HHS Public Access

Author manuscript

IEEE Access. Author manuscript; available in PMC 2023 June 02.

Published in final edited form as:

IEEE Access. 2022 ; 10: 116844–116857. doi:10.1109/access.2022.3218800.

A Hyperparameter-Free, Fast and Efficient Framework to Detect Clusters From Limited Samples Based on Ultra High-Dimensional Features

SHAHINA RAHMAN,

VALEN E. JOHNSON,

SUHASINI SUBBA RAO

Department of Statistics, Texas A & M University, College Station, TX 77843, USA

Abstract

Clustering is a challenging problem in machine learning in which one attempts to group N objects into K_0 groups based on P features measured on each object. In this article, we examine the case where $N \ll P$ and K_0 is not known. Clustering in such high dimensional, small sample size settings has numerous applications in biology, medicine, the social sciences, clinical trials, and other scientific and experimental fields. Whereas most existing clustering algorithms either require the number of clusters to be known a priori or are sensitive to the choice of tuning parameters, our method does not require the prior specification of K_0 or any tuning parameters. This represents an important advantage for our method because training data are not available in the applications we consider (i.e., in unsupervised learning problems). Without training data, estimating K_0 and other hyperparameters–and thus applying alternative clustering algorithms–can be difficult and lead to inaccurate results. Our method is based on a simple transformation of the Gram matrix and application of the strong law of large numbers to the transformed matrix. If the correlation between features decays as the number of features grows, we show that the transformed feature vectors concentrate tightly around their respective cluster expectations in a low-dimensional space. This result simplifies the detection and visualization of the unknown cluster configuration. We illustrate the algorithm by applying it to 32 benchmarked microarray datasets, each containing thousands of genomic features measured on a relatively small number of tissue samples. Compared to 21 other commonly used clustering methods, we find that the proposed algorithm is faster and twice as accurate in determining the “best” cluster configuration.

Keywords

Clustering; gram matrix; high-dimensional features; hyperparameter-free

Corresponding author: Valen E. Johnson (vejohnson@exchange.tamu.edu).

The associate editor coordinating the review of this manuscript and approving it for publication was Qilian Lianghttp://orcid.org/0000-0002-3630-8010.

I. INTRODUCTION

Clustering is an unsupervised learning technique used to discover natural groups among N objects. Vectors of P measurements, or features, measured on each object define the groups. The concept of a cluster can be subjective, and clustering techniques have been studied extensively with varying notions of "similarity" defined on the feature vectors in various scientific domains [1]. Large datasets containing a comparatively small number of features for large numbers of objects have become common in many fields because modern technology facilitates the collection of these data. In contrast, advances in neuroimaging, genomics, motion-tracking, and many other technology-based data collection methods have led to many datasets containing a small number of samples and many features. Limited sample numbers are also common because experimental protocols designed to discriminate between treatment groups have high costs and involve human participants. Hence, it is common to acquire several thousands of features on a limited number of objects in various scientific domains. Such measurements give rise to high-dimensional data where $N \ll P$, leading to a "small data" challenge that requires an entirely different mindset than that used in the "big data," or $N \gg P$ paradigm. As one of many such examples, we consider here a large set of well-studied benchmarked microarray datasets, where thousands of gene expression values were measured on a moderate number of excised tumor tissue samples.

To detect clusters in high dimensional data, one class of clustering techniques (including, [2], [3], [4], [5], [6], [7], [8], [9]) assumes that information regarding meaningful clusters is contained in a small number of features. These algorithms, often referred to as *sparse clustering procedures*, rely on "relevant" feature extraction that influences resulting cluster partitions. Another class of algorithms focuses on finding clusters in a lower-dimensional latent space representation using either deep learning embedding or non-deep learning embedding, often referred to as *embedding clustering procedures*. This class of techniques is popular for clustering non-rectilinear data, like images, text, or web documents. Popular methods include [10], [11], [12], and [13]. Although the deep learning clustering procedures have become popular for large datasets involving thousands of data points, they fail to control the problem of over-fitting when the sample sizes are small.

Over the last two decades, spectral-based clustering (e.g., [14] and [15]) and its variants ([10], [16]) have become popular and are based on detecting clusters using the Gram matrix and variations of it. In specific contexts, an optimal solution is obtained using the spectral decomposition of the Gram matrix and the cluster information embedded in its eigenvectors. The choice for the number of clusters K_0 is not well established and is often based on "eigen-gap" heuristics. Unfortunately, the limiting properties of the eigenvectors in noisy settings can be unstable, and these methods often do not converge to suitable partitions; these limitations are explored in [17] and the study below.

Many existing clustering algorithms are popular in specific applications. Still, they typically have two significant limitations: (1) most current algorithms assume that the number of true clusters, K_0, is known, and (2) they are often equipped with several hyperparameters that need to be finely tuned to apply in various applications. In contrast to the supervised setting, in unsupervised problems like clustering, training data are not available to provide

guidelines on labels. Hence, a technique like cross-validation is not plausible in such a setting. The challenges of selecting hyperparameters are well known, and the clusters produced by algorithms that depend on hyperparameters can be sensitive to their selection. As a result, clustering algorithms that depend on the selection of hyperparameters are often not effective in limited sample settings.

A standard empirical procedure that many clustering algorithms use for selecting hyperparameters is *stability selection* [18], [19]. However, such procedures are not accompanied by any theoretical guarantees [20]. They split the data into repeated folds (like bootstrap samples) and select hyperparameters with the most stable solution. The repeated sampling approaches are often computationally intensive and sometimes lead to infeasible clustering results. A recent discussion on the problems of fine-tuning hyperparameters while studying the sensitivity of various popular and state-of-the-art clustering algorithms in real-life applications is presented in [21]. As pointed out in this article, implementing these methods is computationally challenging when estimating the optimal number of clusters, thus impairing their performance.

In this article, we describe a hyperparameter-free, robust, and computationally efficient clustering framework for high dimensional data, where the number of clusters is not required to be specified by the user. Drawing inspiration from the strong law of large numbers, our framework involves a simple algebraic transformation on the Gram matrix of features. When the correlation among the original features is “weak” as the number of features, P, grows, we show that the transformed feature vectors concentrate tightly around their lower-dimensional expectation vectors. Thus, the proposed transformation significantly simplifies and improves the detection and visualization of the hidden clusters. Unlike spectral clustering, our method does not attempt to discover the underlying clusters from the spectral characteristic of the Gram matrix, which can be expensive to compute. Instead, we propose to detect clusters from transformed lower-dimensional row vectors of the Gram matrix, which significantly reduces computational costs. In this paper, we use the Bayesian Information Criteria (BIC), a model selection tool to estimate the number of clusters, as in other works including, [2], [3], [22], [23], and [24]. While previous methods have implemented BIC on original features, we apply it to a transformed feature matrix to estimate the unknown number of clusters.

The rest of this paper is organized as follows. In Section II, we propose a two-step transformation on the Gram matrix and provide the motivation and illustration of the transformation. In Section III, we state the assumptions on the original feature space. We show that as the number of features grows, the transformed feature vectors concentrate tightly around their cluster-specific means at a rate that is order $O(1 / \sqrt{P})$. We present our clustering algorithm in Section IV. To confirm the performance of our proposed clustering algorithm, we present a large-scale real-life data study in Section V, where 21 high-dimensional clustering algorithms are applied to 32 benchmarked microarray datasets. Finally, we conclude the paper with a discussion on future research directions.

A. ABBREVIATIONS AND ACRONYMS

We denote matrices by upper case bold letters (e.g., $\boldsymbol{X}$) and column vectors by lower case bold letters (e.g., $\mathbf{u}$). We use $\mathbb{1}_P$ to denote the P-dimensional vector of ones, and $\mathbb{I}_A$ to represent the indicator function, which equals 1 if A is true and 0 otherwise. We let $\|\cdot\|_2$ denote the Euclidean distance of a vector or Frobenius norm of a matrix, and $\|\cdot\|_1$ denote the absolute sum of the entries of a vector or matrix. We define $\boldsymbol{I}_m$ as the m dimensional identity matrix. We write $i \simeq j$ if objects i and j are in the same cluster, and $i \not\simeq j$ otherwise. We use $\mathbb{E}(\cdot)$ and $\mathrm{Var}(\cdot)$ to denote the expectation and variance of a random entity. We also use $\mathbb{E}(A \mid B)$ to denote the conditional expectation of A given B. When a P-dimensional random vector $\boldsymbol{u}$ follows a multivariate normal distribution with P-dimensional mean vector, $\boldsymbol{\theta}$, and $P \times P$ covariance matrix, $\boldsymbol{\Gamma}$, we denote the multivariate normal density of $\boldsymbol{u}$ by

$$\mathcal{N}(\boldsymbol{u} \mid \boldsymbol{\theta}, \boldsymbol{\Gamma}) = \frac{\exp\left\{-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\theta})^T \boldsymbol{\Gamma}^{-1}(\boldsymbol{u}-\boldsymbol{\theta})\right\}}{(2\pi)^{N/2}\det(\boldsymbol{\Gamma})^{1/2}}, \tag{1}$$

where $\det(\cdot)$ is the determinant of a matrix.

