal{D}_j$ and compute the ARI.

end for

end for

Compare `MGMM` with `MixAll` and GMM after imputation as a function of the missingness rate.

Benchmark data sets

Simulated Gaussian mixture

For the first clustering task, we consider data that were truly generated from a GMM, which is the setting in which `MGMM` should perform optimally. Data were simulated according to the hierarchical model in Eq. (1). The dimensionality d was set to 2 and the number of cluster components k to 4. The marginal density of the data generating process was:

$$\begin{pmatrix} Y_{i1} \\ Y_{i2} \end{pmatrix} \sim \sum_{j=1}^{4} \pi_j N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j).$$

The means ($\boldsymbol{\mu}_j$) were drawn from a uniform distribution on the square:

$$\{(x, y) : -5 \leq x \leq 5, -5 \leq y \leq 5\}.$$

The component variances were set to 0.9:

$$\Sigma_{11,j} = \mathbb{V}(Y_{i1}|Z_{ij} = 1) = \Sigma_{22,j} = \mathbb{V}(Y_{i2}|Z_{ij} = 1) = 0.9.$$

The covariance was uniformly sampled from the interval $(-0.9, 0.9)$:

$$\Sigma_{12,j} = \mathbb{C}(Y_{i1}, Y_{i2}|Z_{ij} = 1) \sim U(-0.9, 0.9).$$

Marginal membership to each cluster was equally likely, $\pi_j = 0.25$ for $j \in \{1, \ldots, 4\}$. A sample of size $n = 2000$ was generated using the `rGMM` function from `MGMM`. The true (generative) component memberships ($\boldsymbol{z}_i$) were used as the reference when evaluating clustering performance on incomplete data.

RNA sequence data from cancer patients

For the second clustering task, cancer gene expression data [17] were retrieved from the University of California Irvine machine learning repository. These data consist of expression values for 20,531 genes from $n = 801$ patients having 1 of $k = 5$ tumor types. Marginal analysis of variance was performed to identify the 20 most significantly differentially expressed genes (DEGs) across tumor types. The patient by DEG

matrix was then decomposed via principal components analysis (PCA), and the final cluster task was performed on the $d = 5$ leading principal components. Y_{il} represents the expression of patient i along the lth principal component. The patient's observed tumor type was used as the reference when evaluating clustering performance on incomplete data. The tumor types are abbreviated as follows:

- BRCA: Breast carcinoma.
- COAD: Colon adenocarcinoma.
- KIRC: Kidney renal clear-cell carcinoma.
- PRAD: Prostate adenocarcinoma.
- LUAD: Lung adenocarcinoma.

GWAS summary statistics

For the third clustering task, we consider summary statistics, both simulated and real, arising from GWAS for cardiovascular disease risk factors. In this setting, i indexes single nucleotide polymorphisms (SNPs) and Y_{il} is the standardized score (i.e. Z-score) quantifying the magnitude of the observed association between SNP i and phenotype l. The SNPs may belong to one of k clusters, where the Z-scores of SNPs within a cluster may exhibit correlations due to the combination of environmentally-induced correlation of the traits and sample overlap between the GWAS in which the Z-scores were ascertained.

A simulated set of GWAS summary statistics was generated for $d = 3$ traits and 900 SNPs arising from 1 of $k = 3$ clusters. The marginal density was:

$$\begin{pmatrix} Y_{i1} \\ Y_{i2} \\ Y_{i3} \end{pmatrix} \sim \sum_{j=1}^{3} \pi_j N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j).$$

The mean vectors ($\boldsymbol{\mu}_j$) were set to zero, and the cluster covariances ($\boldsymbol{\Sigma}_j$) were set to:

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 2.5 & 2 & 0 \\ 2 & 2.5 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$$
$$\boldsymbol{\Sigma}_2 = \begin{pmatrix} 2.25 & -2 & 0 \\ -2 & 2.25 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$$
$$\boldsymbol{\Sigma}_3 = \begin{pmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 4.5 \end{pmatrix}.$$

Marginal membership to each cluster was equally likely $\pi_j = 0.33$ for $j \in \{1, \ldots, 3\}$. These covariance structures were chosen to represent a variety of situations: (1) pleiotropic SNPs whose effects are positively correlated for the two first traits ($\boldsymbol{\Sigma}_1$); (2) pleiotropic SNPs whose effects are negatively correlated for the two first traits SNPs ($\boldsymbol{\Sigma}_2$); (3) SNPs acting predominantly on the third traits ($\boldsymbol{\Sigma}_3$). A sample of size $n = 900$ was generated using the `rGMM` function from the `MGMM` package. To emulate the omission of non-significant results, which frequently occurs when reporting GWAS summary

statistics, the SNPs were filtered to those having evidence of association with the traits at $p \leq 0.05$ via the omnibus test (9) detailed in the appendix. After filtering, $n = 183$ SNPs remained, with marginal cluster frequencies: $\pi_1 = 0.25$, $\pi_2 = 0.42$, $\pi_3 = 0.33$. The topology of the resulting data set is presented in Fig. 2. The true (generative) component memberships ($\boldsymbol{z}_i$) were used as the reference when evaluating clustering performance on incomplete data.

A set of real GWAS summary statistics for cardiovascular disease risk factors was prepared as described in [14]. These traits were: body mass index (BMI), coronary artery disease (CAD), low density lipoprotein (LDL), triglycerides (TG), waist to hip ratio (WHR), and any strokes (AS). From this collection of traits, we formed three example data sets. The first included {BMI, CAD, LDL} only; the second included {LDL, TG, BMI, AS, CAD}; the third {LDL, TG, BMI, AS, WHR}. We selected independent SNPs that were genome-wide significant (p-value $\leq 10^{-8}$) either marginally or via the omnibus test (9). These data sets contained 165, 166 and 179 SNPs respectively. For each example, a GMM with $k = 3$ components was fit to the complete data (using `FitGMM` from the `MGMM` package), and the cluster assignments from this initial model were used as the reference when evaluating clustering performance on incomplete data. Because the reference clustering partition was directly derived from the data, the benchmark on these examples assess the robustness of the clustering rather than the ability to recover a true, underlying data class assignment.

