INTERPRETABLE CLUSTERING WITH THE DISTINGUISHABILITY CRITERION

A PREPRINT

Ali Turfah
Department of Biostatistics
University of Michigan
Ann Arbor, MI, 48105
aturfah@umich.edu

Xiaoquan Wen*
Department of Biostatistics
University of Michigan
Ann Arbor, MI, 48105
xwen@umich.edu

April 24, 2024

ABSTRACT

Cluster analysis is a popular unsupervised learning tool used in many disciplines to identify heterogeneous sub-populations within a sample. However, validating cluster analysis results and determining the number of clusters in a data set remains an outstanding problem. In this work, we present a global criterion called the Distinguishability criterion to quantify the separability of identified clusters and validate inferred cluster configurations. Our computational implementation of the Distinguishability criterion corresponds to the Bayes risk of a randomized classifier under the 0-1 loss. We propose a combined loss function-based computational framework that integrates the Distinguishability criterion with many commonly used clustering procedures, such as hierarchical clustering, k-means, and finite mixture models. We present these new algorithms as well as the results from comprehensive data analysis based on simulation studies and real data applications.

Keywords Unsupervised learning · Cluster analysis · k-means · Hierarchical clustering · Mixture models

1 Introduction

Cluster analysis is a ubiquitous unsupervised learning approach to uncover latent structures and patterns in observed data. Clustering algorithms have been used in a wide variety of scientific applications, such as animal behavior studies [1], weather anomaly detection [2], disease diagnosis [3, 4, 5], and novel cell type identification [6, 7, 8, 9]. Often, the identified clusters are interpreted to represent distinct populations from which the corresponding samples originate.

Many challenges with cluster analysis, such as determining the number of clusters, arise from an inability to rigorously quantify desired cluster characteristics. While the precise definition of a "meaningful" cluster is usually context-dependent, it is generally accepted that the clusters should display "internal cohesion" (i.e., objects within a cluster are similar to one another) and "external isolation" (i.e., the clusters are well-separated) [10, 11, 12]. Despite this almost universally agreed-upon principle, quantifying the separability of the clusters, i.e., the level of external isolation with respect to internal cohesion, remains an open problem in cluster analysis.

In this paper, we introduce the Distinguishability criterion to measure the separability of a set of assumed clusters. The criterion is motivated by this simple intuition: if all clusters are well separated from each other, then the originating clusters for all data points (whether observed or not) should be easily traceable. To implement the Distinguishability criterion, we formulate labeling the generating cluster for an arbitrary data point as a probabilistic classification problem. Naturally, the difficulty (or lack thereof) of this classification problem can be described by an overall misclassification probability averaged over all possible data points.

We employ a statistical viewpoint to define the Distinguishability criterion for cluster analysis. The partitioned observed data are taken to be realizations from cluster-specific data generative distributions, which are essential for computing the

*Corresponding Author

proposed misclassification probability. Although not all clustering algorithms make explicit distributional assumptions for the presumed clusters, many commonly applied heuristics-based algorithms achieve optimal performance under specific probabilistic generative models [13, 14]. Furthermore, the identified cluster structures from cluster analysis are typically expected to be replicated in future datasets—an implicit assumption for consistent data generative distributions. As a result, statistical inference procedures based on explicit distributional assumptions have become more popular for their ability to not only enhance clustering performance but also to examine and interpret cluster structures identified by both model and heuristics-based clustering methods [15, 16, 17, 18].

The remainder of this paper is organized as follows. We first provide the mathematical definition and properties of the Distinguishability criterion. We then discuss its usage with existing clustering algorithms. Finally, we illustrate the applications of the Distinguishability criterion using both synthetic and real data from various scientific applications. Our implementation of the Distinguishability criterion and the analyses presented in this paper can be found at `https://github.com/aturfah/distinguishability-criterion`.

2 Results

2.1 The Distinguishability Criterion

The proposed Distinguishability criterion measures the overall separability of a given cluster configuration and is derived by quantifying the misclassification probability from a multi-class classification problem.

Given a cluster configuration where each of the K clusters corresponds to a distinct class, we denote the class label for an observation $\boldsymbol{x}$ by $\theta \in \{1, 2, ..., K\}$. We assume a pre-defined classifier, $\delta(\boldsymbol{x}) : \mathbb{R}^p \to \{1, ..., K\}$, and evaluate the classification performance using the 0-1 loss function, i.e.,

$$L(\delta(\boldsymbol{x}), \theta) = \mathbf{1}\{\delta(\boldsymbol{x}) \neq \theta\}.$$

The overall misclassification probability under the assumed cluster configuration, denoted by P_{mc}, is defined as the Bayes risk of the classifier $\delta(\boldsymbol{x})$, i.e.,

$$P_{\text{mc}} = \text{E}_{\boldsymbol{x}}\Big[\, \text{E}_{\theta}\big(\, L(\delta(\boldsymbol{x}), \theta) \mid \boldsymbol{x} \,\big) \,\Big] \tag{1}$$

Using the 0-1 loss ensures that the resulting Bayes risk is a valid probability measurement, ranging from 0 to 1. It is naturally interpreted as the probability of misclassifying a data point under the given cluster configuration, marginalizing all potential $\boldsymbol{x}$ values and their respective true generating clusters. For instance, a P_{mc} value close to 0 signifies a high degree of cluster separation, indicated by a minimal probability of erroneously assigning an arbitrary data point to an incorrect generating cluster (Supplementary Figure 2).

As we are primarily interested in assessing different cluster configurations, the selection of the required classifier is flexible. However, the choice of classifier can impact the computational efficiency of the P_{mc} evaluation. Our implementation focuses on the set of classifiers working directly with the probabilities

$$\pi_k(\boldsymbol{x}) := \Pr(\theta = k \mid \boldsymbol{x}), \;\; \text{for } k = 1, ..., K.$$

This set includes the optimal classifier under the 0-1 loss, δ_o, i.e.,

$$\delta_o(\boldsymbol{x}) = \arg\max_k \pi_k(\boldsymbol{x})$$

Our default classifier for computing P_{mc} is a randomized decision function, δ_r, which assigns a label to an observation $\boldsymbol{x}$ by sampling from a categorical distribution based on the probability distribution $\boldsymbol{\pi}(\boldsymbol{x}) = (\pi_1(\boldsymbol{x}), ..., \pi_K(\boldsymbol{x}))$, i.e.,

$$\delta_r(\boldsymbol{x}) \sim \text{Categorical}\,(\boldsymbol{\pi}(\boldsymbol{x})).$$

In addition to yielding highly comparable P_{mc} values to the optimal classifier within the decision-critical ranges of cluster separation (Supplementary Figure 1), the randomized classifier's computational properties enable highly efficient cluster analysis procedures.

The π_k's are the key quantities bridging the observed clustering data and P_{mc}. They are calculated using Bayes rule,

$$\pi_k(\boldsymbol{x}) \propto \alpha_k(\boldsymbol{X}_c)\, p(\boldsymbol{x} \mid \theta(\boldsymbol{X}_c) = k).$$

The notation emphasizes that both the prior, $\alpha_k(\boldsymbol{X}_c)$, and the likelihood function, $p(\boldsymbol{x} \mid \theta(\boldsymbol{X}_c))$, are directly informed by and estimated from the clustering data. More specifically, the prior quantifies the relative frequency of the observations arising from each assumed cluster, while the likelihood function encodes the characteristics of the corresponding cluster population, such as its centroid and spread information. Computing π_k values is straightforward for model-based clustering algorithms such as Gaussian mixture models (GMMs) [14]. For non-model-based clustering algorithms, explicit distributional assumptions specifying the parametric family of likelihood functions are required. We illustrate these procedures for k-means and hierarchical clustering algorithms in subsequent sections.

In summary, P_{mc} is a probability measurement of global separability across inferred clusters. It can accommodate a wide range of distributional assumptions, making it compatible with a diverse set of clustering procedures and data modalities. Moreover, as a function of clustering data, $\boldsymbol{X}_c$, the estimate of P_{mc} itself is a valid loss function suitable for selecting optimal cluster configurations in cluster analysis.

2.2 Combined Loss Function for Cluster Analysis

In cluster analysis, the desired cluster characteristics are often defined by multiple criteria [19]. A single criterion on its own, including the proposed Distinguishability criterion, is insufficient to define a practically optimal clustering solution. Alternatively, combining multiple loss functions targeting different desired cluster properties can result in more balanced and holistic clustering solutions. This observation leads to a principled way to incorporate P_{mc} with other established clustering criteria and algorithms.

Specifically, let L_1 denote a loss function associated with existing clustering algorithms. Formally, we consider a compound loss, L, as a weighted linear combination of L_1 and P_{mc}, i.e.,

$$L = L_1 + \lambda P_{\text{mc}},\ \lambda > 0. \tag{2}$$

Because of the scale of P_{mc}, it is often convenient to solve the following equivalent constrained optimization problem,

$$\text{Minimize } L_1, \text{ subject to } P_{\text{mc}} \leq \tau, \tag{3}$$

where τ is a pre-specified probability threshold. It is worth noting that the stringency of the τ value may depend on the dimensionality of the clustering data.

The choice of clustering algorithm determines the functional form of L_1. For example, the distortion function or Ward's linkage are natural choices for L_1 when using k-means and hierarchical clustering methods, respectively. Alternatively, the negative of the gap statistic [20] can also be an excellent choice in these application scenarios. For model-based clustering algorithms, the L_1 function can be derived from various model selection criteria, e.g., the negative of Bayesian information criterion (BIC).

2.3 Connections to Related Approaches

The misclassification probability defined by P_{mc} falls into the category of internal clustering validity indices [21, 22, 23], which assess the quality of a clustering solution without additional external information beyond the observed data. This category includes many commonly applied statistical measures, such as the Silhouette index [24], Calinski-Harabaze index [25], Dunn index [26], among others. A common behavior of internal clustering validity indices is that they evaluate both the cohesion (or compactness) within a cluster as well as the separation between clusters. For P_{mc}, the within-cluster cohesion is quantified through the estimated parameters in the likelihood function, $p(\boldsymbol{x} \mid \theta(\boldsymbol{x}_c) = k)$. The separation, relative to the cohesion, is quantified by the overall misclassification probability.

Henning [19] and Melnykov [27] also compute misclassification probabilities to assess the separation between clusters in the context of mixture models for clustering. They introduce the metrics—named "directly estimated misclassification probability" (DEMP) and DEMP+—specifically designed to compute misclassification probabilities between pairs of clusters to inform decisions about the local merging of mixture components. In comparison, P_{mc} is a global measure of the misclassification probability across all clusters. It, too, can be used to combine mixture components to form interpretable clusters, as illustrated in Section 2.4. Additionally, the entropy criterion proposed by Celeux and Soromenho [28, 29, 30] provides another alternative approach to quantify the separability of clusters using a classification problem set-up.