B. FRAMEWORK

Let K_0 denote the unknown number of real clusters. Let $\boldsymbol{z}_i^T = (z_{i,1}, \ldots, z_{i,P}) \in \mathbb{R}^P$ denote the feature vector measured on object i and stack the feature vectors of N objects into an $N \times P$ feature matrix $\boldsymbol{X}_{\mathrm{raw}}$. We standardize the columns of $\boldsymbol{X}_{\mathrm{raw}}$ so that each column has median or mean 0 and standard deviation 1. We denote $\boldsymbol{x}_i^T = (x_{i,1}, \ldots, x_{i,P}) \in \mathbb{R}^P$ as the i^{th} row of $\boldsymbol{X}$ and treat this as the feature vector of object i. For a given K_0, define δ_i to be an integer in $\{1, \ldots, K_0\}$ that denotes the cluster membership of object i. We denote $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)$ as the vector of cluster identifiers, where $\Pr(\delta_i = a) = w_a$ with $\sum_{a=1}^{K_0} w_a = 1$. We assume that $\{\boldsymbol{x}_i\}_{i=1}^N$ are P dimensional random vectors with $\mathbb{E}(\boldsymbol{x}_i \mid \delta_i = a) = \boldsymbol{\mu}_a$, where $\boldsymbol{\mu}_a^T = (\mu_{a,1}, \ldots, \mu_{a,P}) \in \mathbb{R}^P$ and finite covariance matrix $\mathrm{Var}(\boldsymbol{x}_i \mid \delta_i = a) = \boldsymbol{\Sigma}_a \in \mathbb{R}^{P \times P}$. We make no further assumptions on the distribution of $\boldsymbol{x}_i$. We denote the vector of diagonal elements of $\boldsymbol{\Sigma}_a$ as $\boldsymbol{d}_a^T = (\sigma_1^a, \ldots, \sigma_P^a) \in \mathbb{R}^P$. For any two clusters, $a, b \in \{1, \ldots, K_0\}$, we define the following quantities,

$$\theta_{a,b} = \boldsymbol{\mu}_a^T \boldsymbol{\mu}_b / P \in \mathbb{R}, \tag{2}$$

and

$$\theta_a = \boldsymbol{\mu}_a^T \boldsymbol{\mu}_a / P + \boldsymbol{d}_a^T \mathbb{1}_P / P \in \mathbb{R}.$$

II. THE TRANSFORMATION ON GRAM MATRIX

The goal of our clustering algorithm is to partition N objects into K_0 clusters using a transformed Gram matrix rather than the X matrix itself. We assume that K_0 is unknown. The algorithm consists of two simple transformations, followed by the application of standard clustering algorithms to the transformed version of the Gram matrix. To the best of our knowledge, the following transformation has not been previously considered. In

this section, we present the motivation and details of the transformation. We present its theoretical properties in section III.

A. THE G-VECTORS

Definition 2.1: Using the standardized feature matrix, **X**, we construct the matrix **G** according to

$$\boldsymbol{G} = \boldsymbol{X}\boldsymbol{X}^T \ / \ P = (g_{i,j}) \in \mathbb{R}^{N \times N}. \tag{3}$$

Instead of the original feature vector $\boldsymbol{x}_i \in \mathbb{R}^P$, each object i is now represented by row i of the **G** matrix, denoted as $\boldsymbol{g}_i \in \mathbb{R}^N$. We refer to the rows of the **G** matrix as g-vectors.

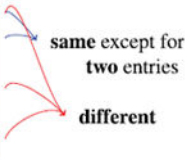
In Lemma 2 of the appendix we show that the clusters can be recovered from the expectation of the **G** matrix. Specifically we show that if object *i* belongs to cluster *a* and object *j* belongs to cluster *b*, then the conditional expectation of the entries of the **G** matrix is

$$\begin{aligned} &\mathbb{E}(g_{i,j} \mid \delta_i = a, \delta_j = b) = \theta_{a,b} \quad \text{for } i \neq j, \\ &\mathbb{E}(g_{i,i} \mid \delta_i = a) = \theta_a, \end{aligned} \tag{4}$$

where, $\theta_{a,b}$ and $\theta_a \in \mathbb{R}$ are defined in (2). We also show in equation (13) and (14) in Lemma 3 that when the features have a weak dependence structure, the variance of the *g-vectors* is of order $O(1/P)$. The law of large numbers implies that the *g-vectors* cluster tightly around their cluster-dependent expectations as the number of features increases. Thus, clustering the objects by their N-dimensional *g-vectors* can be easier than clustering them by their $P(\gg N)$ dimensional feature vectors. We illustrate this critical property of the rows of the **G** matrix in Fig. 1, where clusters are visually more apparent using *g-vectors* than by using the original feature vectors of $\boldsymbol{X}$.

Next, we illustrate the structure of the expectation of the **G** matrix for $N = 6$ objects with P features. Here objects 1 and 2 belong to cluster 1, objects 3 and 4 belong to cluster 2, and objects 5 and 6 belong to cluster 3. There are thus 6 *g-vectors* $\in \mathbb{R}^6$ that correspond to 6 objects. Using equation (2), one can observe the cluster-dependent partition of the expectation of **G** when the objects are arranged according to their cluster labels:

$$\mathbb{E}(\mathbf{G}) = \begin{bmatrix} \theta_1 & \theta_{1,1} & \theta_{1,2} & \theta_{1,2} & \theta_{1,3} & \theta_{1,3} \\ \theta_{1,1} & \theta_1 & \theta_{1,2} & \theta_{1,2} & \theta_{1,3} & \theta_{1,3} \\ \theta_{1,2} & \theta_{1,2} & \theta_2 & \theta_{2,2} & \theta_{2,3} & \theta_{2,3} \\ \theta_{1,2} & \theta_{1,2} & \theta_{2,2} & \theta_2 & \theta_{2,3} & \theta_{2,3} \\ \theta_{1,3} & \theta_{1,3} & \theta_{2,3} & \theta_{2,3} & \theta_3 & \theta_{3,3} \\ \theta_{1,3} & \theta_{1,3} & \theta_{2,3} & \theta_{2,3} & \theta_{3,3} & \theta_3 \end{bmatrix}$$



(5)

Equation (5) illustrates that when objects belong to different clusters, their corresponding *g-vector* expectations are different. However, when objects belong to the same cluster, their *g-vector* expectations are identical–except for two diagonal entries. From this illustration, we can see that if the cluster labels are known, the rearrangement of the diagonal entries can

lead to the re-alignment of the *g-vectors* to achieve similar centroid expectations. However, in an unsupervised setting, we are not provided with knowledge of the cluster labels of the objects. Hence, finding the optimal rearrangement of the diagonals that align the *g-vectors* correctly is not possible without an estimate of the object labels. Therefore, we propose the following transformation on the rows of the **G** matrix to align the *g-vectors* so that they have more similar means on the diagonal of **G**.

B. PARTIAL ALIGNMENT OF G-VECTORS

To realign the columns of **G** so that the cluster means match more closely, we begin by appending the diagonal elements of **G** as an additional column to a juxtaposed version of **G**; we call this matrix **M**. Next, we define the diagonal entries of **M** to be the average of the remaining elements of the corresponding column. More specifically, we define the **M** matrix as follows.

Definition 2.2: Given the N × N matrix **G**, we define the N × (N + 1) dimensional matrix **M** with entries

$$m_{i,j} = \begin{cases} g_{i,j}, & for\ j \neq i = 1, \ldots, N \\ g_{i,i}, & for\ j = N+1 \\ \frac{1}{N-1} \sum_{\substack{j=1 \\ j \neq i}}^{N} g_{i,j}, & for\ j = i\ . \end{cases}$$

We refer the rows of **M** matrix as m-vectors and denote by $\mathbf{m}_i \in \mathbb{R}^{N+1}$ the i^{th} m-vector.

To illustrate the above transformation, we consider an example with $N = 4$ objects where objects 1 and 2 belong to cluster 1, and objects 3 and 4 belong to cluster 2. Then we transform the **G** matrix,

$$\mathbf{G} = \begin{bmatrix} g_{1,1} & g_{1,2} & g_{1,3} & g_{1,4} \\ g_{2,1} & g_{2,2} & g_{2,3} & g_{2,4} \\ g_{3,1} & g_{3,2} & g_{3,3} & g_{3,4} \\ g_{4,1} & g_{4,2} & g_{4,3} & g_{4,4} \end{bmatrix}$$

to the **M** matrix displayed in Equation (6).

It follows that the expectations of the non-diagonal entries of the **M** matrix are identical whenever the objects corresponding to the rows belong to the same cluster. The expectations of *m-vectors* i and j are nearly identical except for their i^{th} and the j^{th} entries. If N is not too small, then the Euclidean distance between the expected *m-vectors* from the same cluster is small, differing only by those two entries.