Imputation methods and `MixAll` parameter settings

Naive mean or median imputation refers to simply setting a missing value to the mean or median of the observed values along that coordinate. For kNN, a missing value was imputed to the (Euclidean) distance-weighted average of the 5 nearest observations with observed data along that coordinate. For MICE, a missing value was imputed to its conditional expectation given the observed coordinates via the method of predictive mean matching; the number of imputations was 10,

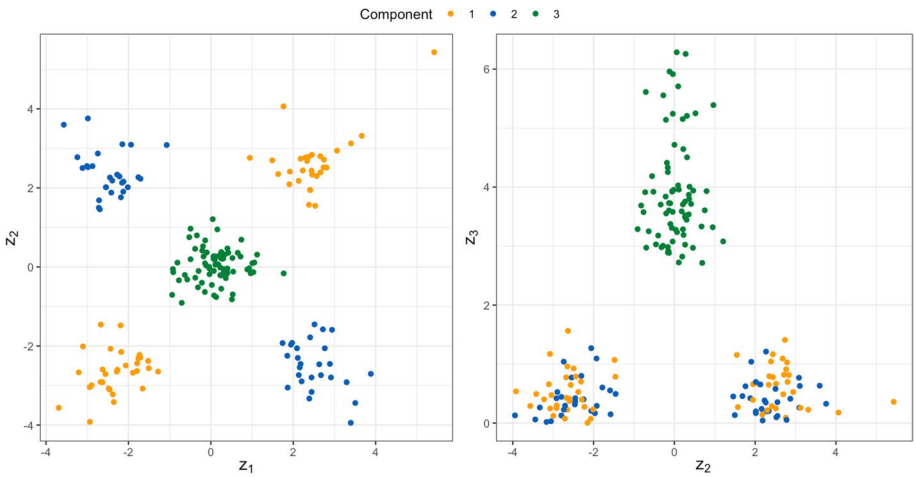

Fig. 2 Scatter Plot of the Simulated GWAS-like Multivariate Z-scores. Observations are colored according to component membership. The left panel plots the second coordinate against the first, and the right panel plots the third coordinate against the second

and the maximum number of Gibbs sampling iterations was 50. For random forest imputation, the number of trees per forest was 100, and the maximum number of refinement iterations was 10. For `MixAll`, the default parameters of the `clusterDiagGaussian` function were adopted; this resulted in all available models being fit, and the best model, according to the integrated completed likelihood criterion, being returned.

Filtering unassignable observations from MGMM and MICE

Since both `MGMM` and MICE provide an indication of the uncertainty in the cluster assignments, we created an additional clustering method in which observations with high assignment uncertainty were regarded as unassignable. This occurs when an observation could very plausibly has originated from more than one of the clusters, and may be exacerbated by excess missing data along a coordinate that helps to differentiate among clusters. This uncertainty can be assessed via the entropy of the posterior membership probabilities:

$$H(\boldsymbol{y}_i) = \frac{1}{\ln(k)} \sum_{j=1}^{k} \hat{\gamma}_{ij}(-1) \ln(\hat{\gamma}_{ij}), \tag{4}$$

where $\hat{\gamma}_{ij}$ is the final responsibility of cluster j for observation i.

For `MGMM`, the entropy of the posterior cluster responsibilities is calculated by `FitGMM` using (4). For MICE, each input data set is multiply imputed, and each of these imputed data set results in one *maximum a posteriori* cluster assignment. The posterior probability of membership to each cluster (i.e. the responsibilities, $\hat{\gamma}_{ij}$) may be approximated by the proportion of imputations on which an observation was assigned to each cluster.

In the filtered versions of `MGMM` and MICE, observations with high assignment uncertainty are identified via entropy and removed from consideration. For a given data set, such as the Cancer RNA-Seq data set, the distribution of entropy for MICE was typically right skewed (Supplementary Fig. 3-A). Consequently, for a fixed entropy threshold, the fraction of observations deemed unassignable is systematically higher for MICE than for `MGMM` (see Supplementary Fig. 3-B). To conduct a fair comparison of the two methods, we proceeded as follows:

1. For MICE, filter out observations with entropy exceeding 0.2 and assess performance on the remaining data.
2. Find the proportion of observations discarded by MICE.
3. Set an entropy threshold for `MGMM` such that the same proportion of observations is excluded as was removed for MICE.
4. Filter out observations with entropy exceeding the `MGMM` threshold and assess performance on the remaining data.

This procedure provides a fair comparison of `MGMM`-filtered and MICE-filtered by adaptively selecting the entropy threshold for `MGMM` in such a way that both methods remove the same fraction observations with high assignment uncertainty.

Benchmark results

Four component mixture of bivariate Gaussians

When the underlying distribution was in fact a GMM, `MGMM` uniformly dominated imputation plus GMM (Fig. 3) at recovering the true cluster assignments. GMM after imputation by kNN and GMM after MICE performed similarly. The performance of `MixAll` was relatively poor compared to `MGMM`, despite the data having truly been generated from a GMM. This underscores the disadvantages of an estimation procedure that incorrectly assumes a diagonal covariance structure. Interestingly, although non-parametric, random forest imputation was not competitive when the true data generating process was a GMM. Naive mean and median imputation strongly under-performed, and at elevated missingness created singularities in the data set that prevented the GMM from converging.