The Distinguishability criterion also agrees with the principle of stability measures commonly employed in clustering analysis. Specifically, if the underlying cluster distributions are all well-separated, as indicated by low $P_{\rm mc}$ values, data sampled repeatedly from these distributions are expected to produce consistent clustering outcomes [31, 32, 33]. Empirical evidence has shown that $P_{\rm mc}$ and various measures of clustering instability are highly correlated, which is demonstrated in the subsequent sections.

2.4 Finite Mixture Models Incorporating $P_{\rm mc}$

Finite mixture models (MM) are probabilistic models that can seamlessly incorporate the Distinguishability criterion. MM-based clustering algorithms primarily infer the distributional characteristics underlying each latent cluster. As a result, no additional assumptions are needed to compute $P_{\rm mc}$ in the MM setting — the required quantities, $\{\alpha_k,\, p(\boldsymbol{x} \mid \theta = k),\, \pi_k\}$, are all direct outputs or by-products from standard MM inference procedures [14], e.g., the EM algorithm.

We make an important distinction between a mixture component and an interpretable cluster, a point previously discussed by [27, 29, 30, 34]. We view mixture models as flexible density estimation devices, where the number of mixture components is chosen to adequately fit the observed data. On the other hand, the distribution of an underlying cluster—characterized by the Distinguishability criterion—may itself be a mixture distribution comprising multiple components. This distinction naturally leads to combining the loss functions represented by $-\mathrm{BIC}$, which evaluates the goodness-of-fit of a mixture density, and $P_{\rm mc}$, which characterizes the separation between potential clusters.

To optimize the combined loss function, we first find $P(\boldsymbol{x})$, the optimal mixture distribution with κ components, by maximizing the BIC. Subsequently, we merge the mixture components into clusters until $P_{\rm mc}$ falls below a pre-specified threshold τ. Since merging mixture components into clusters does not alter the mixture component distributions in any way, the BIC is unchanged by the merging process. It can be shown that merging existing clusters in this manner always decreases $P_{\rm mc}$ (Appendix B). Specifically, under the default randomized classifier δ_r, the reduction of $P_{\rm mc}$ by combining clusters i and j is given by

$$\Delta P_{\rm mc}^{(i,j)} = 2 \int \pi_i(\boldsymbol{x})\pi_j(\boldsymbol{x})\, P(d\boldsymbol{x}). \tag{4}$$

For clusters with little to no overlap, i.e., $\pi_i(\boldsymbol{x})\pi_j(\boldsymbol{x}) \to 0$ for all $\boldsymbol{x}$ values, merging (i, j) results in minimal changes in $P_{\rm mc}$. On the other hand, for clusters with significant overlap, $\Delta P_{\rm mc}^{(i,j)}$ can be substantial.

By further utilizing the cluster merging property of $P_{\rm mc}$ (Proposition 1, Methods Section), i.e.,

$$P_{\rm mc} = \sum_{i<j} \Delta P_{\rm mc}^{(i,j)}, \tag{5}$$

we propose an efficient $P_{\rm mc}$ Hierarchical Merging (PHM) algorithm to sequentially amalgamate mixture components into clusters (Algorithm 1). Briefly, starting by assigning each of the κ mixture components to individual clusters, the PHM algorithm pre-computes $\Delta P_{\rm mc}^{(i,j)}$ for all (i, j) cluster pairs. It then sequentially combines the pairs of clusters with the largest $\Delta P_{\rm mc}^{(i,j)}$ into a single cluster and updates the $\Delta P_{\rm mc}$ values for the remaining clusters. The process is repeated until the updated $P_{\rm mc}$ falls at or below a pre-defined τ value. Intuitively, this procedure prioritizes merging the most similar or closely related clusters at each step, quantified by their $\Delta P_{\rm mc}^{(i,j)}$ value.

By setting $\tau = 0$, the algorithm runs until all mixture components have been merged into a single cluster. The complete merging process can be visualized using a dendrogram (Appendix C), characterizing the hierarchical merging orders between individual mixture components and merged clusters. In many scientific applications, e.g., genetics and single-cell data analysis, such a dendrogram can provide a snapshot of the underlying continuous differentiation process that forms the identified clusters. We provide two examples in our real data applications (Section 2.7).

To illustrate the PHM algorithm with Gaussian mixture models (GMMs), we use the synthetic data from Section 4.1 in Baudry et al. [30]. Specifically, 600 observations are drawn from six Gaussian distributions arranged along the corners of a square in the following manner. Two overlapping Gaussian distributions are placed in the top left corner of the square, each with 1/5 of the samples. The bottom left and top right corners each have a single Gaussian distribution, each contributing 1/5 of the samples. Finally, two overlapping Gaussian distributions are placed in the bottom right corner, each with 1/10 of the samples.

A GMM with six components is selected by BIC using the `R` package `mclust` [35]. The observed clustering data is shown in the left panel of Figure 1, with colors corresponding to an observation's assignment to one of the κ GMM components. The initial cluster configuration labeling each mixture component as a single cluster has $P_{\rm mc} = 0.139$.

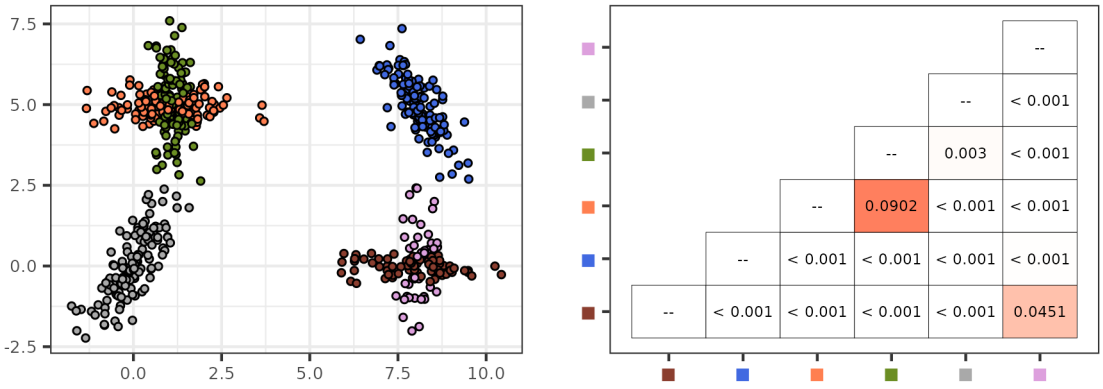


Figure 1: *(Left)* 600 simulated observations drawn from a mixture of six two-dimensional Gaussian distributions. Colors indicate the cluster assignment labels to each of the six estimated mixture components. *(Right)* Heatmap visualizing ΔP_{mc} values for the estimated mixture components. The intensity of the color indicates the relative proportion of P_{mc} contributed by the overlap between these components (i.e., $\Delta P_{\text{mc}}^{(i,j)}$).

The heatmap in the right panel of Figure 1 visualizes the $\Delta P_{\text{mc}}^{(i,j)}$ contributions from each pair of mixture components, showing that the main sources of P_{mc} come from the overlapping components in the top left and bottom right corners.

With a threshold of $\tau = 0.01$, the PHM algorithm sequentially combines the mixture components in the top left and bottom right corners, reducing the values of P_{mc} to 0.049 and 0.004, respectively. In the end, the algorithm returns four clusters: one corresponding to the components in each of the four corners.

2.5 Applications in k-means Clustering

Applying the Distinguishability criterion to heuristics-based clustering algorithms requires additional distributional assumptions to compute P_{mc}. Although popular algorithms of this kind—such as k-means and hierarchical clustering—rely on intuitive heuristics, their implicit connections to probability models have been well studied. These results provide insights into the data types for which certain non-model-based clustering algorithms are expected to be optimal. The k-means algorithm, in particular, has been shown as equivalent to approximately optimizing a multivariate Gaussian classification likelihood function [13, 14]. Hence, when applying the Distinguishability criterion with the usual k-means distortion function, it seems natural to assume that data within each inferred partition are normally distributed. It then

Algorithm 1: P_{mc} Hierarchical Merging (PHM) algorithm

Input: Input data X, P_{mc} threshold τ
Result: Groupings of mixture components into clusters
Procedure PHM
  Fit a mixture model to X, determining the number of components by maximizing BIC
  Initialize clusters to individual mixture components
  Compute $\Delta P_{\text{mc}}^{(i,j)}$ for all pairs of clusters i, j
  while $P_{\text{mc}} > \tau$ **do**
    Group clusters i, j with maximal $\Delta P_{\text{mc}}^{(i,j)}$ into a single cluster k'
    Update the distribution quantities for this new cluster:
      $\alpha_{k'} \leftarrow \alpha_j + \alpha_i$
      $p(\boldsymbol{x} \mid \theta = k') \leftarrow \alpha_{k'}^{-1} \cdot [\, \alpha_i \cdot p(\boldsymbol{x} \mid \theta = i) + \alpha_j \cdot p(\boldsymbol{x} \mid \theta = j) \,]$
      $\pi_{k'}(\boldsymbol{x}) \leftarrow \pi_i(\boldsymbol{x}) + \pi_j(\boldsymbol{x})$
    Update $P_{\text{mc}} \leftarrow P_{\text{mc}} - \Delta P_{\text{mc}}^{(i,j)}$
    Compute $\Delta P_{\text{mc}}^{(k',k)}$ for all uninvolved clusters k: $\Delta P_{\text{mc}}^{(k',k)} = \Delta P_{\text{mc}}^{(i,k)} + \Delta P_{\text{mc}}^{(j,k)}$
  return

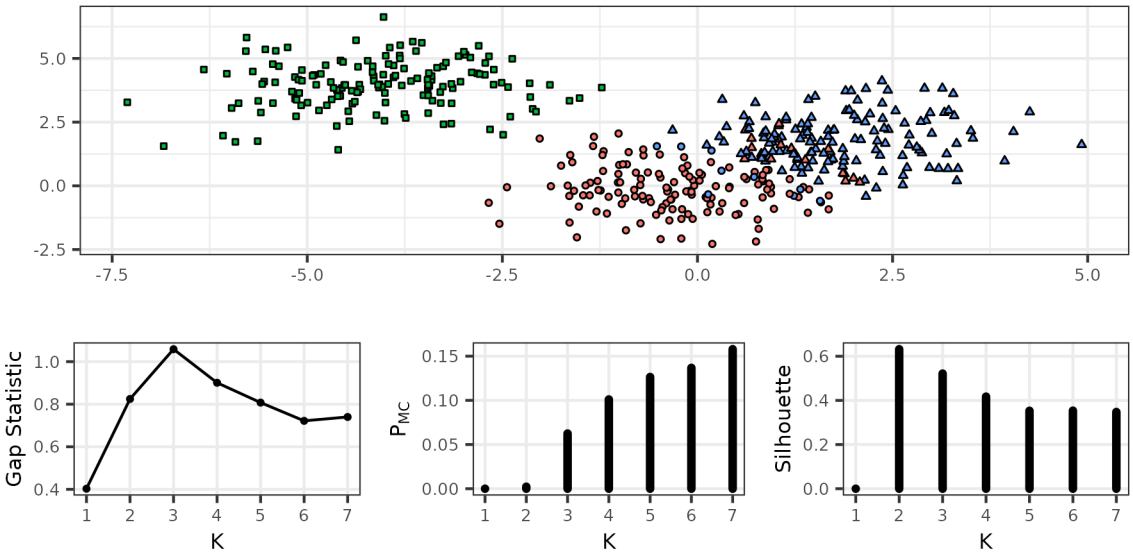


Figure 2: *(Left)* 450 simulated observations drawn from a mixture of three Gaussian distributions. Color indicates true generating distribution while shape indicates the assigned k-means cluster. (*Center and Right*) Value of the gap statistic, P_{mc}, and the Silhouette index for different numbers of clusters based on the k means clustering partition with Gaussian cluster distributions.

becomes straightforward to estimate the necessary parameters and compute P_{mc} given the partitioned data from the k-means output.