When the original features are weakly dependent (see assumption 2 below), Fig. 1 shows how the *g-vectors* concentrate around their expected means. By Definition 2.2, it follows that the *m-vectors* also concentrate around their expectations as P increases, despite the small bias introduced by the diagonal of **M**. Importantly, when objects, say, i and $(i + 1)$ are in the same cluster, the difference between the expectations of $m_{i,i}$ and $m_{i+1,i}$ is generally

smaller than the difference between the expectations of $g_{i,i}$ and $g_{i+1,i}$ because the diagonal elements of **G** tend to be larger than the non-diagonal elements.

The *m-vectors* are usually more easily clustered than the original feature vectors for two reasons. First, because the elements of **M** represent averages of cross-products of P features, they are less variable than standardized feature vectors. Second, the dimension of the *m-vectors* is, by assumption, much lower than the dimension of the original feature vectors (i.e., $N \ll P$). The *m-vectors* thus simplifies the task of clustering for standard clustering techniques that can effectively detect clusters in a lower-dimensional space.

Fig. 2 illustrates this effect for benchmarked, high-dimensional microarray data presented in *Armstrong-v1* (described further below in Tables 1 and 2). These data contain 1, 083 gene expression values for 72 tissue samples taken from patients suffering from one of three leukemia subtypes (Acute Myeloid Leukemia (AML), Acute Lymphocytic Leukemia (ALL), and Mixed-lineage leukemia (MLL)). To visualize this high-dimensional data, we applied two popular non-linear dimension reduction techniques, *t-SNE* [26] and *UMAP* [27], to both the original gene expression data $\boldsymbol{x}_i \in \mathbb{R}^P$ and the *m-vectors*, $\boldsymbol{m}_i \in \mathbb{R}^{N+1}$. This figure demonstrates that t-SNE and UMAP more clearly separate the AML, ALL, and MLL tissue samples when they are applied to the *m-vectors*.

C. FULL ALIGNMENT OF G-VECTORS

The goal of this final transformation is to make the means of the *m-vectors* for objects in the same cluster equal. In section II-B we defined entry i of the *m-vector* m_i to be the mean of all the entries of column i of **G**, excluding $g_{i,i}$. We now replace this value with the average of entries in column i *that are estimated to belong to the same cluster*. That is, we use the estimate of the cluster label vector $\hat{\boldsymbol{\delta}}$, obtained after clustering the *m-vectors* to define $\mathbf{M}^{\hat{\boldsymbol{\delta}}} \in \mathbb{R}^{N \times (N+1)}$ as follows.

Definition 2.3: Define $\mathbf{M}^{\hat{\boldsymbol{\delta}}}$ to be the N × (N + 1) matrix having elements

$$m^{\hat{\boldsymbol{\delta}}}_{i,j} = \begin{cases} \dfrac{\sum_{\substack{j=1 \\ j \neq i}}^{N} g_{j,i} \cdot \mathbb{I}_{\{\hat{\delta}_j = \hat{\delta}_i\}}}{\sum_{\substack{j=1 \\ j \neq i}}^{N} \mathbb{I}_{\{\hat{\delta}_j = \hat{\delta}_i\}}}, & for\ j = i \\ m_{i,j}, & for\ j \neq i\ . \end{cases}$$

We refer the rows of $M^{\hat{\boldsymbol{\delta}}}$ as $m^{\hat{\boldsymbol{\delta}}}$-vectors.

This additional transformation can be significant for small N when the pairwise distance between the expectations of the *m-vectors* for objects in the same cluster is not negligibly small. Because the expectations of the *m-vectors* for objects in the same cluster are slightly different in the **M** matrix, model selection criteria tend to overfit the number of clusters. Hence, in the model-selection phase of our algorithm, we reestimate the diagonal elements of **M** using the current estimates of the cluster labels. Experimentally we find that this transformation improves the performance of model selection criteria when N is small. We

do not use the $\mathbf{M}^{\hat{\hat{\boldsymbol{\delta}}}}$ matrix in the estimation phase of the algorithm to avoid repeated updates of the **M** matrix within each iteration of a clustering algorithm; doing so can increase computational costs without providing significant gains in accuracy.

$$\mathbf{M} = \begin{bmatrix} \frac{g_{2,1}+g_{3,1}+g_{4,1}}{3} & g_{1,2} & g_{1,3} & g_{1,4} & g_{1,1} \\ g_{2,1} & \frac{g_{1,2}+g_{3,2}+g_{4,2}}{3} & g_{2,3} & g_{2,4} & g_{2,2} \\ g_{3,1} & g_{3,2} & \frac{g_{1,3}+g_{2,3}+g_{4,3}}{3} & g_{3,4} & g_{3,3} \\ g_{4,1} & g_{4,2} & g_{4,3} & \frac{g_{1,4}+g_{2,4}+g_{3,4}}{3} & g_{4,4} \end{bmatrix}. \quad (6)$$

We now define the expectation of the $m^{\hat{\hat{\boldsymbol{\delta}}}}$-vectors as follows:

Definition 2.4: If $\delta_i = a$ (i.e., object i belongs to cluster a), we denote the expectation of row i of $\mathbf{M}^{\boldsymbol{\delta}}$ by $\boldsymbol{\theta}_a$, defined as

$$\mathbb{E}\left(m_i^{\boldsymbol{\delta}} \mid \delta_i = a\right) = \boldsymbol{\theta}_a \equiv \left(\{\theta_{a,\delta_j}\}_{j=1}^{N}, \theta_a\right) \in \mathbb{R}^{N+1}, \quad (7)$$

where $\theta_{a,\delta_j} \in \mathbb{R}$ is defined in (2) for $\delta_j \in \{1, \ldots, K\}$.

To illustrate, consider again the example in (6). The transformation from the **G** matrix to $\mathbf{M}^{\boldsymbol{\delta}}$ is

$$\mathbf{M}^{\boldsymbol{\delta}} = \begin{bmatrix} g_{2,1} & g_{1,2} & g_{1,3} & g_{1,4} & g_{1,1} \\ g_{2,1} & g_{1,2} & g_{2,3} & g_{2,4} & g_{2,2} \\ g_{3,1} & g_{3,2} & g_{4,3} & g_{3,4} & g_{3,3} \\ g_{4,1} & g_{4,2} & g_{4,3} & g_{3,4} & g_{4,4} \end{bmatrix}.$$

Let $\boldsymbol{\Theta} = \mathbb{E}\left[\mathbf{M}^{\boldsymbol{\delta}} \mid \boldsymbol{\delta}\right]$, where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_{\delta_1}, \cdots, \boldsymbol{\theta}_{\delta_N})^T \in \mathbb{R}^{N \times (N+1)}$ represents the expected value of $\mathbf{M}^{\boldsymbol{\delta}}$ in the transformed space. Then this transformation of the **M** matrix makes the expected value of $m_i^{\boldsymbol{\delta}}$ and $m_j^{\boldsymbol{\delta}}$ equal whenever objects *i* and *j* belong to the same cluster. Using the example considered in (6), we now illustrate $\boldsymbol{\Theta}$ in the equation (8) below.

$$\mathbb{E}(\mathbf{M}^{\boldsymbol{\delta}}) = \begin{bmatrix} \theta_{1,1} & \theta_{1,1} & \theta_{1,2} & \theta_{1,2} & \theta_1 \\ \theta_{1,1} & \theta_{1,1} & \theta_{1,2} & \theta_{1,2} & \theta_1 \\ \theta_{1,2} & \theta_{1,2} & \theta_{2,2} & \theta_{2,2} & \theta_2 \\ \theta_{1,2} & \theta_{1,2} & \theta_{2,2} & \theta_{2,2} & \theta_2 \end{bmatrix} = \boldsymbol{\Theta}. \quad (8)$$

III. CONCENTRATION OF THE m^{δ}-VECTORS

In this section, we assume the following separability condition holds on the distinct cluster means on the transformed space.

Assumption 1: There exists $\eta > 0$ such that

$$| \, | \boldsymbol{\theta}_a - \boldsymbol{\theta}_b | \, |_2 > \eta, \tag{9}$$

for any two clusters, $a \neq b \in \{1, \ldots, K_0\}$.

Such a condition is required to ensure the identification of distinct clusters. It also implies that the proportion of informative features measured on objects does not converge to 0 as P grows.

In the following lemma, we obtain a rate of convergence for the L_2 distance between $\mathbf{M}^{\boldsymbol{\delta}}$ and $\boldsymbol{\Theta}$. To allow for convergence in the high-dimensional settings, we make the following assumptions about the covariance matrix of the feature vector $\boldsymbol{x}_i \in \mathbb{R}^P$.

Recall that $\boldsymbol{\Sigma}_a = \mathrm{Var}(\boldsymbol{x}_i | \delta_i = a)$ denotes the $P \times P$ covariance matrix of the feature vector when an object belongs to cluster a. We define $\tau_P = \max_{1 \leq a \leq K_0} \|\boldsymbol{\Sigma}_a\|_1^{1/2}$, where $\|\boldsymbol{A}\|_1$ is defined to be the sum of the absolute value of the entries of the matrix $\boldsymbol{A}$.