RNA sequence data from cancer patients

For the Cancer RNA-Seq data set, where the true generative model is unlikely to be a GMM, `MGMM` remained highly effective at recovering the true tumor type of the patient (see Fig. 4). `MixAll` evinced the worst performance when the proportion of missing data was $\leq 15\%$, however its performance deteriorated more slowly than the other methods, allowing it to become competitive when the proportion of missing data surpassed 35%. This may be because the simpler model assumed by `MixAll` is easier to fit when the data set is small and the proportion of missing values high. Random forests and kNN + GMM were competitive with `MGMM`, and outperformed when the proportion of missing data was $\geq 35\%$. Mean and median imputation were again not competitive, particularly when the proportion of missing data was $\geq 20\%$. MICE performed only slightly better than mean and median imputations. Linear imputation method may be ill-suited for separating the BRCA, LUDA, and PRAD tumor types.

GWAS summary statistics

Finally, we considered clustering vectors of GWAS summary statistics arising when the same SNPs are tested for association with multiple traits. This analysis is of interest for identifying pleiotropy, individual SNPs that have effects on multiple traits, and polygenicity, collections of multiple SNPs that have effects on common traits. Such analyses are often performed by combining data from multiple independent studies, and missingness arises because not all SNPs or all traits were ascertained in all studies. Further, this analysis would generally only include SNPs that were significantly associated with at least one trait.

Here we discuss one simulated and one real data example; two additional real data examples are presented in the appendix. For the simulated summary statistics in Fig. 5, the clustering task is same as the one presented on Fig. 3. The three clusters are clearly separated. Yet, the task remains challenging due to the specific and unusual cluster topology arising from GWAS data. Since the underlying distribution was in fact a GMM, `MGMM` again performs very well, only falling off when the proportion of missing data reaches $\geq 40\%$. In this example, `MixAll` performed was competitive with `MGMM`, although `MGMM` did outperform until the proportion of missing data become high. For

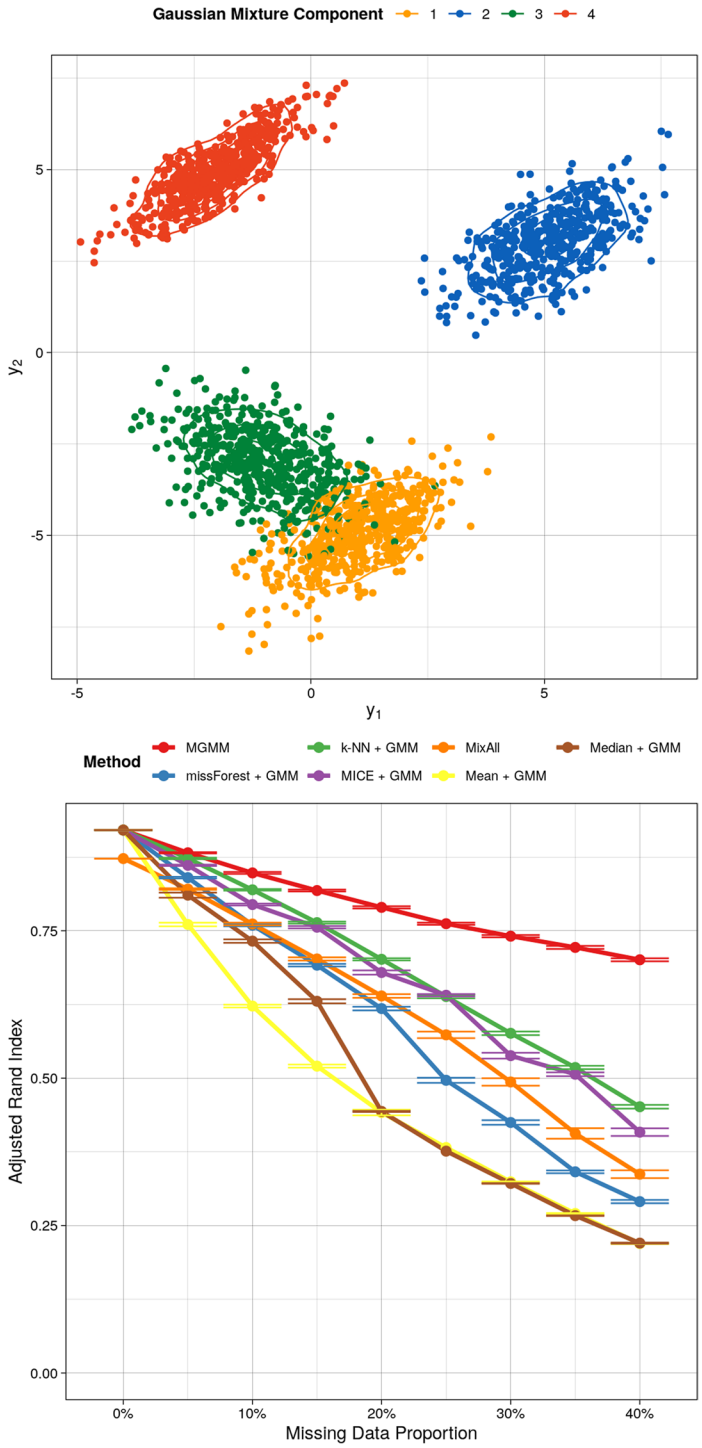

Fig. 3 Benchmarking for the Mixture of Gaussians Data Set. The top panel includes the observations as simulated, colored according to the mixture component. The bottom panel presents the adjusted Rand index as a function of the missing data proportion for 8 different approaches to handling missing data; a higher value indicates better agreement between the predicted and true cluster assignments, adjusting for chance. Error bars represent the standard error of the mean across 20 simulation replicates