We illustrate a procedure to determine the number of clusters (K) in k-means clustering by optimizing the combined loss of P_{mc} and the gap statistic [20]. For a given data partitioning from the k-means algorithm, the gap statistic compares the observed within-cluster dispersion to the expected dispersion under a null reference distribution. It subsequently estimates the optimal number of clusters corresponding to the largest gap statistic value from a range of potential K values. We consider observations drawn from three Gaussian clusters in $\mathbb{R}^2$ centered at (0, 0), (1.75, 1.75), and (-4, 4) with identity covariance matrices ($n_k = 150$). The observed data are shown in Figure 2, with substantial overlap between the data generated from two of the three distributions. The k-means algorithm is performed for $K = 1$ to 7 using the R package `ClusterR` [36] with `kmeans++` initialization [37]. In addition to the P_{mc} values, we also present the averaged Silhouette index [24] for each value of $K = 2, \ldots, 7$.

The values of the gap statistic, P_{mc}, and the Silhouette index for different K are shown in Figure 2. When used alone, the gap statistic selects $K = 3$, coinciding with the number of generating distributions. There is a noticeable difference in the inferred clusters' separability between $K = 2$ ($P_{\text{mc}} = 0.002$) and $K = 3$ ($P_{\text{mc}} = 0.062$). For a P_{mc} threshold $\tau = 0.01$, optimizing the combined loss function (3) leads to selecting $K = 2$, which seems most reasonable judging by the data visualization. This decision is also consistent with the Silhouette index, which takes a maximal value of 0.632 at $K = 2$ compared to 0.522 at $K = 3$. Additionally we find that P_{mc} is highly negatively correlated with cluster stability measures. We present the values of the stability measure proposed by Lange et al. [32] using the adjusted Rand index to compute cluster similarities as well as the prediction strength [33] in Supplementary Table 1.

As noticed in the original paper [20], the gap statistic can struggle to determine K when the underlying generative distributions have substantial overlapping support. The above numerical example illustrates that the Distinguishability criterion can alleviate this challenge in k-means clustering.

2.6 Hypothesis Testing in Hierarchical Clustering

It has become increasingly common to perform formal statistical testing in post-clustering analysis to reduce cluster over-identification. Recent works by Gao et al. [16], Chen et al. [17], and Grabski et al. [18] highlight the necessity and importance of such analysis in scientific applications, where false positive findings of clusters are considered more costly than false negative findings. In many application contexts, it is often of interest to assess whether the partitions of the observed data output by heuristics-based clustering algorithms (e.g., hierarchical clustering) could arise from a single homogeneous distribution (most commonly, a single Gaussian distribution) by chance. P_{mc} emerges as a

natural test statistic in this parametric hypothesis testing framework because its values are expected to be quantitatively different under the null and alternative scenarios.

Formally, we consider testing the null hypothesis,

$$H_0 : \text{the data are generated from a single Gaussian distribution}$$

in the hierarchical clustering setting [15, 16, 18]. Following hierarchical clustering of observed data, we compute P_{mc} for the pair of inferred clusters resulting from the first split of the dendrogram (i.e., $K = 2$). The null distribution of P_{mc} can be estimated by a simple Monte Carlo procedure that repeatedly samples data from H_0, performs hierarchical clustering, and computes P_{mc} for the partitions defined by the first split. Alternatively, the p-value of the P_{mc} statistic can be derived from the standard parametric bootstrap procedure [38].

We present the results of simulation studies examining the performance of P_{mc} in this hypothesis testing setting.

First, we illustrate P_{mc}'s ability to control for spurious cluster detection. For 5,000 simulation replicates, we draw 150 observations from a $N(0, 1)$ distribution and perform hierarchical clustering based on the squared Euclidean distance with the Ward linkage. We compute P_{mc} and derive the corresponding p-values based on the estimated null distribution from 5,000 additional Monte Carlo simulations, each with a sample size of 150. We also compute the P_{mc} p-values using a bootstrap procedure, using 500 bootstrap replicates to estimate the p-value for each simulation replicate. For comparison, we also compute the p-values from the selective inference procedure proposed by Gao et al. [16] as well as the standard two-sample t-test. The results are shown in the top panel of Figure 3. As expected, the p-values from the two-sample t-test are all in the range of 10^{-20} to 10^{-40}, providing no control for false positive findings. This is because the naive t-test fails to take into account that the hierarchical clustering procedure always partitions data according to their observed values even under H_0, as noted by [16]. In contrast, p-values derived from P_{mc} and Gao et al.'s methods are both roughly uniformly distributed under H_0, suggesting well-controlled type I error rates. Specifically, we find that $P_{\text{mc}} = 0.094$ corresponds to the cutoff at the 5% α level, and the realized type I error rate is 4.8%. The type I error rate from the bootstrap procedure using P_{mc} at the same control level is 1%, which is more conservative.

Second, we examine the power of the P_{mc}-based hypothesis test to identify truly separated clusters. Specifically, we draw samples from two distinct Gaussian distributions, $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$, where a range of $|\mu_1 - \mu_2|/\sigma$

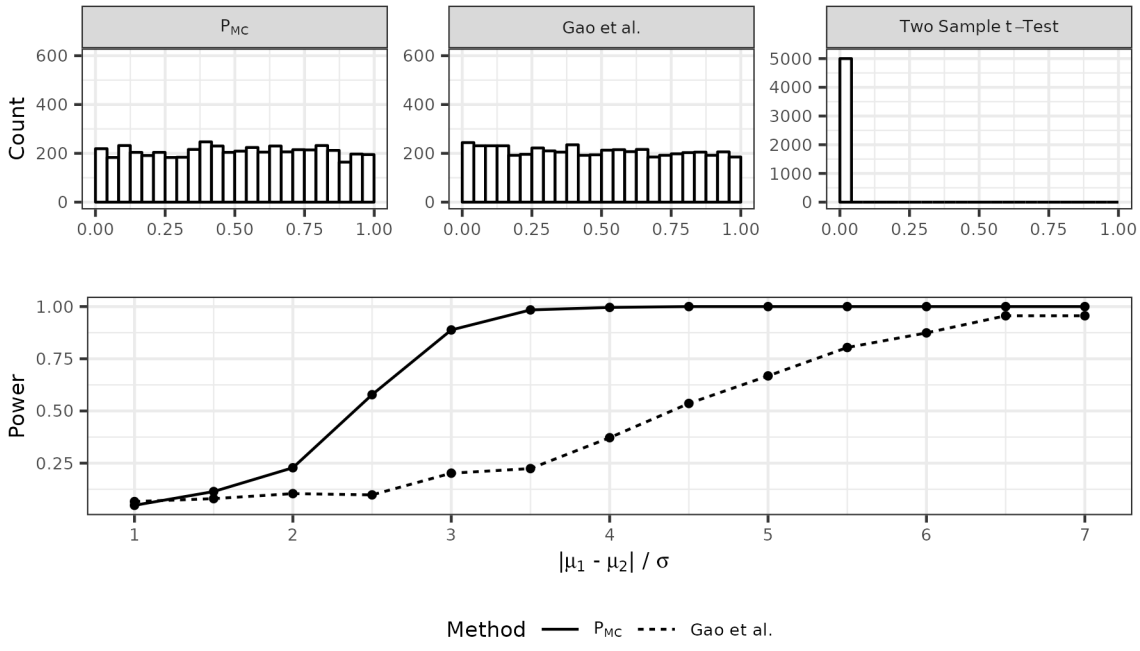


Figure 3: (*Top*) The distribution of p-values based on P_{mc}, Gao et al.'s selective inference procedure, and the two-sample t-test for 5,000 simulation replicates. (*Bottom*) Power comparison between P_{mc} and Gao et al.'s method to detect the presence of two Gaussian clusters controlling the type I error rate at level $\alpha = 0.05$ as the cluster separability increases. The power at each value $|\mu_1 - \mu_2|/\sigma$ is calculated based on 500 simulation replicates.

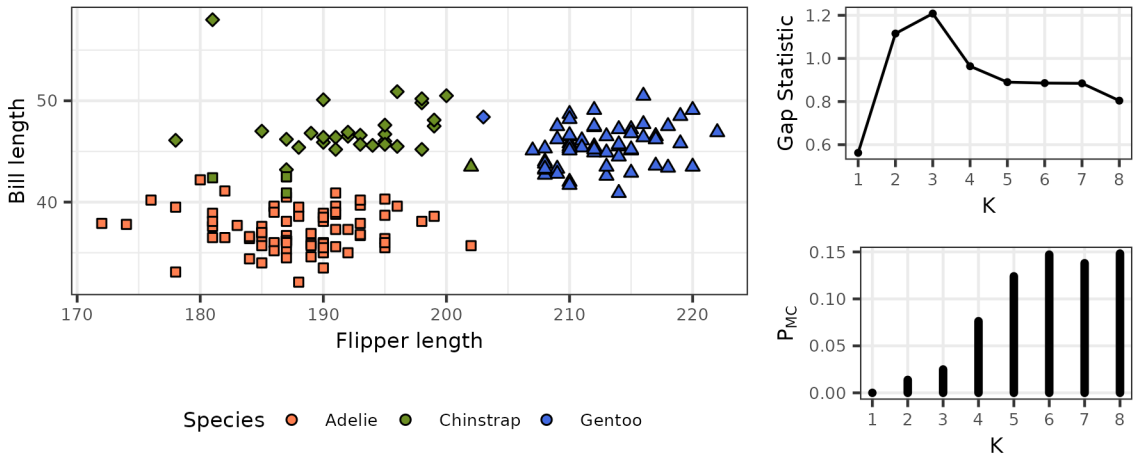


Figure 4: *(Left)* Bill and flipper lengths for the subset of female penguins with complete data from the un-scaled `palmerpenguins` data. Color indicates species, while shape indicates the assigned k-means cluster at $K = 3$. *(Right)* Value of the gap statistic and P_{mc} for different numbers of clusters based on the k-means clustering partition with Gaussian cluster distributions.

values are selected from the set $\{1, 1.5, \ldots, 7\}$. In each alternative scenario represented by a unique $|\mu_1 - \mu_2|/\sigma$ value, we draw 75 observations from each cluster distribution and compute the p-values using P_{mc} as well as Gao et al.'s procedure. We calculate the power based on 500 replicates for each alternative scenario. As can be seen in the bottom panel of Figure 3, P_{mc} exhibits higher power than Gao et al.'s procedure for moderate-to-large degrees of cluster separation (defined by $|\mu_1 - \mu_2|/\sigma \in [2, 6]$).