We let $y_{i,p} = x_{i,p} - \mu_{\delta_{i,p}}$ denote the centered feature values for each feature $p = 1, \ldots, P$. We also define the vector $z_i^T = (y_{i,1}^2, \ldots, y_{i,P}^2) \in \mathbb{R}^P$, and let $\boldsymbol{\Upsilon}_a = \mathrm{Var}(z_i | \delta_i = a)$ and $\kappa_P = \max_{1 \leq k \leq K_0} \|\boldsymbol{\Upsilon}_k\|_1^{1/2}$.

We now make the following assumption.

Assumption 2: We assume that $\boldsymbol{\kappa}_P/P = O(P^{-1/2})$ and $\boldsymbol{\tau}_P/P = O(P^{-1/2})$ as $P \to \infty$.

We call the features weakly dependent when they follow assumption 2.

Lemma 1: Suppose Assumption 1 and 2 hold. Let $\mu_{\sup} = \sup_{a,p} |\mu_{a,p}| < \infty$ and $\sigma_{\sup} = \sup_{a,p} \sqrt{\sigma_p^a} < \infty$. Then

$$\mathbb{E}\|\mathbf{M}^{\boldsymbol{\delta}} - \boldsymbol{\Theta}\|_2^2 \leq \Delta_P^2$$

where

$$\Delta_P = \frac{1}{P}\Big[N\{(N-1)\tau_P^2(2\mu_{\sup} + \sigma_{\sup})^2 + (\kappa_P + 2\tau_P\mu_{\sup})^2\}\Big]^{1/2}.$$

The proof of this lemma is provided in the appendix.

For a correctly specified cluster configuration in which $\hat{\boldsymbol{\delta}} = \boldsymbol{\delta}$, this lemma implies that $\mathbf{M}^{\boldsymbol{\delta}}$ converges in probability to $\boldsymbol{\Theta}$. For weakly dependent features, the rate of convergence is of order $O_p(P^{-1/2})$. For $K \leq K_0$, the identifiability condition (9) guarantees that this sum-of-squares is bounded away from 0. For $K > K_0$, the BIC penalty is sufficiently large to prevent sub-clusters from a given cluster from forming since the decrease in the sum-of-squares accumulated from such a split cannot offset a fixed penalty greater than $\log(N)$. Thus, standard center-based clustering algorithms are likely to identify the correct cluster identifiers, provided that the conditions stated above are satisfied and P is sufficiently large.

IV. THE CLUSTERING ALGORITHM: GMAC

To estimate the underlying number of clusters K_0 and the cluster indicator vector $\boldsymbol{\delta}$, we maximize a quasi-mixture likelihood for each possible value of $K = 1, \ldots, K_{max}$. Here, $K_{\max}$ is a user-defined upper bound on K_0, which may equal N if a prior bound is unknown. Because the elements of $\mathbf{M}$ and $\mathbf{M}^{\boldsymbol{\delta}}$ represent an average of P pairwise products, under certain regularity conditions, the rows of $\mathbf{M}$ and $\mathbf{M}^{\boldsymbol{\delta}}$ converge to a multivariate normal distribution. We, therefore, maximize a quasi-mixture likelihood function of the form

$$L_K(\mathbf{M}) = \sum_{i=1}^{N} \log\left[\sum_{k=1}^{K} w_k \mathcal{N}(m_i \mid \boldsymbol{\theta}_k, \boldsymbol{\Gamma}_k)\right], \tag{10}$$

where $\mathcal{N}$ is defined in (1) and the mixing weights w_k satisfy $w_k \geq 0$ and $\sum_{k=1}^{K} w_k = 1$.

Let $\hat{L}_K(\mathbf{M}) = L_K(\mathbf{M}; \hat{w}_K, \hat{\boldsymbol{\theta}}_K, \hat{\boldsymbol{\Gamma}}_K)$ denote the maximized quasi log-likelihood for an assumed value of K. For a given K, the EM algorithm [28] is used to obtain maximum likelihood estimates for the parameters $\{\hat{w}_k, \hat{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\Gamma}}_k\}_{k=1}^{K}$ of the mixture model and the latent cluster identifiers $\hat{\boldsymbol{\delta}}$.

We estimate the number of clusters by maximizing the Bayesian Information Criterion (BIC) [29], which can be expressed as

$$\mathrm{BIC}_K = 2\hat{L}_K(\mathbf{M}^{\hat{\boldsymbol{\delta}}}) - \nu_K \log N,$$

where ν_K is the number of estimated parameters in the likelihood function $L_K(\cdot)$.

The EM algorithm is only guaranteed to arrive at a local maximum of the mixture likelihood criterion [30]. Consequently, the choice of starting values for $\boldsymbol{\delta}$ is important. Previous studies on clustering low-dimensional multivariate mixture models by [31], [32], and [2] suggest that an initial estimate, $\hat{\boldsymbol{\delta}}$, obtained from agglomerative hierarchical clustering technique can provide effective initialization. We thus implemented an agglomerative hierarchical clustering technique on the $(N + 1)$ dimensional rows of the $\mathbf{M}$ matrix to obtain an initial estimate $\hat{\boldsymbol{\delta}}$ for a given K.

The pseudocode for implementing the resulting algorithm is described in Algorithm 1.

V. A COMPARATIVE STUDY ON A GENE EXPRESSION DATASETS

The evaluation of machine learning methods on real data is essential when comparing algorithms since the settings of simulation studies can be adjusted to favor one particular algorithm over another. Here, we test our clustering algorithm on a large collection of benchmarked, high-dimensional microarray datasets to evaluate the performance accuracy of our method. However, there is no acknowledged measure of choice to compare partitions and in practice many measures are used. To validate cluster solutions to a reference clustering, we use an information theoretic measure, *adjusted mutual information* (AMI), proposed by [33] and [34] that has been popular in validating many clustering studies([8], [11], [35], etc). The AMI between two cluster configurations, say A and B, is defined as

$$\mathrm{AMI}(A, B) = \frac{MI(A, B) - E[MI(A, B)]}{\sqrt{H(A)H(B)} - E[MI(A, B)]},$$

where $H(\cdot)$ denotes entropy and $\mathrm{MI}(\cdot, \cdot)$ denotes mutual information. An AMI value of 1 occurs when the two partitions are equal, and 0 represents the AMI value expected by chance under a hypergeometric sampling model for the partitions.

A. BENCHMARKED MICROARRAY DATA SETS

The gene expression data sets that we studied are available at DataLink [36]. Thirty-two studies of various tissue samples were included in our comparison. As discussed in [37], the scales upon which features are measured can have a strong influence on determining estimated cluster configurations. Because of the wide dynamic range of gene expression data, some of the data sets, including *Alizadeh-v1, Alizadeh-v2, Alizadeh-v3, Bittner, Garber, Lapointe-v1, Liang, Risinger, Singh-v1, Tomlins-v1* and *West*, was preprocessed and centered by the original authors. We did not apply additional transformations to these data. For the remaining data sets, we took the *logarithm transformation* on the feature matrix $\boldsymbol{X} \in \mathbb{R}^{N \times P}$ and then standardized each column (gene) of the resulting feature matrix by *centering* (with median) and *scaling* (with standard deviation). The "*Microarray-data*" folder, available in the GitHub repository https://github.com/srahman-24/GMac, provides all the transformed datasets and their respective cluster information.

Shah and Koltun [11] provided a recent comparison of 12 popular clustering methods for these 32 gene expression data sets. They based their comparison on the adjusted mutual information (AMI) [33] of estimated cluster configurations. The "true" cluster configurations of these data sets are well-studied, validated and available at DataLink [36]. To evaluate our method, GMAC, we compared its performance to the clustering methods considered in [11], adding 8 additional state-of-the-art algorithms to the comparison.

Algorithm 1 GMAC

Input: $X \in \mathbb{R}^{N \times P}$ and $K_{\max}$.
Output: $\widehat{K}$; Cluster identifier $\widehat{\boldsymbol{\delta}} = (\widehat{\delta}_1, \ldots, \widehat{\delta}_N)$.