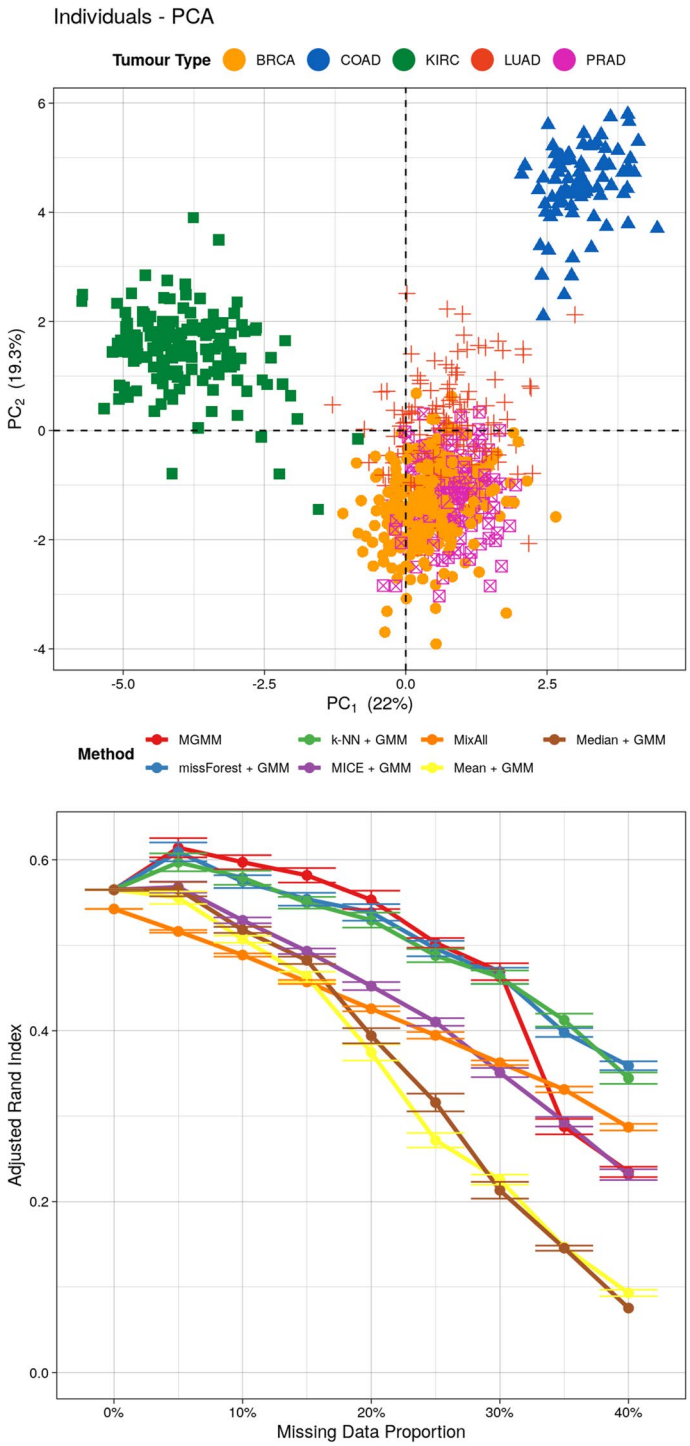

Fig. 4 Benchmarking for the Cancer RNA-Seq Data Set. The top panel includes the projection of the expression data for $n = 801$ cancer patients onto the first two principal components. Observations are colored according to tumor type. The bottom panel presents the adjusted Rand index as a function of the missing data proportion; a higher value indicates better agreement between the predicted and true cluster assignments, adjusting for chance. Error bars represent the standard error of the mean across 20 simulation replicates