2.7 Real Data Applications

2.7.1 Cluster Analysis of Palmer Penguin Data

We analyze the `penguins` data from the `palmerpenguins` package [39], which consists of bill, flipper, and mass measurements from three species of penguins in the Palmer Archipelago between 2007 and 2009. Following the analysis by [16], we consider the subset of female penguins with complete data for bill and flipper length (both measured in millimeters), leaving us with 165 observations. Prior to performing the clustering, we center and scale the observations so that the measurements have zero mean and unit variance. Following the procedure laid out in Section 2.5, we perform k-means clustering of the scaled observations into $K = 1, \ldots, 8$ clusters and compute the gap statistic as well as P_{mc} for each grouping of the observations. Figure 4 shows the data used for clustering and the values of the gap statistic and P_{mc} for different values of K.

For the P_{mc} threshold of 0.05, the combined loss defined by the gap statistic and P_{mc} is optimized at $K = 3$. Both the visualization of the clustering data and the external species information suggest that the result is reasonable. We observe that the values of P_{mc} in this example are strongly negatively correlated with other cluster validity indices [24, 32, 33] to evaluate the k-means clustering partitions (Supplementary Table 2). We repeat this analysis using hierarchical clustering based on the squared Euclidean distance with the Ward linkage and come to the same conclusions (Supplementary Figure 3 and Supplementary Table 3).

2.7.2 Inferring Population Structure from HGDP data

In this illustration, we apply the PHM algorithm to perform cluster analysis on the genetic data from the Human Genome Diversity Project (HGDP) [40, 41], aiming to identify population structures. The dataset comprises 927 unrelated individuals sampled worldwide and genotyped at 2,543 autosomal SNPs. The geographic sampling locations are broadly divided into 7 continental groups: Europe, Central/South Asia (C/S Asia), Africa, Middle East, the Americas, East Asia, and Oceania.

Following the standard procedures for genetic data analysis, we pre-process the genotype matrix using principal component analysis (PCA) and select the first 5 PCs for our analysis based on the elbow point of the scree plot (Supplementary Figure 4). To apply the PHM algorithm, we fit a GMM to the dimension-reduced PC score matrix and select the model with 9 components based on the BIC. Each individual's posterior component assignment probability

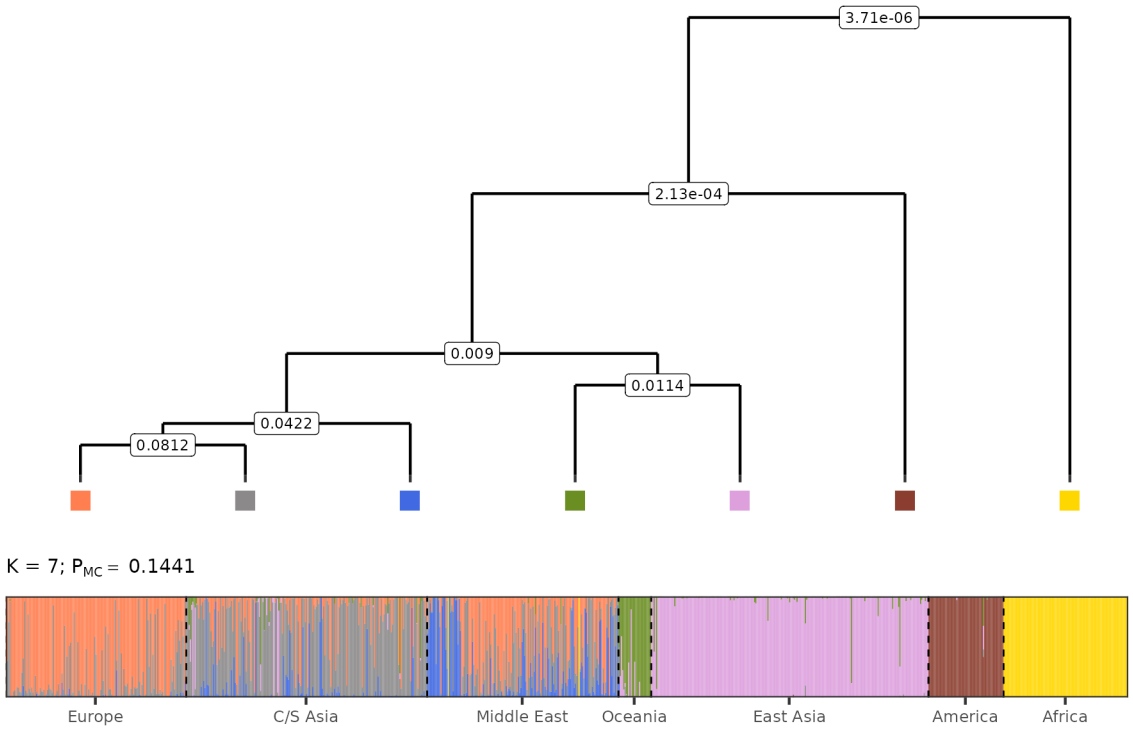


Figure 5: (*Top*) Dendrogram visualizing the order of component merges in the PHM procedure. Numeric values indicate the reduction of P_{mc} by the merge. Colors correspond to the mixture components from the distruct plot. (*Bottom*) Distruct plot for posterior component assignment probability, with color indicating mixture components. Observations are grouped by the geographic region of their sampling location.

π_k is shown in the distruct plot [42] in Figure 5. Except for the Europe, Central/South Asia, and Middle East groups, the remaining continental groups tend to correspond to unique mixture components.

Starting with assigning each mixture component to its own cluster and an initial $P_{\text{mc}} = 0.1437$, we proceed with the steps of the PHM algorithm with $\tau = 0$, i.e., merging until all observations belong to a single cluster and P_{mc} is decreased to 0. Figure 5 visualizes the merging process as a dendrogram. The numbers in each node indicate the reduction of P_{mc}, i.e., ΔP_{mc}, by combining the corresponding branches into a single cluster. A smaller value of ΔP_{mc} indicates that the clusters being combined are more distinct. Because the algorithm always prioritizes merging the most overlapping clusters, the merging sequence reflects the relative genetic dissimilarities between different clusters. This relationship can be straightforwardly interpreted from the dendrogram. For example, the first two merges — reducing P_{mc} to 0.063 and 0.021, respectively — form a cluster representing samples from Europe, the Middle East, and Central/South Asia, reflecting a close genetic relationship and noticeable genetic admixture among these population groups. The general structure of the dendrogram can be roughly explained by the likely path of historical human migrations: from Africa into the Middle East, from the Middle East to Europe and Central/South Asia, from Central/South Asia to East Asia, and from East Asia to Oceania and the Americas. The overall pattern of human genetic diversity among the continental groups identified from our analysis is also corroborated by more sophisticated genetic analysis using additional information (e.g., haplotype analysis) [41] and high-coverage genome sequencing data [43].

2.7.3 Cluster Analysis of Single-cell RNA Sequence Data

In this illustration, we apply the PHM algorithm to single-cell RNA sequencing (scRNA-seq) data from a sample of peripheral blood mononuclear cells to identify the different cell types. The data (sequenced on the Illumina NextSeq 500 and freely available from 10x Genomics) consist of gene expression counts for 2,700 single cells at 13,714 genes.

The raw sequence data are quality-controlled and pre-processed using standard procedures implemented in the `Seurat` package [44], leaving us with 2,638 cells. Principal component analysis is subsequently performed on the normalized and scaled expression counts for dimension reduction. The first 10 components are selected based on the elbow point

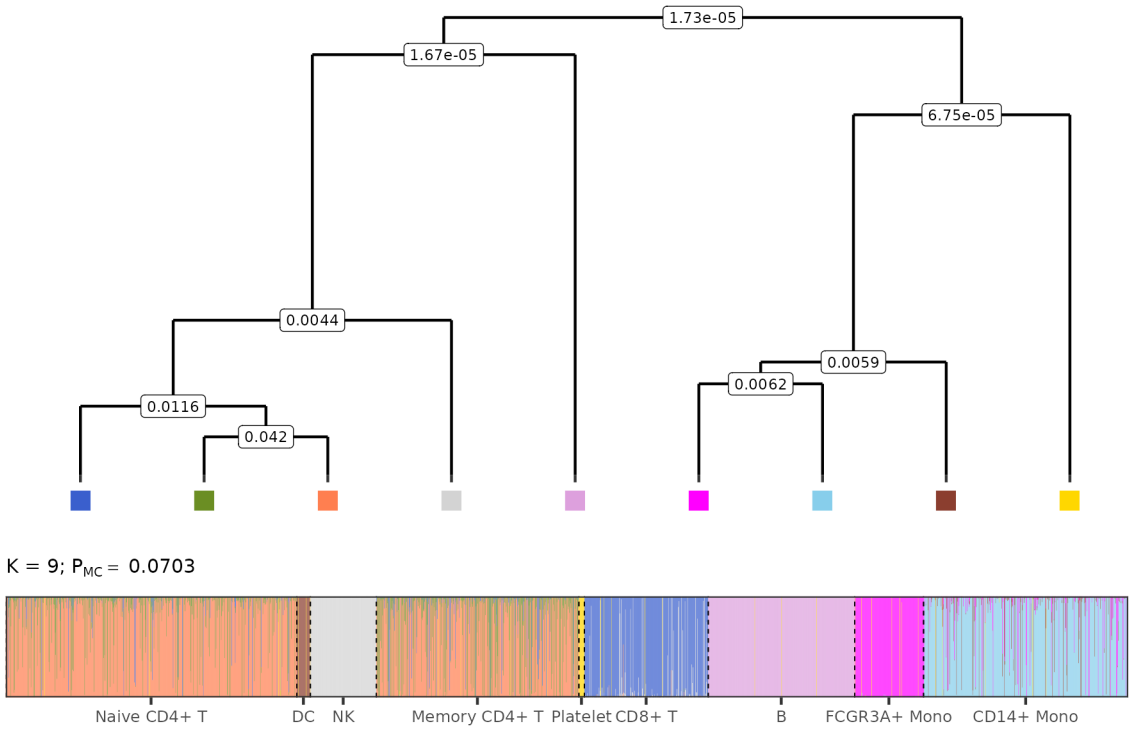


Figure 6: (*Top*) Dendrogram visualizing the order of component merges in the PHM procedure. Numeric values indicate the reduction of P_{mc} by the merge. Colors correspond to the mixture components from the distruct plot. (*Bottom*) Distruct plot for posterior component assignment probability, with color indicating mixture components. Observations are grouped by annotated cell type.

of the scree plot (Supplementary Figure 5). The resulting 2,638 $\times$ 10 data matrix is used for our cluster analysis. Additionally, each cell is annotated as one of the following cell types: Naïve CD4+ T cells, Memory CD4+ T cells, CD8+ T cells, Natural Killer (NK) cells, B cells, CD14+ monocytes, FCGR3A+ monocytes, Dendritic cells (DC), and Platelets, using known biomarkers. The annotated cell type information is not used in our cluster analysis procedure.