1) **G-step:** Construct the $N \times N$ similarity matrix, $\mathbf{G} = XX^T/P$.
2) **M-step:** Rearrange the diagonal elements of $\mathbf{G}$ to construct the $N \times (N+1)$ matrix $\mathbf{M}$.
3) **for** $K = 1$ **to** $K_{\max}$:
   a. **Initialize:** Use agglomerative hierarchical clustering on $\mathbf{M}$ to initialize the cluster identifier $\widehat{\boldsymbol{\delta}}$ having K clusters.
   b. **Repeat**
      i) *Maximization* : **for** $k = 1$ **to** K :
      $n_k = \sum_{i=1}^{N} \mathbb{I}_{\{\widehat{\delta}_i = k\}}$; $\widehat{w}_k = n_k/N$; $\widehat{\boldsymbol{\theta}}_k = \sum_{i=1}^{N} m_i \mathbb{I}_{\{\widehat{\delta}_i = k\}}/n_k$;
      $\widehat{\boldsymbol{\Gamma}}_k = \sum_{i=1}^{N} (m_i - \widehat{\boldsymbol{\theta}}_k)(m_i - \widehat{\boldsymbol{\theta}}_k)^T \mathbb{I}_{\{\widehat{\delta}_i = k\}}/n_k$.
      ii) *Expectation* : **for** $i = 1$ **to** N:
      Set $\widehat{\delta}_i = \widehat{k}$ if $\arg\max_{k=1}^{K} \widehat{w}_k \mathcal{N}\left(m_i | \widehat{\boldsymbol{\theta}}_k, \widehat{\boldsymbol{\Gamma}}_k\right) = \widehat{k}$.
      until until $\widehat{\boldsymbol{\delta}}$ doesn't change.
   c. **Model selection :** Given $\widehat{\boldsymbol{\delta}}$, calculate BIC_K based on $\mathbf{M}^{\widehat{\boldsymbol{\delta}}}$.
4) **Return** K and $\widehat{\boldsymbol{\delta}}$ that maximize BIC_K

We broadly divided the 21 other clustering methods into *two categories*, according to whether or not the algorithms estimated the number of clusters. Out of 21, 13 algorithms *required* pre-specification of the number of clusters K_0 : k-means++ (KM++) [38], Gaussian mixture models (GMM) [32], fuzzy clustering (FUZZY) [39], mean-shift clustering (MS) [40], agglomerative hierarchical clustering with ward linkage (AC-W), normalized cuts (N-Cuts) [15], Zeta l-links (ZELL) [41], spectral embedded clustering (SEC) [10], clustering using local discriminant models and global integration (LDMGI) [42], path integral clustering (PIC) [43], sparse k-means (SP-KM) [4], sparse subspace clustering (SSC) [44] and deep embedding clustering (DEC) [12]. We also considered 8 other clustering methods that *do not require* the pre-specification of K_0: Extended K-means with BIC (X-KM) [22], affinity propagation (AP) [45], a robust graph continuous clustering (RCC) [11], GAP statistics implemented with sparse K-means (GAP+S-KM) [4], [46], a model based clustering with variable selection (CVAR) [2], a high dimensional data clustering (HDDC) [24], a graph clustering based on tensors (SPEC) [16] and recent clustering method with simultaneous variable selection and estimation of K that is based on resampling method (S4) [9]. Overall, we compared GMAC with 13 clustering algorithms that require

pre-specification of the number of clusters and 8 clustering algorithms that do not. Out of the 8 clustering methods in the second category, X-KM, CVAR, HDDC and GMAC implement the BIC criterion to select the number of clusters K. Details of the hyperparameter settings used for these methods are provided in Table 4 in appendix. The number of clusters was estimated in all implementations of GMAC.

B. RESULTS

We provide the AMI values achieved by each clustering algorithm for each data set in Table 1 and 2. For brevity, we have excluded the AMI results from GMM, FUZZY, MS, SSC, and DEC and displayed the AMI results from 17 algorithms that yielded the best AMI value for at least 1 out of 32 data sets in Table 1 and 2. Figs. 4 and 3 summarizes the results of these comparisons. The barplot depicts the frequency at which each method yielded the highest accuracy based on AMI. The figure shows that GMAC achieved the highest AMI for 15 data sets, while the next best algorithm, S4, provided the highest AMI in 8 data sets. The boxplot in Fig. 3 shows how well the clustering methods estimate the number of clusters. Fig. 3 shows that GMAC provides the best estimates, $\widehat{K}$, of K_0.

We also compared the computational performance of GMAC to six other algorithms that, like GMAC, also estimated the number of clusters, K_0. Because the runtime complexity of the GMAC algorithm is linear in P, it was typically faster than the other algorithms (Fig. 5). Indeed, except for the AP algorithm (which achieved the best AMI in three microarray data set in Table 2 and relatively produced the largest bias and variability in estimating the number of clusters K shown in Fig. 3), GMAC was faster, on average, than all other algorithms that we tested, and was substantially faster for $NP < 400,000$. All comparisons were performed on a workstation with an Intel(R) Core(TM) i7-3770 CPU clocked at 3.40GHz with 8.00 GB RAM. The data sets and algorithms that produced these results are available at https://github.com/srahman-24/GMac.

VI. CONCLUSION

The clustering task is exceptionally challenging when high-dimensional features are collected on only a limited number of samples ($N \ll P$). Most existing clustering algorithms either assume the number of clusters is known a priori or require tuning several hyperparameters. The lack of training data makes the selection of hyperparameters very difficult in unsupervised settings. Stability methods used to select hyperparameters suffer high computational costs and typically do not guarantee an optimal solution. Our clustering algorithm offers a simple and computationally efficient technique to detect and visualize cluster configurations. It estimates the number of clusters and does not require the specification of hyperparameters; this allows the user to avoid subjective or default parameter choices. The overall complexity of the current algorithm is $O(N^2P)$, which makes it efficient in $N \ll P$ settings. Our method yielded the highest accuracy, as measured by AMI, more than twice as often as 21 competing algorithms when applied to 32 real-world genomic datasets.

A software implementation of our proposed algorithm, GMAC, is available in the R package, *RJcluster* [47].[1] In ongoing work, we plan to extend the algorithm for efficient

clustering in the “big data” paradigm, where both N and P are large. We are also developing transformations to cluster non-rectilinear high-dimensional data, like images and text documents.

ACKNOWLEDGMENT

The authors are grateful to Zoltan Szabo and Raymond Wong for their valuable time in providing detailed comments which helped us in reformatting the paper. They thank Anirban Bhattacharya and Irina Gaynavova for discussion, Marina Romanyuk for checking the computation times, and Rachael Shudde for maintaining the R package.

This work was supported in part by the National Institute of Health of United States of America under Grant CA R01 158113, and in part by the National Science Foundation of United States of America under Grant DMS-1812054.

Biographies

SHAHINA RAHMAN received the B.S. degree in mathematics from the University of Calcutta, Kolkata, India, in 2008, the M.S. degree in statistics from the Indian Statistical Institute, Kolkata, in 2010, and the Ph.D. degree in statistics from Texas A&M University, College Station, TX, USA, in 2015.

From 2015 to 2017, she was a Research Scientist at the Institute für Medizinische Biometrie und Statistik (IMBI), Freiburg, Germany. At Texas A&M University, she worked as a Postdoctoral Fellow (2017–2019) and an Instructional Assistant Professor (2019–2022) with the Department of Statistics. Her research interests include clustering, computational statistics, scalable unsupervised algorithms, and high-dimensional methodology.

Dr. Rahman is currently a member of the Scientific Committee for *Computational and Methodological Statistics*. She was a recipient of the ConocoPhillips Data Science Faculty Fellow Award in 2021.

VALEN E. JOHNSON received the B.S. degree in mathematics from the Rensselaer Polytechnic Institute, NY, USA, in 1981, the M.S. degree in mathematics from the University of Texas, Austin, TX, USA, in 1985, and the Ph.D. degree in statistics from The University of Chicago, Chicago, IL, USA, in 1989.

He has served as a tenured Professor at Duke University (1989–2002), University of Michigan (2002–2004), University of Texas M. D. Anderson Cancer Center (2004–2012), and Texas A&M University (2012–2022). From 2001 to 2002, he was a Research Scientist at the Los Alamos National Laboratory, Los Alamos, NM, USA. He served as the Deputy Chairperson for the Department of Bio-Statistics (2007–2010) and the Interim Head of Division of Quantitative Sciences (2011–2012) at the University of Texas M. D. Anderson Cancer Center, Houston, USA. Since 2016, he has been a University Distinguished Professor of statistics with the Texas A&M University, College Station, TX, USA; also the Department Head of Statistics (2014–2018), and also the Dean of the College of Science (2018–2022). He is the author of two books and holds two patents. His highly cited research articles are

[1]GMAC has been implemented as *RJcluster* in an open-source repository, CRAN, as an R package.

published in journals including Nature, *Proceedings of the National Academy of Sciences, Nature Human Behaviour, Journal of the American Statistical Association*, and *Journal of the Royal Statistical Society Series*. His current research interests include Bayesian hypothesis testing, variable selection, and clustering.

Dr. Johnson served as a member of the Board of Directors for International Society for Bayesian Analysis. He is a fellow of the American Association for the Advancement of Science (AAAS), a fellow of the American Statistical Association (ASA), a fellow of the Royal Statistical Society (RSS), and an Elected Member of the International Statistical Institute. He was a recipient of the Savage Award from International Society for Bayesian Statistics for Outstanding Thesis in Bayesian Statistics and Econometrics, in 1989, and runner-up for the Francois Erbsmann Award for contributions to medical imaging, in 1991. He has previously served as a Co-Editor of Bayesian Analysis and as an Associate Editor of the *Journal of the American Statistical Association* and IEEE TRANSACTIONS ON MEDICAL IMAGING.