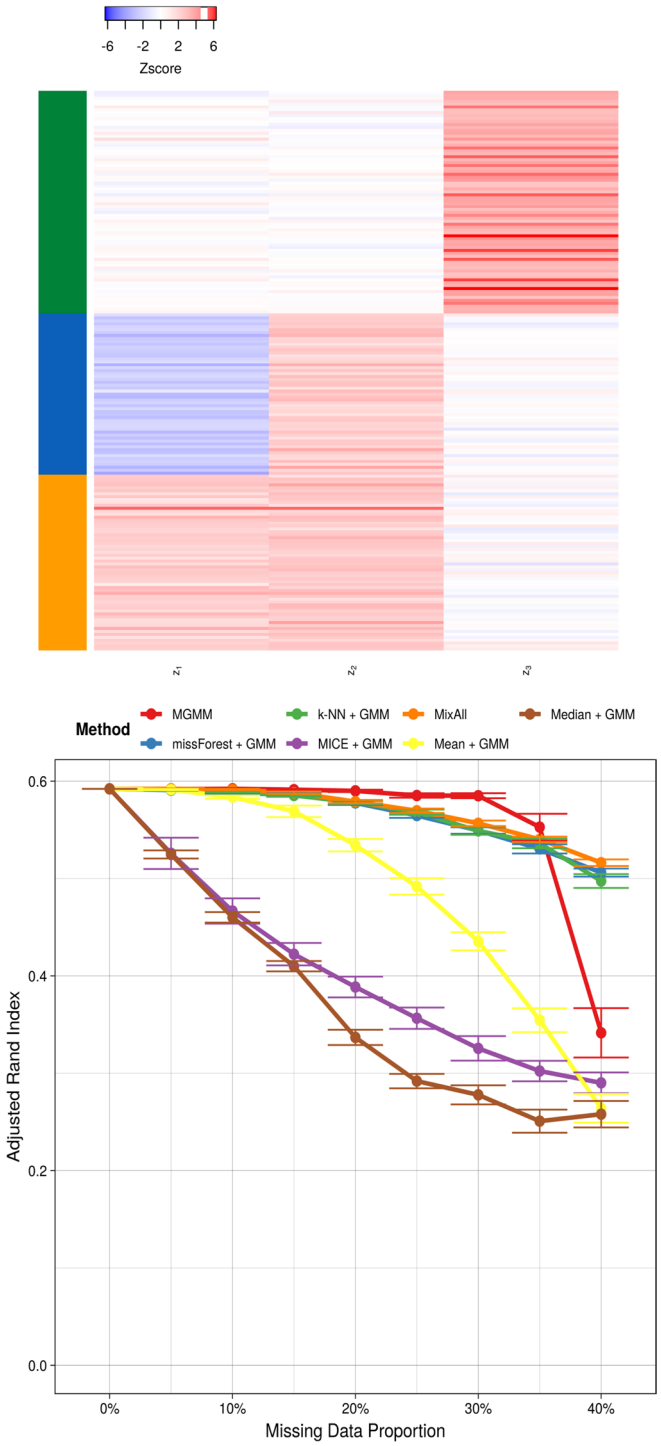

Fig. 5 Benchmarking Simulated Multi-trait GWAS Summary Statistics. The left panel presents a heat map colored according to the normalized genetic effect, with SNPs as rows and traits as columns. The colorbar on the left represents the true cluster assignments. The right panel presents the adjusted Rand index as a function of the missing data proportion; a higher value indicates better agreement between the predicted and true cluster assignments, adjusting for chance. Error bars represent the standard error of the mean across 20 simulation replicates

this simulation, the data generation process `MixAll` because the covariance structure is truly diagonal for the 3rd component of the mixture, and close to diagonal for the 2nd. As for the RNA-Seq example (Fig. 4), random forest and kNN were competitive with `MGMM`, outperforming at very high missingness. Surprisingly, MICE under-performed naive mean imputation, and was comparable to native median imputation.

An analogous clustering task applied to summary statistics from real GWAS of BMI, CAD, and LDL is presented in Fig. 6. The three clusters, identified by applying a 3-component GMM to the data before the introduction of missingness, appear well-differentiated on the heat map. kNN and random forests offered the best performance, followed by `MGMM`, whose performance deteriorated at missingness $\geq$ 35%. The deficit in performance of `MGMM` compared to kNN and random forests, even at low missingness, likely reflects a departure of the true data generating process from a GMM. Similarly, this departure likely explains the overall lower performance of `MixAll` for these data. As in the case of simulated GWAS summary statistics, MICE was not competitive, performing similarly to naive mean and median imputation. Two alternative examples with different set of traits are presented in Supplementary Material (see Supplementary Figs. 1 and 2).

Comparison of MICE-filtered and MGMM-filtered

By effectively removing poorly classifiable observations from consideration, filtering is expected to improve the clustering quality, but only if those observations with high assignment uncertainty are correctly identified. Therefore, the comparative performance of `MGMM`-filtered and MICE-filtered provides an indication of how well each strategy was able to identify those observations with high cluster assignment uncertainty. We present the performances of the two methods on four data sets in Fig. 7.

Four component mixture of bivariate Gaussians

For the Gaussian mixture simulation data set (Fig. 7A), filtering out unassignable observations strikingly improved the classification accuracy of both MICE and `MGMM`. However, `MGMM`-filtered performed better for all missing data ratios. Thus, when the data are in fact generated by a GMM, `MGMM` correctly assesses cluster assignment uncertainty, providing users with a mechanism for identifying observations with low-confidence cluster assignments.

RNA sequence data from cancer patients

For the cancer RNA-Seq data set (Fig. 7B), entropy-based filtering again significantly improved the performance of both methods, suggesting that assignment entropy provides a reliable method for identifying unassignable observations. Note that the filtered data set contained sufficiently many observations to correctly evaluate performance (see Appendix 5). `MGMM`-filtered outperformed MICE-filtered at lower missingness, while MICE-filtered performed better when the missing data proportion exceeded 30%. The same trend was observed for the unfiltered versions of `MGMM` and MICE. This example demonstrates that even when the underlying distribution is not a GMM, `MGMM` is able to accurately assess cluster assignment uncertainty at practical missing ratios.