We fit a GMM on the dimension-reduced PC matrix and select the model with 9 components based on the optimal BIC. The posterior component assignment probability π_k for each cell is visualized in Figure 6. With the exception of the Naïve and Memory CD4+ T cells, each cell type corresponds to a unique mixture component.

Starting with each mixture component as its own cluster and an initial $P_{\text{mc}} = 0.0703$, we perform the merging procedure until all components have been combined into a single cluster. The merging process is represented by the dendrogram in Figure 6. We note that the pattern shown in the dendrogram has a striking similarity to the known immune cell differentiation trajectories. For example, the first merge combines the two types of CD4+ T cells, reducing P_{mc} from 0.070 to 0.042. Furthermore, CD8+ T cells, CD4+ T cells, NK cells, and B cells—all derived from Lymphoid progenitor cells—are grouped together in the dendrogram and separated from the branch consisting of FCGR4A+, CD14+ monocytes, and Dendritic cells—all of which are derived from Myeloid progenitor cells.

3 Conclusion and Discussion

In this work, we introduce the Distinguishability criterion, P_{mc}, to quantify the separability of clusters inferred from cluster analysis procedures. We discuss the intuition behind the criterion, as well as the derivation and properties of P_{mc}. We propose a combined loss function-based computational framework that integrates the Distinguishability criterion with available model and heuristics-based clustering algorithms and demonstrate its use with synthetic and real data applications.

The proposed P_{mc} is an internal clustering validity index to assess the separability of the clustering results, with unique advantages over alternative validity indices. Since P_{mc} is measured on the probability scale, the threshold for our

proposed constrained optimization problem is interpretable. Additionally, P_{mc} measurements are directly comparable across different datasets and various types of clustering applications, enabling future work on assessing the replicability of clustering analysis.

While our numerical illustrations primarily use Gaussian or mixtures of Gaussian distributions to evaluate P_{mc}, it is important to highlight the proposed computational framework's flexibility and compatibility with any valid parametric likelihood function. As a result, the applications of P_{mc} can be extended to a more diverse class of latent variable models, e.g., latent Dirichlet allocation (LDA) and generalized factor analysis models, in order to help address the similar model selection problems that arise with their use. We will explore these extensions in our future work.

The properties of P_{mc} are best utilized in our proposed PHM algorithm, which is motivated by Baudry et al.'s entropy criterion-based merging procedure [30]. By combining mixture components into clusters, both algorithms enjoy excellent model fit and interpretability for the inferred clusters. However, the constrained optimization formulation described above leads to a more interpretable stopping rule for the PHM algorithm compared to Baudry et al.'s procedure. Furthermore, the PHM algorithm shows improved computational efficiency—due to the cluster merging property of P_{mc}, the complexity of the PHM algorithm is invariant to the sample size of the observed clustering data, making it more suitable for analyzing large-scale data. Finally, we note that the dendrogram visualizing the complete merging procedure can be used to represent a coalescent process that has many applications across scientific disciplines, such as developmental biology, human genetics, and evolutionary biology. One possible future research direction is shifting the focus from cluster analysis to uncovering the underlying coalescent trees by incorporating more context-specific information, with the PHM algorithm as a natural starting point.

It is also possible to extend the PHM algorithm to work with a general class of hard clustering algorithms that output optimal partitions of observed data. One strategy is to treat each output partition as a distinct population sample and estimate its distribution using a finite mixture model. We can then re-normalize the mixture proportions globally and initilize the clusters at the level of the output partitions, at which point applying the PHM algorithm is straightforward. This simple strategy extends the applications of the Distinguishability criterion to a more diverse class of clustering algorithms, including density-based and graph clustering algorithms.

4 Methods

4.1 Overview of Clustering Methods

We briefly review existing clustering methods, focusing on the algorithms used in this paper. The available clustering methods can be roughly classified into two categories: heuristics-based clustering methods, represented by the k-means and hierarchical clustering algorithms, and model-based clustering methods, represented by finite-mixture models. The heuristics-based clustering methods typically do not make explicit distributional assumptions and instead perform hard clustering by outputting optimal partitions of the observed data based on their corresponding objective functions. The model-based clustering methods perform formal statistical inference of the latent cluster structures underlying the observed data.

4.1.1 k-means clustering

k-means clustering is a widely used heuristics-based clustering algorithm for partitioning a sample into clusters. As the name suggests, this procedure identifies clusters by their centroids (defined as the mean of all points in the cluster) and assigns observations to the cluster with the nearest centroid. More formally, for a specified number of clusters K, the algorithm groups the observations into disjoint sets $\mathcal{C}_1, \ldots, \mathcal{C}_K$ to minimize the distortion function, i.e.,

$$\min_{\mathcal{C}_1,\ldots,\mathcal{C}_K} \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \left|\left| \boldsymbol{x}_i - \sum_{j \in \mathcal{C}_k} \boldsymbol{x}_j / |\mathcal{C}_k| \right|\right|_2^2,$$

which corresponds to minimizing the within-cluster variances. The clustering is usually performed using Lloyd's algorithm [45], which alternates between assigning observations to the cluster with the nearest centroid and updating the cluster centroid to eventually arrive at a locally optimal solution.

It has been shown that the k-means algorithm is equivalent to an approximate maximum likelihood procedure, where the underlying probability model assumes a multivariate Gaussian distribution for each latent cluster [13, 14]. Hence, the observation frequency of each cluster and the corresponding parameters for the Gaussian distribution can be estimated straightforwardly using the partitioned output from the k-means algorithm.

4.1.2 Hierarchical clustering

Hierarchical clustering produces a sequence of partitions for observations. In the commonly used agglomerative hierarchical clustering procedure, each observation comprises its own cluster in the initial partition. Subsequent clusterings in the sequence are formed by repeatedly combining the most similar pair of groups of observations into a single group until all observations have been combined into a single cluster. The similarity between groups of observations in hierarchical clustering is typically defined by a distance measure between pairs of observations, such as the squared Euclidean distance, in addition to a function that generalizes this similarity to groups of observations (referred to as a linkage function). Commonly used linkage functions between groups of observations are single linkage, complete linkage, average linkage, and the Ward linkage.

The hierarchical clustering algorithm is also often connected to probability models assuming a Gaussian data distribution underlying each cluster [46, 14], and explicit Gaussian assumptions are commonly used for model-selection or post-selection inference in hierarchical clustering [15, 16, 18].

4.1.3 Model-based Clustering

Finite mixture models are the most representative approach for model-based clustering, where a mixture distribution with finite components models the observed clustering data. Traditionally, each mixture component is taken to correspond to a homogeneous subpopulation or cluster. The goal of inference in the mixture model setting is to estimate the characteristics of each mixture component, i.e., $p(\boldsymbol{x} \mid \theta = k)$, and the corresponding mixture proportion, i.e., α_k. The expectation-maximization (EM) algorithm is commonly used to find maximum likelihood estimates of the parameters of interest. The Gaussian mixture model (GMM), which models each mixture component using a unique Gaussian distribution, is probably the most commonly used mixture model in practice. This is because, among other reasons, the GMM is considered to be a universal approximator [47] and is flexible enough to fit diverse data types.

Unlike most heuristics-based clustering approaches, model-based clustering algorithms perform soft (or fuzzy) clustering, as they do not directly partition the observed data. Instead, every data point has an associated (posterior) probability distribution over all possible cluster assignments. A post-hoc classification procedure, with a pre-specified decision rule using the cluster assignment probabilities, can be applied to partition the observed data.

For a more thorough review of model-based clustering, we refer the reader to [14, 48, 49]

4.2 Derivation and Estimation of P_{mc}

The Bayes risk for a general classifier $\delta(\boldsymbol{x}) : \mathbb{R}^p \mapsto \{1, \ldots, K\}$ assigning a point $\boldsymbol{x}$ to one of the K clusters can be derived as follows,

$$\begin{aligned} P_{\text{mc}} &= \mathbb{E}_{\boldsymbol{x}}\Big[\mathbb{E}_{\theta}\big[L(\delta(\boldsymbol{x}), \theta) \mid \boldsymbol{x}\big]\Big] \\ &= \int \left(\sum_{j=1}^{K} \sum_{i \neq j} \pi_i(\boldsymbol{x}) \, \Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x}) \right) P(d\boldsymbol{x}) \\ &= \int \left(\sum_{j=1}^{K} (1 - \pi_j(\boldsymbol{x})) \, \Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x}) \right) P(d\boldsymbol{x}) \end{aligned} \tag{6}$$

For a more detailed derivation, see Appendix A. The marginal data distribution $P(\boldsymbol{x})$, with loose notation, is given by

$$P(\boldsymbol{x}) = \sum_{k=1}^{K} \Pr(\theta = k) \, p(\boldsymbol{x} \mid \theta = k) = \sum_{k=1}^{K} \alpha_k \, p(\boldsymbol{x} \mid \theta = k).$$

For the default randomized decision rule, $\delta_r(\boldsymbol{x}) \sim \text{Categorical}\,(\boldsymbol{\pi}(\boldsymbol{x}))$,

$$\Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x}) = \pi_j(\boldsymbol{x}). \tag{7}$$

Thus,

$$
\begin{aligned}
P_{\mathrm{mc},\,\delta_r} &= \int \left(\sum_{j=1}^{K} \pi_j(\boldsymbol{x})(1-\pi_j(\boldsymbol{x})) \right) P(d\boldsymbol{x}) \\
&= 2 \sum_{i<j} \int \pi_i(\boldsymbol{x}) \pi_j(\boldsymbol{x}) P(d\boldsymbol{x})
\end{aligned}
\tag{8}
$$

To compute P_{mc} using the optimal decision rule, $\delta_o(\boldsymbol{x}) = \arg\max_k \pi_k(\boldsymbol{x})$, we define a partition of the sample space, $\cup_{k=1}^{K} \mathcal{R}_k$, such that $\mathcal{R}_k := \{\boldsymbol{x} : \delta_o(\boldsymbol{x}) = k\}$. It follows that,

$$
\Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x}) = \mathbf{1}\{\boldsymbol{x} \in \mathcal{R}_j\}, \tag{9}
$$

and

$$
\begin{aligned}
P_{\mathrm{mc},\,\delta_o} &= \int \left(\sum_{j=1}^{K} \sum_{i \neq j} \pi_i(\boldsymbol{x})\, \mathbf{1}\{\boldsymbol{x} \in \mathcal{R}_j\} \right) P(d\boldsymbol{x}) \\
&= \int \left(1 - \max_k \pi_k(\boldsymbol{x}) \right) P(d\boldsymbol{x})
\end{aligned}
\tag{10}
$$

Note that,

$$
\sum_{j=1}^{K} \pi_j(\boldsymbol{x})(1-\pi_j(\boldsymbol{x})) \geq \sum_{j=1}^{k} \pi_j(\boldsymbol{x})(1-\max_k \pi_k(\boldsymbol{x})) = 1 - \max_k \pi_k(\boldsymbol{x}),\ \forall \boldsymbol{x} \tag{11}
$$

Hence,

$$
P_{\mathrm{mc},\,\delta_r} \geq P_{\mathrm{mc},\,\delta_o}
$$

Unless otherwise specificied, we use the notation P_{mc} to refer to $P_{\mathrm{mc},\,\delta_r}$ by default. In this paper, we estimate P_{mc} by plugging in the point estimates of the α_k's as well as the key distributional parameters in the corresponding likelihood functions obtained from the observed data.