SUHASINI SUBBA RAO received the B.S. and M.S. degrees in mathematics from the University of Manchester, U.K., in 1997, and the Ph.D. degree in statistics from the University of Bristol, U.K, in 2001. She is currently a tenured Professor with the Texas A&M University. Her current research interests include nonstationary time series, graphical models, and spectral theory. She is currently a Co-Editor of the *Journal of Time Series Analysis* and also an Associate Editor of *Sankhya, Korean Journal of Applied Statistics*, Statistics, *Probability and Mathematical Statistics*, and *Mathematical Methods of Statistics*.

APPENDIX PROOF OF LEMMA 3.1

The proof of Lemma 1 follows from the proof of the following two lemmas.

Lemma 2: Recall that $g_{i,j} = \sum_{p=1}^{P} x_{i,p}x_{j,p} / P$ with $\mathbb{E}(x_{i,p} \mid \delta_i) = \mu_{\delta_{i,p}}$ and $\mathbb{E}(x_i \mid \delta_i) = \boldsymbol{\mu}_{\delta_i}$ and $\mathrm{Var}(x_i \mid \delta_i) = \boldsymbol{\Sigma}_{\delta_i}$. Also recall that for clusters $a, b \in \{1, \ldots, K_0\}$, in equation 2 we defined $\theta_{a,b} = \boldsymbol{\mu}_a^T \boldsymbol{\mu}_b / P$ and $\theta_a = \boldsymbol{\mu}_a^T \boldsymbol{\mu}_a / P + \boldsymbol{d}_a^T 1_P / P$, where $\boldsymbol{d}_a^T = (\sigma_1^a, \ldots, \sigma_P^a)$ are the diagonal entries of $\boldsymbol{\Sigma}_a$. Then for $a, b \in \{1, \ldots, K_0\}$, $\mathbb{E}[g_{i,j} \mid \delta_i = a, \delta_j = b] = \theta_{a,b}$ and $\mathbb{E}[g_{i,i} \mid \delta_i] = \theta_a$.

Proof:

Suppose Assumption 2 holds. We define a P-dimensional vector $y_i = (y_i, 1, \ldots, y_{i,P})$ with components $y_{i,p} = x_{i,p} - \mu_{\delta_{i,p}}$. It follows that $\mathbb{E}[y_{i,p} \mid \delta_i = a] = 0$ and $\mathbb{E}[y_{i,p}^2 \mid \delta_i = a] = \sigma_p^a$. For $a \neq b \in \{1, \ldots, K_0\}$, If $i \neq j$, $\delta_i = a$, and $\delta_j = b$, then

$$g_{i,j} - \theta_{a,b} = \frac{1}{P}\sum_{p=1}^{P} y_{i,p}y_{j,p} + \frac{1}{P}\sum_{p=1}^{P} \mu_{a,p}y_{j,p} + \frac{1}{P}\sum_{p=1}^{P} \mu_{b,p}y_{i,p}, \tag{11}$$

and for $i = j$

$$g_{i,i} - \theta_a = \frac{1}{P}\sum_{p=1}^{P}(y_{i,p}^2 - \sigma_p^a) + \frac{2}{P}\sum_{p=1}^{P}\mu_{a,p}y_{i,p}\,. \tag{12}$$

Using the above and evaluating the conditional expectations $\mathbb{E}[g_{i,j} \mid \delta_i, \delta_j]$ and $\mathbb{E}[g_{i,i} \mid \delta_i]$ proves

$$\mathbb{E}(g_{i,j} \mid \delta_i = a, \delta_j = b) = \theta_{a,b} \text{ and } \mathbb{E}(g_{i,i} \mid \delta_i = a) = \theta_a\,.$$

Lemma 3: Suppose Assumption 2 holds. Recall in section III for the P-dimensional vector y_i^T, we defined $\boldsymbol{\Upsilon}_k = \mathrm{Var}(y_i \mid \delta_i = k)$ and $\kappa_P = \sup_{1 \le k \le K_0} \|\boldsymbol{\Upsilon}_k\|_1^{1/2}$ and $\tau_P = \sup_{1 \le k \le K_0} \|\boldsymbol{\Sigma}_k\|_1^{1/2}$. Let $\mu_{\sup} = \sup_{k,p} |\mu_{k,p}|$ and $\sigma_{\sup} = \sup_{k,p}\sqrt{\sigma_p^k}$. Then for $i \neq j$

$$\left(\mathbb{E} \mid g_{i,j} - \theta_{\delta_i,\delta_j} \mid^2\right)^{1/2} \le \frac{\tau_P}{P}(2\mu_{\sup} + \sigma_{\sup}) \tag{13}$$

and

$$\left(\mathbb{E} \mid g_{i,i} - \theta_{\delta_i} \mid^2\right)^{1/2} \le \frac{1}{P}(\kappa_P + 2\tau_P\mu_{\sup})\,. \tag{14}$$

Furthermore,

$$\begin{aligned}\mathbb{E}\left\|m_i^{\boldsymbol{\delta}} - \boldsymbol{\theta}_{\delta_i}\right\|_2^2 \le \frac{1}{P^2}[&(N-1)\tau_P^2(2\mu_{\sup} + \sigma_{\sup})^2 \\ &+ (\kappa_P + 2\tau_P\mu_{\sup})^2],\end{aligned} \tag{15}$$

and

$$\begin{aligned}\mathbb{E}\left\|\mathbf{M}^{\boldsymbol{\delta}} - \boldsymbol{\Theta}\right\|_2^2 \le \frac{N}{P^2}[&(N-1)\tau_P^2(2\mu_{\sup} + \sigma_{\sup})^2 \\ &+ (\kappa_P + 2\tau_P\mu_{\sup})^2]\,.\end{aligned} \tag{16}$$

Proof. We first prove (13), for the case $i \neq j$. We use (11) to give the bound

$$\begin{aligned}(\mathbb{E} \mid g_{i,j} - \theta_{\delta_i,\delta_j} \mid^2)^{1/2} \le &\left[\mathbb{E}\left(\frac{1}{P}\sum_{p=1}^{P} y_{i,p}y_{j,p}\right)^2\right]^{1/2} \\ &+ \left[\mathbb{E}\left(\frac{1}{P}\sum_{p=1}^{P} \mu_{\delta_i,p}y_{j,p}\right)^2\right]^{1/2} \\ &+ \left[\mathbb{E}\left(\frac{1}{P}\sum_{p=1}^{P} \mu_{\delta_j,p}y_{i,p}\right)^2\right]^{1/2}.\end{aligned}$$

Recall that $\mathbb{E}(A^2) = \mathbb{E}[\mathbb{E}(A^2 \mid \delta)]$ and if $\mathbb{E}[A \mid \delta] = 0$, then $\mathbb{E}(A^2) = \mathbb{E}[\mathrm{var}(A \mid \delta)]$. This implies

$$(\mathbb{E} \mid g_{i,j} - \theta_{\delta_i \delta_j} \mid^2)^{1/2} \leq A_{1,P} + A_{2,P} + A_{3,P}, \tag{17}$$

where $A_{1,P} = \left(\mathbb{E}\left[\mathrm{var}\left(\frac{1}{P}\sum_{p=1}^{P} y_{i,p} y_{j,p} \mid \delta_i, \delta_j\right)\right]\right)^{1/2}$,

$$A_{2,P} = \left(\mathbb{E}\left[\mathrm{var}\left(\frac{1}{P}\sum_{p=1}^{P} \mu_{\delta_j,p} y_{i,p} \mid \delta_i, \delta_j\right)\right]\right)^{1/2}, \text{and}$$
$$A_{3,P} = \left(\mathbb{E}\left[\mathrm{var}\left(\frac{1}{P}\sum_{p=1}^{P} \mu_{\delta_i,p} y_{j,p} \mid \delta_i, \delta_j\right)\right]\right)^{1/2}.$$

We now bound each of the terms $A_{1,P}$, $A_{2,P}$ and $A_{3,P}$. To bound $A_{1,P}$ we use the following decomposition:

$$\begin{aligned}
&\mathrm{var}\left(\frac{1}{P}\sum_{p=1}^{P} y_{i,p} y_{j,p} \mid \delta_i = a, \delta_j = b\right) \\
&= \frac{1}{P^2}\sum_{p_1, p_2 = 1}^{P} \mathrm{cov}\left(y_{i,p_1} y_{j,p_1}, y_{i,p_2} y_{j,p_2} \mid \delta_i = a, \delta_j = b\right) \\
&= \frac{1}{P^2}\sum_{p_1, p_2 = 1}^{P} \mathrm{cov}[y_{i,p_1}, y_{i,p_2} \mid \delta_i = a]\mathrm{cov}\left(y_{j,p_1}, y_{j,p_2} \mid \delta_j = b\right) \\
&\leq \sup_{a,p} \sigma_{a,p} \frac{1}{P^2} \sup_{a} \sum_{p_1, p_2 = 1}^{P} \left| \mathrm{cov}\left(y_{i,p_1}, y_{i,p_2} \mid \delta_i = a\right)\right| \\
&\leq \frac{\tau_P^2}{P^2}\sigma_{\sup}^2.
\end{aligned}$$