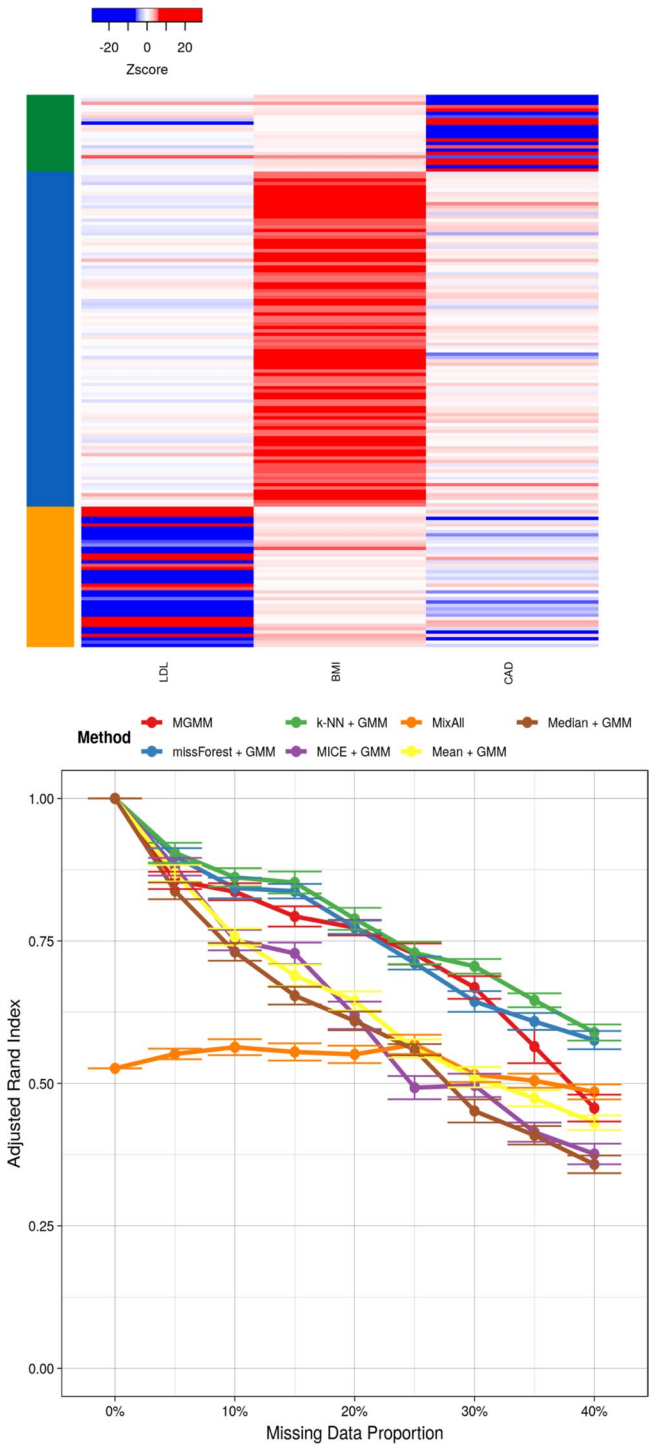

Fig. 6 Benchmarking Real Multi-trait GWAS Summary Statistics for 3 Cardiovascular Risk Factors. These were body mass index (BMI), coronary artery disease (CAD), and low density lipoprotein (LDL). The left panel presents a heat map colored according to the standardized genetic effect, with SNPs as rows and traits as columns. The colorbar on the left represents the true cluster assignments. The right panel presents the adjusted Rand index as a function of the missing data proportion; a higher value indicates better agreement between the predicted and true cluster assignments, adjusting for chance. Error bars represent the standard error of the mean across 20 simulation replicates

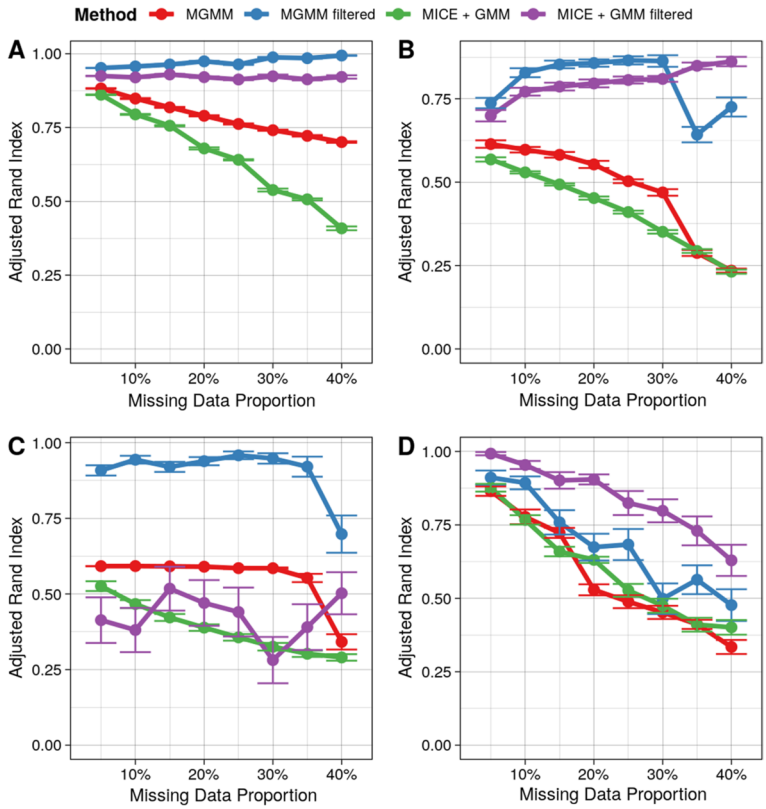

Fig. 7 Performances of MICE-filtered and MGMM-filtered on four Benchmark Data Sets. The adjusted Rand index as a function of the missing data proportion for **A** the Four Component Mixture of Bivariate Gaussians simulation, **B** Cancer RNA-Seq Data Set, **C** Simulated Multi-trait GWAS Summary Statistics, **D** 2nd Example of Real Multi-trait GWAS Summary Statistics for Cardiovascular Risk Factors. Error bars represent the standard error of the mean across 40 simulation replicates

GWAS summary statistics

For the GWAS summary statistic data sets, the comparative performances of the two methods depend on the structure of the data. On the simulated multi-trait GWAS summary statistics (Fig. 7C), filtering drastically improved the performance of MGMM, whereas filtering did little, if anything, to improve the performance of MICE. This suggest that MICE-based imputation entropy was not an effective gauge of assignment uncertainty for these data. The non-linearity and absence of correlation among the variables probably explains the poor performance of MICE.

On the 2nd example of real GWAS summary statistics for cardiovascular risk factors, MICE-filtered performed best overall, and entropy-based filtering improved the performance of MICE more so than the performance of MGMM. The unfiltered versions of MICE and MGMM performed comparably. The strong correlations among the traits studied likely explains the good performance of MICE-filtered for these data. It is also important to note that, for the GWAS data sets, the reference labels used to compute the adjusted rand index are not the true classes *per se*, but rather the clustering obtained on complete data (see the "Methods" section). Therefore, the performance assessment in this example is more a measure of the robustness of the clustering procedure to the presence of missing data than a measure of the capacity to identify true underlying classes.