4.3 Lower and Upper Bounds of P_{mc}

When all clusters are well-separated, P_{mc} approaches its lower bound at 0. More specifically, assuming well-separated clusters, both of the following conditions should hold:

$$
\pi_i(\boldsymbol{x})\, \pi_j(\boldsymbol{x}) \to 0,\ \forall\, \boldsymbol{x} \text{ and } (i,j) \text{ pairs},
$$

and

$$
\max_k \pi_k(\boldsymbol{x}) \to 1,\ \forall\, \boldsymbol{x}.
$$

Hence, both decision rules (δ_r and δ_o) approach perfect classification accuracy.

P_{mc} is maximized when all clusters are completely overlapping, i.e.,

$$
p(\boldsymbol{x} \mid \theta = i) = p(\boldsymbol{x} \mid \theta = j)\ \forall\, \boldsymbol{x} \text{ and } (i,j) \text{ pairs}. \tag{12}
$$

Thus, $\pi_k(x) = \alpha_k,\ \forall \boldsymbol{x}$. It follows that

$$
\max P_{\mathrm{mc},\,\delta_r} = \sum_{k=1}^{K} \alpha_k\,(1-\alpha_k),
$$

and

$$\max P_{\mathrm{mc},\,\delta_o} = 1 - \max_k \alpha_k.$$

In the special case that $\alpha_k = 1/K$ for all k values,

$$\max P_{\mathrm{mc},\,\delta_r} = \max P_{\mathrm{mc},\,\delta_o} = \frac{K-1}{K}. \tag{13}$$

4.4 The Cluster Merging Property of P_{mc}

The merging property is specific to the default P_{mc}, evaluated using the randomized decision rule δ_r. For a given cluster configuration with $K \geq 2$, consider merging two arbitrary clusters to form a new combined cluster. Let P_{mc} and $P_{\mathrm{mc}}^{\dagger}$ denote the misclassification probabilities before and after the merging, respectively. The following proposition summarizes the merging property.

Proposition 1. *Merging two existing clusters indexed by i and j leads to*

$$\Delta P_{\mathrm{mc}}^{(i,j)} := P_{\mathrm{mc}} - P_{\mathrm{mc}}^{\dagger} = 2 \int \pi_i(\boldsymbol{x})\pi_j(\boldsymbol{x})\, P(d\boldsymbol{x}) \geq 0.$$

Furthermore,

$$P_{\mathrm{mc}} = \sum_{i<j} \Delta P_{\mathrm{mc}}^{(i,j)}$$

Proof. Appendix B. □

The cluster merging property forms the basis of the PHM algorithm. As $\Delta P_{\mathrm{mc}}^{(i,j)}$ can be pre-computed for all pairs of clusters from the initial configuration, the subsequent updates for P_{mc}—merging one pair of clusters at a time—become straightforward to compute.

4.5 Numerical Evaluation of P_{mc}

Evaluating P_{mc} numerically can be challenging, especially when clustering data are high-dimensional. For low-dimensional data, it is possible to evaluate Eqn (6) by numerical integration using various quadrature methods. However, they generally do not scale well when the clustering data dimensionality becomes larger than 5. When the marginal data distribution, $P(\boldsymbol{x})$, can be directly sampled from (as in the case in all examples presented in this paper), Monte Carlo (MC) integration becomes an efficient solution. Specifically, we sample M data points from $P(\boldsymbol{x})$ and approximate P_{mc} by

$$\hat{P}_{\mathrm{mc}} = \frac{1}{M} \sum_{i=1}^{M} \left(\sum_{j=1}^{K} \Big(1 - \pi_j(\boldsymbol{x}_i)\Big)\, \Pr(\delta(\boldsymbol{x}_i) = j \mid \boldsymbol{x}) \right) \tag{14}$$

The unique advantage of the Monte Carlo integration method is that its error bound is always $O(1/\sqrt{M})$ regardless of the dimensionality of $\boldsymbol{x}$. We provide comparisons between the Monte Carlo method and the numerical integration for evaluating P_{mc} in some low-dimensional settings (Supplementary Method 1 and Supplementary Table 4), indicating that the MC integration method is accurate and efficient.

References

[1] Elke Braun, Bart Geurten, and Martin Egelhaaf. Identifying prototypical components in behaviour using clustering algorithms. *PloS one*, 5(2):e9361, 2010.

[2] S Wibisono, MT Anwar, Aji Supriyanto, and IHA Amin. Multivariate weather anomaly detection using dbscan clustering algorithm. In *Journal of Physics: Conference Series*, volume 1869, page 012077. IOP Publishing, 2021.

[3] Parvez Ahmad, Saqib Qamar, and Syed Qasim Afser Rizvi. Techniques of data mining in healthcare: a review. *International Journal of Computer Applications*, 120(15), 2015.

[4] Jui-Hung Kao, Ta-Chien Chan, Feipei Lai, Bo-Cheng Lin, Wei-Zen Sun, Kuan-Wu Chang, Fang-Yie Leu, and Jeng-Wei Lin. Spatial analysis and data mining techniques for identifying risk factors of out-of-hospital cardiac arrest. *International Journal of Information Management*, 37(1):1528–1538, 2017.

[5] Sarah Shafqat, Saira Kishwer, Raihan Ur Rasool, Junaid Qadir, Tehmina Amjad, and Hafiz Farooq Ahmad. Big data analytics enhanced healthcare systems: a review. *The Journal of Supercomputing*, 76:1754–1799, 2020.

[6] Juan Xie, Anjun Ma, Yu Zhang, Bingqiang Liu, Changlin Wan, Sha Cao, Chi Zhang, and Qin Ma. Qubic2: a novel biclustering algorithm for large-scale bulk rna-sequencing and single-cell rna-sequencing data analysis. *bioRxiv*, page 409961, 2018.

[7] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

[8] Itamar Kanter, Piero Dalerba, and Tomer Kalisky. A cluster robustness score for identifying cell subpopulations in single cell gene expression datasets from heterogeneous tissues and tumors. *Bioinformatics*, 35(6):962–971, 2019.

[9] Ren Qi, Anjun Ma, Qin Ma, and Quan Zou. Clustering and classification methods for single-cell rna-sequencing data. *Briefings in bioinformatics*, 21(4):1196–1208, 2020.

[10] John Harmon Wolfe. *Object cluster analysis of social areas*. PhD thesis, University of California, 1963.

[11] Richard M Cormack. A review of classification. *Journal of the Royal Statistical Society: Series A (General)*, 134(3):321–353, 1971.

[12] Gbeminiyi John Oyewole and George Alex Thopil. Data clustering: application and trends. *Artificial Intelligence Review*, 56(7):6439–6475, 2023.

[13] Hans H Bock. Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis*, 23(1):5–28, 1996.

[14] Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, June 2002.

[15] Patrick K. Kimes, Yufeng Liu, David Neil Hayes, and James Stephen Marron. Statistical significance for hierarchical clustering. *Biometrics*, 73(3):811–821, January 2017.

[16] Lucy L Gao, Jacob Bien, and Daniela Witten. Selective inference for hierarchical clustering. *Journal of the American Statistical Association*, pages 1–11, 2022.

[17] Yiqun T Chen and Daniela M Witten. Selective inference for k-means clustering. *arXiv preprint arXiv:2203.15267*, 2022.

[18] Isabella N. Grabski, Kelly Street, and Rafael A. Irizarry. Significance analysis for clustering with single-cell rna-sequencing data. *Nature Methods*, 20(8):1196–1202, July 2023.

[19] Christian Hennig. What are the true clusters? *Pattern Recognition Letters*, 64:53–62, October 2015.

[20] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[21] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of intelligent information systems*, 17:107–145, 2001.

[22] Minho Kim and RS Ramakrishna. New indices for cluster validity assessment. *Pattern Recognition Letters*, 26(15):2353–2363, 2005.

[23] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *2010 IEEE international conference on data mining*, pages 911–916. IEEE, 2010.

[24] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[25] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[26] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[27] Volodymyr Melnykov. Merging mixture components for clustering through pairwise overlap. *Journal of Computational and Graphical Statistics*, 25(1):66–90, January 2016.

[28] Gilles Celeux and Gilda Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification*, 13(2):195–212, September 1996.

[29] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, July 2000.

[30] Jean-Patrick Baudry, Adrian E. Raftery, Gilles Celeux, Kenneth Lo, and Raphaël Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2):332–353, January 2010.

[31] Ulrike Von Luxburg et al. Clustering stability: an overview. *Foundations and Trends® in Machine Learning*, 2(3):235–274, 2010.

[32] Tilman Lange, Volker Roth, Mikio L Braun, and Joachim M Buhmann. Stability-based validation of clustering solutions. *Neural computation*, 16(6):1299–1323, 2004.

[33] Robert Tibshirani and Guenther Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.

[34] Christian Hennig. Methods for merging gaussian mixture components. *Advances in Data Analysis and Classification*, 4(1):3–34, January 2010.

[35] Luca Scrucca, Michael Fop, T. Brendan Murphy, and Adrian E. Raftery. mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):289–317, 2016.

[36] Lampros Mouselimis. *ClusterR: Gaussian Mixture Models, K-Means, Mini-Batch-Kmeans, K-Medoids and Affinity Propagation Clustering*, 2023. R package version 1.3.1.

[37] David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pages 1027–1035, 2007.

[38] David V Hinkley. Bootstrap methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 50(3):321–337, 1988.

[39] Allison Marie Horst, Alison Presmanes Hill, and Kristen B Gorman. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*, 2020. R package version 0.1.0.

[40] L Luca Cavalli-Sforza. The human genome diversity project: past, present and future. *Nature Reviews Genetics*, 6(4):333–340, 2005.

[41] Donald F Conrad, Mattias Jakobsson, Graham Coop, Xiaoquan Wen, Jeffrey D Wall, Noah A Rosenberg, and Jonathan K Pritchard. A worldwide survey of haplotype variation and linkage disequilibrium in the human genome. *Nature genetics*, 38(11):1251–1260, 2006.

[42] Noah A Rosenberg. Distruct: a program for the graphical display of population structure. *Molecular ecology notes*, 4(1):137–138, 2004.

[43] Anders Bergström, Shane A McCarthy, Ruoyun Hui, Mohamed A Almarri, Qasim Ayub, Petr Danecek, Yuan Chen, Sabine Felkel, Pille Hallast, Jack Kamm, et al. Insights into human genetic variation and population history from 929 diverse genomes. *Science*, 367(6484):eaay5012, 2020.