It follows that

$$A_{1,P} \leq \left(\mathbb{E}\left[\mathrm{var}\left(\frac{1}{P}\sum_{p=1}^{P} y_{i,p} y_{j,p} \mid \delta_i, \delta_j\right)\right]\right)^{1/2} \leq \frac{\tau_P}{P}\sigma_{\sup}.$$

Using a similar argument to bound the conditional variance inside $A_{2,P}$, we have

$$\begin{aligned}
&\mathrm{var}\left(\frac{1}{P}\sum_{p=1}^{P} \mu_{i,p} y_{j,p} \mid \delta_i = a, \delta_j = b\right) \\
&\leq \mu_{\sup}^2 \frac{1}{P^2}\sum_{p_1, p_2 = 1}^{P} \left| \mathrm{cov}(y_{i,p_1}, y_{i,p_2})\right| \leq \frac{1}{P^2}\mu_{\sup}^2 \tau_P^2.
\end{aligned}$$

This leads to

$$A_{2,P} \leq \frac{\tau_P}{P}\mu_{\sup},$$

and by a similar argument to A_3, $A_{3,P} \leq \frac{\tau_P}{P}\mu_{\sup}$. Substituting these bounds into (17) we obtain

$$(\mathbb{E} \mid g_{i,j} - \theta_{\delta_i,\delta_j} \mid^2)^{1/2} \leq \frac{\tau_P}{P}(2\mu_{\sup} + \sigma_{\sup}),$$

thus proving (13). We next bound $(\mathbb{E} \mid g_{i,i} - \theta_{\delta_i} \mid^2)^{1/2}$. We use (12) to give

$$(\mathbb{E} \mid g_{i,i} - \theta_{\delta_i} \mid^2)^{1/2} \leq B_{1,P} + B_{2,P} + B_{3,P}, \tag{18}$$

where

$$B_{1,P} = \left(\mathbb{E}\left[\operatorname{var}\left(\frac{1}{P} \sum_{p=1}^{P} y_{i,p}^2 \mid \delta_i \right) \right] \right)^{1/2}$$

$$B_{2,P} = \left(\mathbb{E}\left[\operatorname{var}\left(\frac{1}{P} \sum_{p=1}^{P} \mu_{\delta_i,p} y_{i,p} \mid \delta_i \right) \right] \right)^{1/2}$$

and $$B_{3,P} = \left(\mathbb{E}\left[\operatorname{var}\left(\frac{1}{P} \sum_{p=1}^{P} \mu_{\delta_i,p} y_{i,p} \mid \delta_i \right) \right] \right)^{1/2}.$$

Using the same methods used to bound $A_{2,P}$ and $A_{3,P}$, it is straightforward to show that $B_{2,P}$, $B_{3,P} \leq \tau_P \mu_{\sup}/P$. To bound $B_{1,P}$ we note that

$$\begin{aligned} &\operatorname{var}\left(\frac{1}{P} \sum_{p=1}^{P} y_{j,p}^2 \mid \delta_i = a \right) \\ &\quad = \frac{1}{P^2} \sum_{p_1, p_2 = 1}^{P} \operatorname{cov}[y_{i,p1}^2, y_{i,p2}^2 \mid \delta_i = a] \\ &\quad \leq P^{-2} \kappa_P^2, \end{aligned}$$

which follows from Assumption 2. Thus $B_{1,P} \leq P^{-1} \kappa_P$. Substituting into (18) gives

$$\left(\mathbb{E} \mid g_{i,i} - \theta_{\delta_i})^2 \mid \right)^{1/2} \leq \frac{1}{P}(\kappa_P + 2\tau_P \mu_{\sup}),$$

thus proving (14).

To prove (15), we apply (13) and (14), leading to

$$\begin{aligned} &\mathbb{E}\left\| m_i^{\boldsymbol{\delta}} - \boldsymbol{\theta}_{\delta_i} \right\|_2^2 \\ &\quad = \sum_{i,j=1, i \neq j}^{N} \mathbb{E}(g_{i,j} - \theta_{\delta_i,\delta_j})^2 + \sum_{i=1}^{N} \mathbb{E}(g_{i,i} - \theta_{\delta_i})^2 \\ &\quad \leq (N-1)\frac{1}{P^2}\tau_P^2(2\mu_{\sup} + \sigma_{\sup})^2 \\ &\qquad + \frac{1}{P^2}(\kappa_P + 2\tau_P \mu_{\sup})^2 \\ &\quad \leq \frac{1}{P^2}[(N-1)\tau_P^2(2\mu_{\sup} + \sigma_{\sup})^2 \\ &\qquad + (\kappa_P + 2\tau_P \mu_{\sup})^2], \end{aligned}$$

proving (15). Finally, to prove (16) we note that

$$\mathbb{E}\left\|\mathbf{M}^{\boldsymbol{\delta}} - \boldsymbol{\Theta}\right\|_2^2 = \sum_{i=1}^{N} \mathbb{E}\left\|m_i^{\boldsymbol{\delta}} - \boldsymbol{\theta}_{\delta_i}\right\|_2^2.$$

By substituting (15) in the above we have

$$\mathbb{E}\left\|\mathbf{M}^{\boldsymbol{\delta}} - \boldsymbol{\Theta}\right\|_2^2 \leq \frac{N}{P^2}\left[(N-1)\tau_P^2(2\mu_{\text{sup}} + \sigma_{\text{sup}})^2 + (\kappa_P + 2\tau_P\mu_{\text{sup}})^2\right],$$

which proves (16) and hence Lemma 1.

APPENDIX PERFORMANCE IN SIMULATIONS

To evaluate our method when the underlying “truth” was known, we compared our method to eight other methods. We estimated the number of clusters using a simulation design that was proposed in [23], a similar design was also used in other works, including [3] and [48]. The eight other algorithms tested included extended K-means with BIC (X-KM) [22], affinity propagation (AP) [45], GAP statistics implemented with sparse K-means (GAP+S-KM) [4], [46], a model based clustering with variable selection (CVAR) [2], a high dimensional data clustering (HDDC) [24], a robust graph continuous clustering (RCC) [11], a graph clustering based on tensors (SPEC) [16] and a recent clustering method that performs simultaneous variable selection and estimation of K based on a resampling method (S4) [9]. We used the default parameter settings for each of these methods as implemented in their associated software packages. These included the following R packages: apcluster [49], cvarsel [50], cluster [51], HDclassif [52], sparcl [53], Spectrum [16], and RJcluster [47], as well as the implementation of X-KM in the Python library, ‘pyclustering.cluster.xmeans’. For the S4 and RCC methods, we used code and parameter settings available at S4github and RCCgithub, respectively.

The simulation designs always contained $K_0 = 4$ clusters, and P varied in {100, 200, 500, 1000, 2000}. For each P, we generated 100 replications, and each replication contained 100 observations, with 15 observations taken from cluster 1, 20 observations from cluster 2, 35 observations from cluster 3, and 30 observations from cluster 4. Only 10% of the features were informative in the sense that the distributions of the informative features differed across the clusters. Specifically, five percent of the features in each cluster were generated according to normal distribution with a mean taken from the first row of Table 3 and standard deviation 1, and five percent were generated from a normal distribution with a mean taken from the second row in Table 3 and standard deviation 1. The remaining 90% of features for each cluster were generated as independent $N(0, 1)$ random deviates. The latter features were thus non-informative in identifying the clusters. We used a common value of $\sigma = 1$, as suggested in [23]. This setting corresponds to a high “signal-to-noise ratio (SNR)” scenario. Samples used in this simulation can be obtained from the R package, *RJcluster* [47] using the function

```
simulate_HD_data(size_vector = c(15, 20,
35, 30), p = 100, sparsity = 0.1).
```

TABLE 3.

Means of features in the simulation study.

Features	Cluster 1	Cluster 2	Cluster 3	Cluster 4
First 5%	2.5	0	0	−2.5
Second 5%	1.5	1.5	−1.5	−1.5
Remaining 90%	0	0	0	0

Based on the 100 replications for each *P*, we report the average Adjusted Mutual Index (AMI) and proportion of times the correct cluster number was identified as performance measures. These performance measures are displayed in Figs. 4 and 7. The simulations demonstrate that GMAC provided the highest AMI values and most accurate estimates of the number of clusters for this simulation design.

A. COMPUTATION TIMES

We compared the execution time of our proposed method, GMAC algorithm to other algorithms which *do not require* the prespecification of the number of clusters. The overall distribution of the computation times taken by these algorithms across 32 datasets is displayed in Fig. 5. Execution times are displayed in log seconds. The GMAC, AP, RCC, and SPEC algorithms are much more computationally efficient than GAP, HDDC, S4 and CVAR methods. Excluding the AP algorithm, which provided the best AMI in only 3 microarray dataset, the GMAC algorithm was significantly faster than all of the remaining algorithms.