Filtered observations

Importantly, filtering out unassignable observations based on entropy did not strongly enrich the remaining data for complete cases (see Supplementary Fig. 5). Therefore, the general improvements in performance observed with filtering cannot be trivially explained by the selective removal of incomplete observations, and point instead to the accurate identification of observations that could plausibly have arisen from more than 1 cluster.

Running time benchmark

In terms of running time (see Fig. 8), simple and biased imputation scheme such as mean and median imputation were consistently fastest. The running time of k-NN and random forest remained low for all missing rate. `Mixall` was slower than `MGMM` on complete data and for realistic missing rate. `MGMM` was competitive for the missingness rates usually encountered in real data ($\sim$10%) but its running time increased steeply as the proportion of missing data became large. MICE followed by `MGMM` was the slowest method overall and was slow even for low missingness rates. This naturally follows from needing to perform multiple rounds of imputation followed by GMM estimation with MICE. Although MICE

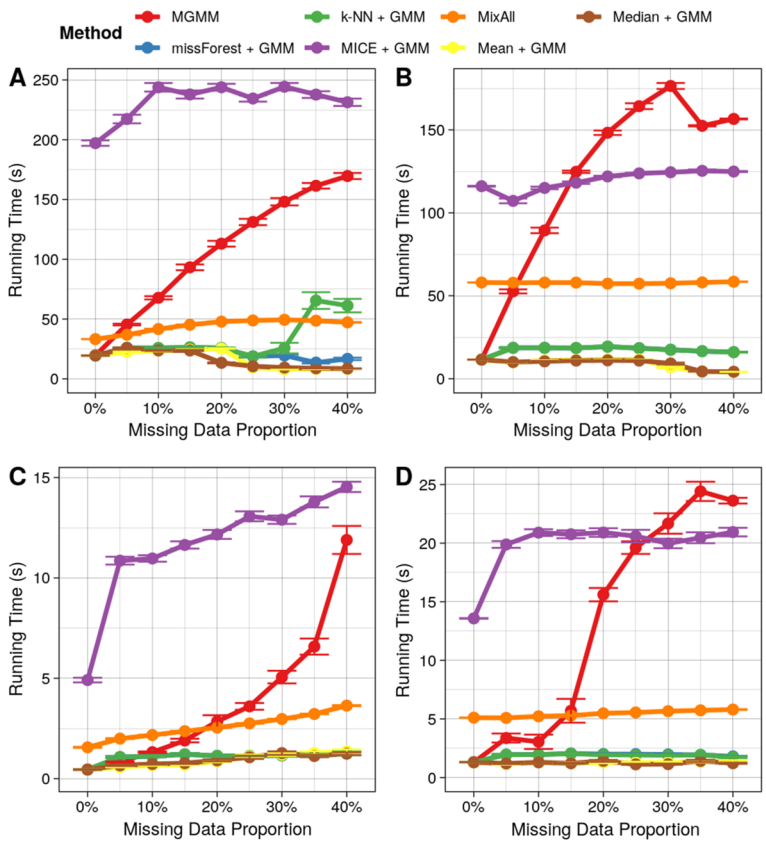

Fig. 8 Comparison of computation time for studied methods. The computation time as a function of the missing data proportion for **A** the Four Component Mixture of Bivariate Gaussians simulation, **B** Cancer RNA-Seq Data Set, **C** Simulated Multi-trait GWAS Summary Statistics, **D** 2nd Example of Real Multi-trait GWAS Summary Statistics for Cardiovascular Risk Factors. Error bars represent the standard error of the mean across 40 simulation replicates

and MGMM were generally the slowest methods, these are also the only methods that provide a mechanism for identifying and filtering out observations with high assignment uncertainty. Moreover, to put the computation cost into perspective, the longest observed running time was 344 seconds, which remains tractable (obtained with MICE on the Mixture of Bivariate Gaussians example, Fig. 8A).

Discussion

We conducted a comparative benchmark to assess the capacity of MGMM versus MixAll and standard GMM after imputation to correctly identify true cluster assignments in data containing missing values. We established that for data sets following a distribution close to a GMM, MGMM is able to recover the true class assignment more accurately than imputation followed by standard GMM. When the underlying data generating process is in fact a GMM, then as a correctly specified maximum likelihood procedure, MGMM is optimal. MGMM consistently outperformed the other existing GMM implementation that allows for missing data (i.e. MixAll), except when the proportion of missing data became excessive. The better performance of MGMM at low levels of missingness is likely because MGMM places no restrictions on the form of the covariance matrix. At high levels of missingness, adopting the parsimonious assumption of a diagonal covariance structure, as is done by MixAll, can be advantageous. However, for a fixed proportion of missing data, MGMM should match or exceed the performance of MixAll as sample sizes increases. In addition, MGMM correctly assess its level of uncertainty in clustering assignments, providing a mechanism for identifying and separating out observations whose cluster assignments are unreliable.

GMMs are not well-suited to all clustering tasks. Direct application of MGMM was less effective than non-linear imputation, via kNN or random forests, followed by standard GMM in cases where the clusters present in the observed data were poorly differentiated, or the missingness was high (e.g. 40%). This observation emphasizes the need to assess the appropriateness of a GMM before applying MGMM to a clustering problem. Since kNN and random forest imputation, followed by standard GMM, were typically competitive with MGMM in the real data examples, these methods may be used to perform sensitivity analysis on the final cluster assignments. On the other hand, standard GMM following kNN or random forest imputation will not appropriately propagate uncertainty due to missing data. This can lead to inaccurate estimates of the posterior membership probabilities, particularly for observations with multiple missing elements, and failure to identify observations whose cluster assignments are unreliable. Thus, an approach such as MGMM-filtered, which accurately assesses assignment uncertainty and removes unclassifiable observations from consideration, may be more reliable. The framework proposed by [6], and elaborated upon here, of using an EM-type algorithm to fit mixture models in the presence of both missing data and unknown class assignments, may be extended to estimates mixtures of non-Gaussian distributions. Extending MGMM to estimate such mixtures in the presence of missing data is among our future directions.