[44] Yuhan Hao, Tim Stuart, Madeline H Kowalski, Saket Choudhary, Paul Hoffman, Austin Hartman, Avi Srivastava, Gesmira Molla, Shaista Madad, Carlos Fernandez-Granda, and Rahul Satija. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*, 2023.

[45] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[46] Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

[47] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[48] Paul D McNicholas. Model-based clustering. *Journal of Classification*, 33:331–373, 2016.

[49] Isobel Claire Gormley, Thomas Brendan Murphy, and Adrian E Raftery. Model-based clustering. *Annual Review of Statistics and Its Application*, 10:573–595, 2023.

Appendix A Derivation of P_{mc}

To compute the Bayes risk for a general classifier $\delta(\boldsymbol{x})$ under the 0-1 loss, we first evaluate its posterior expected loss $\mathbb{E}_\theta\big[L(\delta(\boldsymbol{x}),\theta) \mid \boldsymbol{x}\big]$ as follows:

$$\begin{aligned}
\mathbb{E}_\theta\big[L(\delta(\boldsymbol{x}),\theta) \mid \boldsymbol{x}\big] &= \Pr(\theta \neq \delta(\boldsymbol{x}) \mid \boldsymbol{x}) \\
&= \sum_{j=1}^{K} \Pr(\delta(\boldsymbol{x}) = j,\, \theta \neq j \mid \boldsymbol{x}) \\
&= \sum_{j=1}^{K} \Pr(\theta \neq j \mid \boldsymbol{x})\, \Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x}) \\
&= \sum_{j=1}^{K} \sum_{i \neq j} \Pr(\theta = i \mid \boldsymbol{x})\, \Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x}) \\
&= \sum_{j=1}^{k} \sum_{i \neq j} \pi_i(\boldsymbol{x})\, \Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x}) \\
&= \sum_{j=1}^{k} (1 - \pi_j(\boldsymbol{x}))\, \Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x})
\end{aligned}$$

Note that $\Pr(\theta \neq j \mid \boldsymbol{x}, \delta(\boldsymbol{x})) = \Pr(\theta \neq j \mid \boldsymbol{x})$. Subsequently,

$$\begin{aligned}
P_{\mathrm{mc}} &= \mathbb{E}_{\boldsymbol{x}}\Big[\mathbb{E}_\theta\big[L(\delta(\boldsymbol{x}),\theta) \mid \boldsymbol{x}\big]\Big] \\
&= \int \left(\sum_{j=1}^{K} \sum_{i \neq j} \pi_i(\boldsymbol{x})\, \Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x}) \right) P(d\boldsymbol{x}) \\
&= \int \left(\sum_{j=1}^{K} (1 - \pi_j(\boldsymbol{x}))\, \Pr(\delta(\boldsymbol{x}) = j \mid \boldsymbol{x}) \right) P(d\boldsymbol{x})
\end{aligned}$$

Appendix B Proof of Proposition 1

Proof. Consider merging two existing clusters i, j to a new combined cluster k'. It follows that

$$\alpha_{k'} = \alpha_i + \alpha_j$$

and

$$p(\boldsymbol{x} \mid \theta = k) = \frac{\alpha_i\, p(\boldsymbol{x} \mid \theta = i) + \alpha_j\, p(\boldsymbol{x} \mid \theta = j)}{\alpha_{k'}}$$

Consequently, by applying Bayes rule,

$$\pi_{k'}(\boldsymbol{x}) = \pi_i(\boldsymbol{x}) + \pi_j(\boldsymbol{x}) \tag{15}$$

Let S denote the set of indices of the existing clusters not impacted by the merge, where $|S| = K - 2$. By Eqn (8), P_{mc} can be written as

$$
\begin{aligned}
P_{\mathrm{mc}} = 2 & \sum_{m,n \in S,\, m<n} \int \pi_m(\boldsymbol{x}) \pi_n(\boldsymbol{x}) P(d\boldsymbol{x}) \\
& + 2 \sum_{l \in S} \int \pi_l(\boldsymbol{x}) \pi_i(\boldsymbol{x}) P(d\boldsymbol{x}) + 2 \sum_{l \in S} \int \pi_l(\boldsymbol{x}) \pi_j(\boldsymbol{x}) P(d\boldsymbol{x}) \\
& + 2 \int \pi_i(\boldsymbol{x}) \pi_j(\boldsymbol{x}) P(d\boldsymbol{x})
\end{aligned}
$$

By Eqn (15),

$$
2 \sum_{l \in S} \int \pi_l(\boldsymbol{x}) \pi_i(\boldsymbol{x}) P(d\boldsymbol{x}) + 2 \sum_{l \in S} \int \pi_l(\boldsymbol{x}) \pi_j(\boldsymbol{x}) P(d\boldsymbol{x}) = 2 \sum_{l \in S} \int \pi_l(\boldsymbol{x}) \pi_{k'}(\boldsymbol{x}) P(d\boldsymbol{x})
$$

and note that,

$$
P^{\dagger}_{\mathrm{mc}} = 2 \sum_{m,n \in S,\, m<n} \int \pi_m(\boldsymbol{x}) \pi_n(\boldsymbol{x}) P(d\boldsymbol{x}) + 2 \sum_{l \in S} \int \pi_l(\boldsymbol{x}) \pi_{k'}(\boldsymbol{x}) P(d\boldsymbol{x})
$$

It becomes evident that

$$
\Delta P^{(i,j)}_{\mathrm{mc}} = P_{\mathrm{mc}} - P^{\dagger}_{\mathrm{mc}} = 2 \int \pi_i(\boldsymbol{x}) \pi_j(\boldsymbol{x}) P(d\boldsymbol{x}) \geq 0 \tag{16}
$$

Plugging in the expression of $\Delta P^{(i,j)}_{\mathrm{mc}}$ into Eqn (8) yields

$$
P_{\mathrm{mc}} = \sum_{i<j} \Delta P^{(i,j)}_{\mathrm{mc}} \tag{17}
$$

□

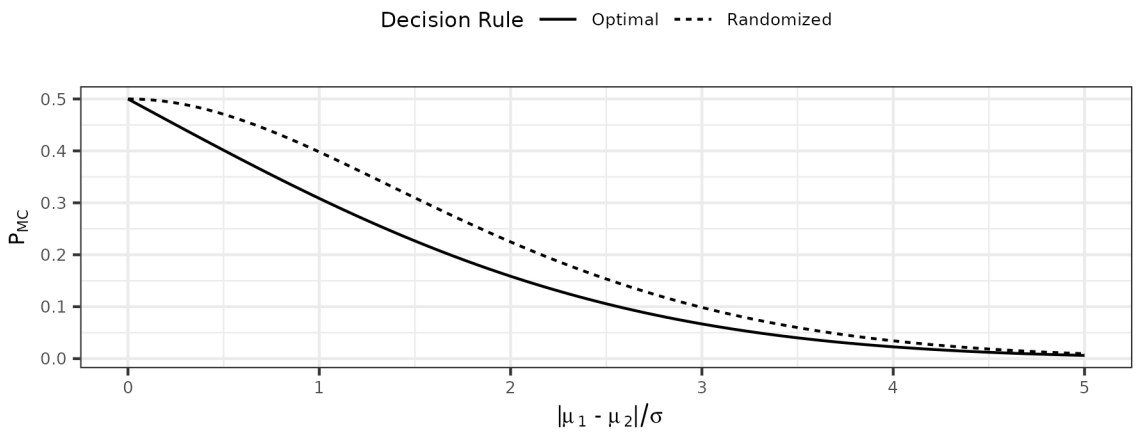
Remark The merging property is specific to the randomized decision rule δ_r under the 0-1 loss. For the optimal decision rule, δ_o, it can be shown that $P^{\dagger}_{\mathrm{mc}} \leq P_{\mathrm{mc}}$ after merging a pair of existing clusters. However, the quantitive expression for $\Delta P^{(i,j)}_{\mathrm{mc}}$ is analytically intractable.

Appendix C P_{mc} Dendrogram Construction

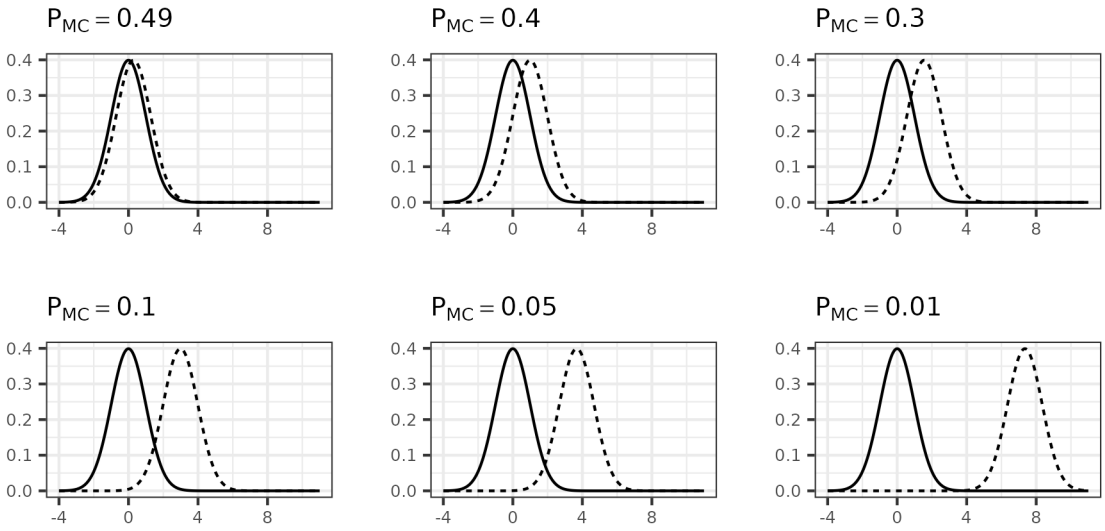
We use a dendrogram to visualize the PHM procedure described in Algorithm 1. The leaf nodes in the tree correspond to the individual components of the mixture model (MM) used to fit the data. The edges between parent and child nodes correspond to a cluster merge step; we present the ΔP_{mc} reduction from the merge on the parent node for the merged clusters. In this way, it is possible to determine the value of P_{mc} after each successive merge by subtracting the cumulative ΔP_{mc} values from the leaf nodes of the tree up to the height of a specific merge from the P_{mc} of the initial cluster configuration.

The height of a merge in the tree is determined as follows. Let P^{0}_{mc} denote the P_{mc} value at the initial cluster configuration and, for a given merge, let $P^{\ddagger}_{\mathrm{mc}}$ be the value of the criterion *prior* to that merge taking place. We place the merge in the dendrogram at a height corresponding to the $\log_{10}$ scaled ratio of these values, i.e., $\log_{10} P^{0}_{\mathrm{mc}} / P^{\ddagger}_{\mathrm{mc}}$. The $\log_{10}$ transformation prevents the merges from early on in the procedure from being "squashed" to the bottom of the tree. We opt to use the P_{mc} value before the merge occurs rather than after to avoid dividing by zero when calculating the height of the final merge (which would result in $P_{\mathrm{mc}} = 0$).

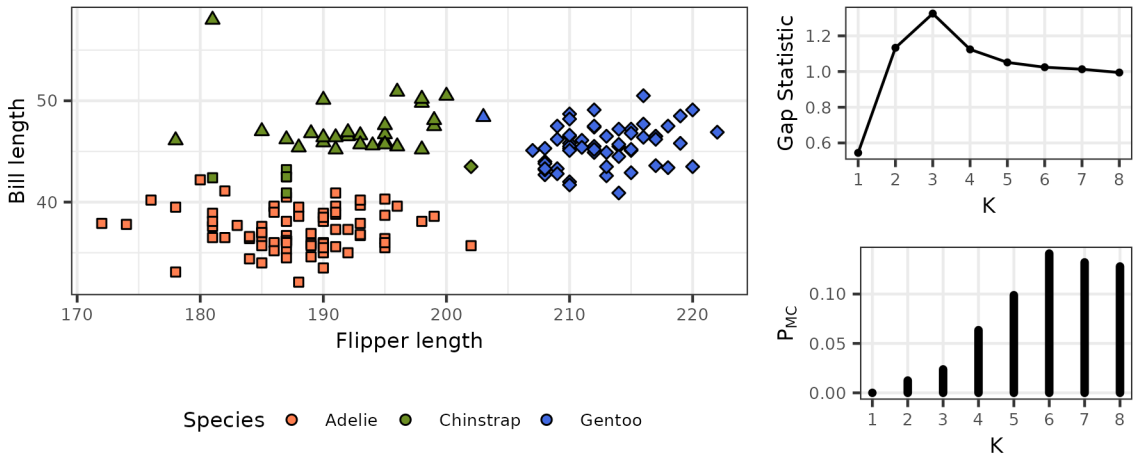
Supplementary Figures



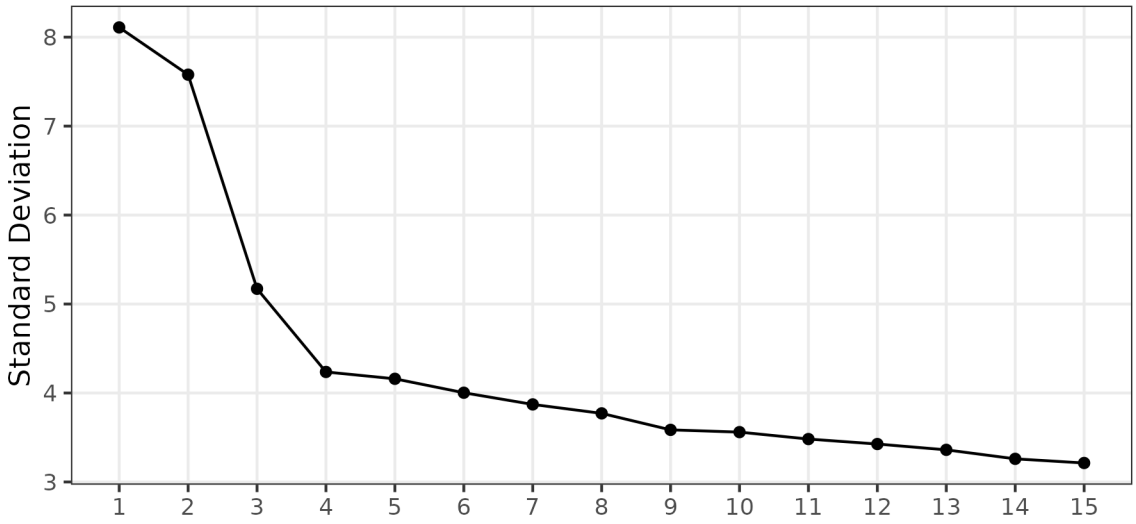
Supplementary Figure 1: Values of P_{mc} based on the randomized and optimal decision rules δ_r and δ_o. The value P_{mc} is shown in the y-axis and is calculated for two univariate Gaussian distributions $N(\mu_1, \sigma)$ and $N(\mu_2, \sigma)$ where $\pi_1 = \pi_2 = 0.5$. The x-axis indicates the degree of cluster separation in terms of the distribution parameters.



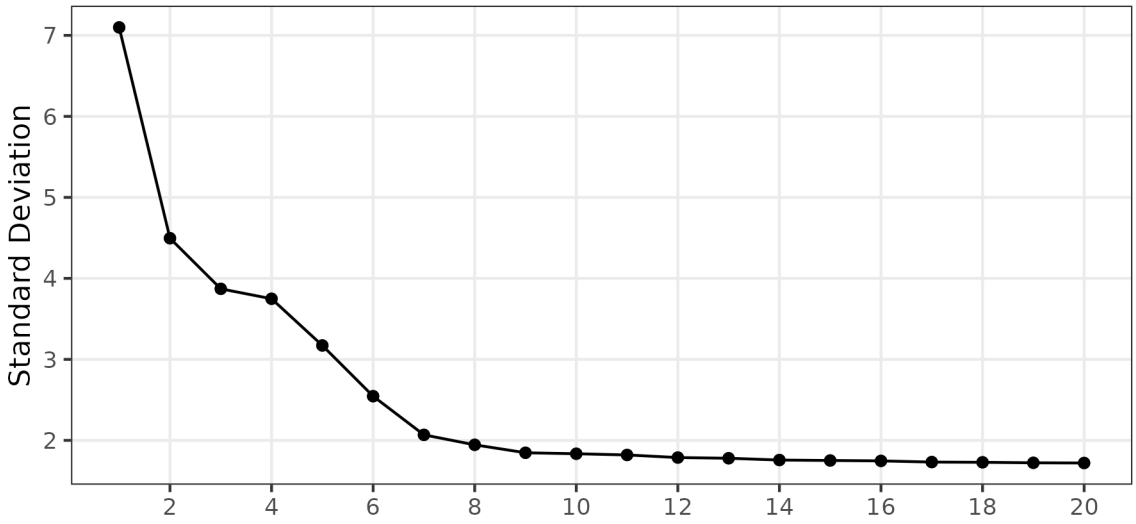
Supplementary Figure 2: Distribution plots for two univariate Gaussian distributions $N(0, 1)$ (solid line) and $N(\mu, 1)$ (dashed line) at decreasing values of P_{mc}, where $\pi_1 = \pi_2 = 0.5$. The distance between the two centroids, $|\mu|$, determines the specific P_{mc} value.



Supplementary Figure 3: (*Left*) Bill and flipper lengths for the subset of `palmerpenguins` data. Color indicates species while shape indicates the assigned hierarchical clustering partition. (*Right*) Value of the gap statistic and P_{mc} based on hierarchical clustering for different numbers of clusters with Gaussian cluster distributions.



Supplementary Figure 4: Scree plot visualizing standard deviation captured by each of of the principal component vectors from the HGDP data.



Supplementary Figure 5: Scree plot visualizing standard deviation captured by each of of the principal component vectors from the scRNA-seq data.

Supplementary Tables

K	Gap	P_{mc}	Silhouette	Stability (ARI)	Prediction Strength
1	0.403	0.000	—	—	—
2	0.824	0.002	0.632	0.995	1.000
3	1.058	0.062	0.522	0.961	0.842
4	0.901	0.101	0.417	0.758	0.618
5	0.808	0.126	0.353	0.723	0.519
6	0.722	0.137	0.353	0.630	0.679
7	0.740	0.158	0.348	0.619	0.399

Supplementary Table 1: Values of P_{mc} and other cluster validity indices for the simulated k-means example data.

K	Gap	P_{mc}	Silhouette	Stability (ARI)	Prediction Strength
1	0.562	0.000	—	—	—
2	1.115	0.014	0.583	0.957	0.902
3	1.208	0.025	0.595	0.947	0.897
4	0.964	0.076	0.468	0.757	0.458
5	0.890	0.124	0.372	0.679	0.470
6	0.886	0.147	0.385	0.646	0.362
7	0.884	0.138	0.390	0.641	0.333
8	0.804	0.148	0.369	0.627	0.383

Supplementary Table 2: Values of P_{mc} and other cluster validity indices for the Palmer penguins data based on the k-means clusterings of the observations.

K	Gap	P_{mc}	Silhouette	Stability (ARI)	Prediction Strength
1	0.545	0.000	—	—	—
2	1.133	0.012	0.581	0.909	0.942
3	1.325	0.024	0.595	0.943	0.899
4	1.124	0.063	0.483	0.757	0.544
5	1.051	0.099	0.470	0.702	0.464
6	1.024	0.141	0.382	0.652	0.413
7	1.013	0.132	0.387	0.656	0.366
8	0.994	0.128	0.381	0.645	0.345

Supplementary Table 3: Values of P_{mc} and other cluster validity indices for the Palmer penguins data based on the hierarchical clusterings of the observations.

p	P_{mc}	Elapsed (s)	$\hat{P}_{\mathrm{mc}}$	$\sigma(\hat{P}_{\mathrm{mc}})$	Elapsed (s)
1	0.13144	0.011	0.13143	0.00056	1.010
2	0.13144	0.164	0.13141	0.00041	0.885
3	0.13144	3.163	0.13133	0.00042	0.994
4	0.13144	23.061	0.13132	0.00058	0.854
5	0.13145	23.689	0.13140	0.00051	0.954

Supplementary Table 4: P_{mc} values computed using cubature methods and Monte Carlo integration based on 50 replicates. The P_{mc} and $\hat{P}_{\mathrm{mc}}$ values are averages across all replicates. Elapsed time (in seconds) is the average time for a single replicate.

Supplementary Methods

1 Monte Carlo P_{mc} Comparison

Here we compare P_{mc} computed using standard cubature methods to $\hat{P}_{\text{mc}}$ estimated using a Monte Carlo (MC) integral (as described in Section 4.5). We use 50 replicates to obtain the timing measurements and quantify the uncertainty in the MC integral. All values presented are means over these replicates unless otherwise specified. For the MC integration procedure we use $M = 10^5$ sample points, and parallelize the computation over 8 cores.

The distribution in question consists of three Gaussian clusters with $\pi_k = 1/3$ in $\mathbb{R}^p$ with I_p variance. The cluster are centered at 0^p and $\pm d^p$, where d^p is the p-dimensional vector whose elements are all d. d is set so that the Euclidean distance between d^p and the origin is fixed to be 3; i.e. $d = \sqrt{3^2/p}$. This is so that the value of P_{mc} remains fixed across dimensions and we do not need to worry about the dimensionality affecting the true value of P_{mc}. The results for dimension $p = 1, \ldots, 5$ are presented in Supplementary Table 4.

Both approaches produce highly similar values of P_{mc} across dimensions. The standard deviation of the MC estimates is quite low, indicating stability in the estimation procedure. However, while the cubature method evaluation time rapidly increases in p, the time for the MC procedure is roughly constant. These together highlight the MC estimation procedure as a viable and accurate approach to estimate P_{mc}, especially in moderate dimensional data (i.e., $p \geq 3$) where cubature methods may struggle to reach a solution in a reasonable time.