Conclusion

We have presented MGMM, a powerful, general purpose R package for maximum likelihood-based estimation of GMMs in the presence of missing data, and demonstrated that MGMM often outperforms both MixAll and imputation followed by standard

GMM on various real and simulated data sets. In contrast to estimation after imputation, `MGMM` uses the ECM algorithm to efficiently and unbiasedly obtain the maximum likelihood estimates of all model parameters while properly accounting for the uncertainty introduced by the presence of missing values; and in contrast to `MixAll`, which also employs maximum likelihood estimation, `MGMM` does not assume the data are uncorrelated. To our knowledge, `MGMM` is the only publicly available method for fitting GMMs that properly accounts for missing data without imposing simplifying assumptions, and our benchmark is the first extensive study of how estimating GMMs while properly accounting for missing data compares with the *ad hoc* procedure of estimation after imputation. In addition, the supporting information (Additional file 1) provides a clear step-by-step derivation of our ECM algorithm, providing a foundation for extending this work to missingness-aware mixtures of other distributions. The functionalities of the `MGMM` package [7] are carefully documented and comprise: the generation of random data under a specified GMM, the fitting of GMMs to data sets containing missing values, the drawing of multiple imputations for a fitted model, and the computation of a panel of clustering criteria to identify the optimal number of clusters.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-022-04740-9.

Additional file 1: Detailed derivation; discussion of assumptions, cluster-number selection, and multiple-imputation; additional simulation materials.

Additional file 2. Replication scripts and benchmark data.

Acknowledgements
The authors thank the anonymous reviewers for their valuable suggestions.

Author contributions
ZM Software Conception and implementation, manuscript writing and reviewing. HJ Benchmark conception and implementation, manuscript writing and reviewing. HA Manuscript reviewing. All authors read and approved the final manuscript.

Funding
This research was supported by the Agence Nationale pour la Recherche (ANR-20-CE36-0009-02). This work is supported in part by funds from the National Science Foundation (NSF: # 1636933 and # 1920920).

Availability of data and materials
`MGMM` is available as an `R` package on CRAN: https://CRAN.R-project.org/package=MGMM. The data used in this study are either publicly available or were randomly generated according to the procedure detailed in the Materials and Methods. We provide an archive (Additional file 2) containing replication scripts and data.

Declaration

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 27 October 2021 Accepted: 16 May 2022
Published: 1 June 2022

References

1. Murphy KP. Machine learning: a probabilistic perspective. 1st ed. Cambridge: The MIT Press; 2012.

2. Fraley C, Raftery A. mclust: software for model-based cluster analysis. J Classif. 1999;16:297–306.
3. Benaglia T, Chauveau D, Hunter D, Young D. Mixtools: an r package for analyzing mixture models. J Stat Softw. 2009;32(6):1–29. https://doi.org/10.18637/jss.v032.i06.
4. Iovleff S, Bathia P. MixAll: clustering and classification using model-based mixture models. R Foundation for Statistical Computing, Vienna, Austria 2019. R Foundation for Statistical Computing. https://CRAN.R-project.org/package=MixAll
5. Julienne H, Laville V, McCaw ZR, He Z, Guillemot V, Lasry C, Ziyatdinov A, Vaysse A, Lechat P, Ménager H, Goff WL, Dube MP, Kraft P, Ionita-Laza I, Vilhjálmsson BJ, Aschard H. Multitrait genetic-phenotype associations to connect disease variants and biological mechanisms. bioRxiv 2020. https://doi.org/10.1101/2020.06.26.172999
6. Ghahramani Z, Jordan M. Supervised learning from incomplete data via an em approach. In: Advances in neural information processing systems 6. Morgan-Kaufmann; 1994. pp. 120–127.
7. McCaw Z. MGMM: Missingness aware Gaussian mixture models. R Foundation for Statistical Computing, Vienna, Austria 2021. R Foundation for Statistical Computing. https://CRAN.R-project.org/package=MGMM
8. Meng X-L, Rubin DB. Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika. 1993;80(2):267–78.
9. Buuren SV, Groothuis-Oudshoorn K. mice: multivariate imputation by chained equations in r. J Stat Softw. 2010;45:1–68.
10. Little R, Rubin D. Statistical analysis with missing data. 2nd ed. New York: Wiley; 2002.
11. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria 2017. R Foundation for Statistical Computing. https://www.R-project.org/
12. Kowarik A, Templ M. Imputation with the r package vim. J Stat Softw. 2016;74(7):1–16.
13. Stekhoven DJ, Bühlmann P. Missforest-non-parametric missing value imputation for mixed-type data. Bioinformatics. 2011;28(1):112–8.
14. Julienne H, Lechat P, Guillemot V, Lasry C, Yao C, Araud R, Laville V, Vilhjalmsson B, Ménager H, Aschard H. JASS: command line and web interface for the joint analysis of GWAS results. NAR Genomics Bioinform. 2020;2(1):003.
15. Rand WM. Objective criteria for the evaluation of clustering methods. J Am Stat Assoc. 1971;66(336):846–50.
16. Hubert L, Arabie P. Comparing partitions. J Classif. 1985;2(1):193–218.
17. Weinstein JN, Collisson EA, Mills GB, Shaw KRM, Ozenberger BA, Ellrott K, Shmulevich I, Sander C, Stuart JM, Network CGAR, et al. The cancer genome atlas pan-cancer analysis project. Nat Genet. 2013;45(10):1113.

Publisher's Note