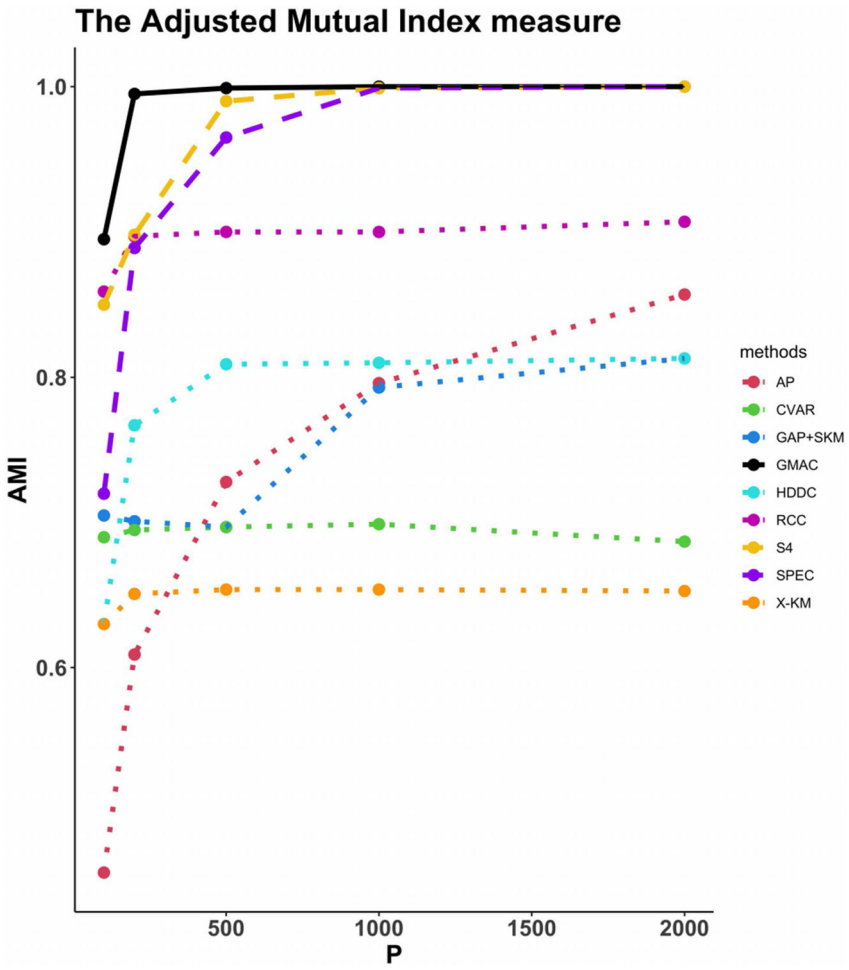


FIGURE 6.

This plot summarizes the average Adjusted Mutual Index (AMI) that each clustering methods achieved over 100 replications of the simulation design for each value of *P*.

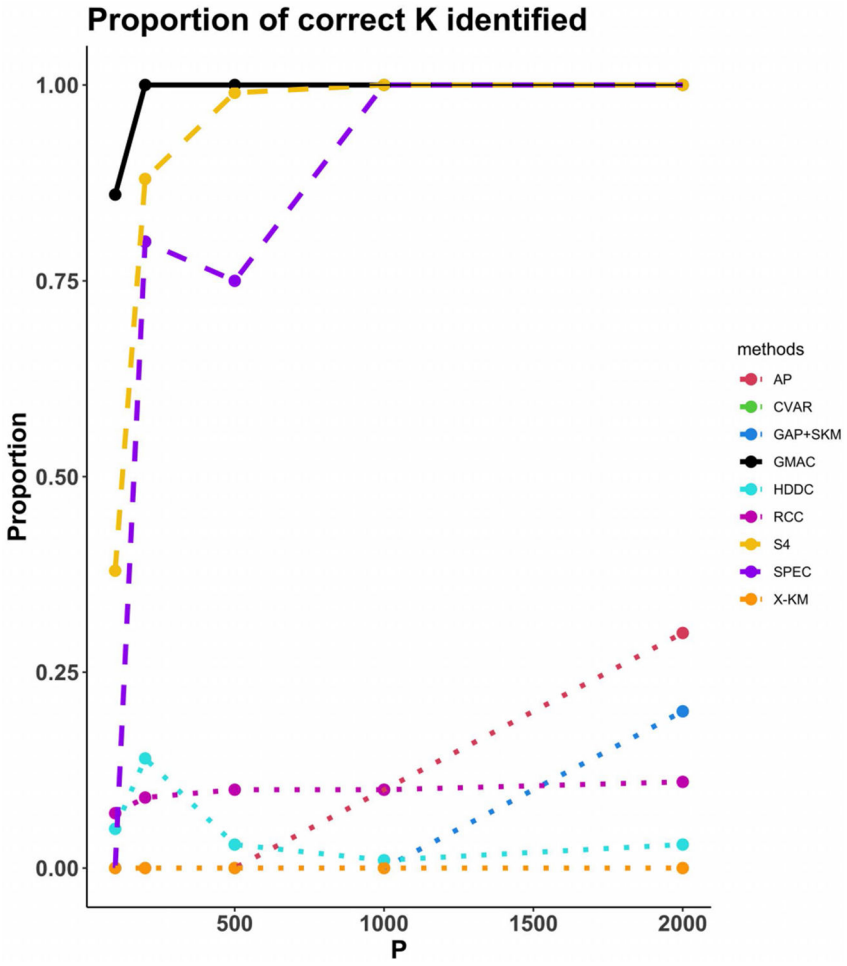


FIGURE 7.
This plot summarizes the proportion of correct cluster number identifications that each clustering method recorded over 100 replications of the simulation design for each value of *P*.

TABLE 4.

Hyperparameters and software used for other methods in the comparative study.

Methods	hyper-parameters	Values	Software
AP	iter.max	100	apcluster (R package)
	s	negDistMat(r=2)	
CVAR	search	headlong	clustvarsel (R package)
	direction	forward	
	parallel	T	
	iter.max	100	
GAP (with partitioning around mediod)	maximum number of clusters	20	cluster (R package)
	d.power	2	

Methods	hyper-parameters	Values	Software
	bootstrap samples	max(100,n)	cluster
	metric	Euclidean	
	iter.max	100	
HDDC	max number of clusters	20	HDclassif (R package)
	model	"ALL"	
	threshold	0.2(default)	
	criterion	bic(default)	
	d_{max}	100(default)	
S-KM	iter.max	100	sparcl (R package)
	wbounds grid	[1, 10](default)	
	nperm	100	
SPEC	method	2(default: multimodal eigen gap)	Spectrum (R package)
	kernel-type	density(default)	
	maxk	20	
	Nearest-Neighbor	7(default)	
	iter.max	100	
S4	lam1	1.5	S4 (R package)
	iteratio	100	
	kvector	2:7	
GMAC	Cmax	20	RJcluster (R package)
	iter.max	100	
	penalty	bic	

B. HYPERPARAMETER SETTINGS OF OTHER CLUSTERING METHODS

We used the same hyperparameters settings recommended in [11] for the following clustering algorithms: **KM**++, **AC-W**, **N-CUT**, **ZELL**, **SEC**, **LDMGI**, **PIC** and **RCC**. Table 4 provides the hyperparameter settings and software used to obtain the results from the other methods we used in addition.

APPENDIX DATA TRANSFORMATIONS IN MICROARRAY GENE STUDIES

We took the logarithmic transformation of all data that contained only positive values. Several data sets were already preprocessed and centered and were therefore not log-transformed further. These data sets included Alizadeh-v1,v2,v3, Bittner, Garber, Lapointe-v1, Liang, Risinger, Singh-v1, Tomlins-v1 and West. For the remaining data sets, after logarithm transformation we standardized by centering on the median and scaling by the standard deviation. All the transformed data on which the clustering algorithms were applied can be found in the "Microarray-data" folder in the RJclust folder provided in the github repository https://github.com/srahman-24/GMac.

Data Availability and Software

The datasets used for the comparisons are available at DataLink. We executed all algorithms on a workstation with an Intel(R) Core(TM) i7-3770 CPU clocked at 3.40GHz with 8.00 GB RAM. The datasets and algorithms that produced the results are available at https://github.com/srahman-24/GMac.

REFERENCES

[1]. Aggarwal CC and Reddy CK, Data Clustering: Algorithms and Applications. Boca Raton, FL, USA: CRC Press, 2013.

[2]. Raftery AE and Dean N, "Variable selection for model-based clustering," J. Amer. Stat. Assoc, vol. 101, no. 473, pp. 168–178, Mar. 2006.

[3]. Pan W and Shen X, "Penalized model-based clustering with application to variable selection," J. Mach. Learn. Res, vol. 8, no. 5, 2007.

[4]. Witten DM and Tibshirani R, "A framework for feature selection in clustering," J. Amer. Stat. Assoc, vol. 105, no. 490, pp. 713–726, 2010. [PubMed: 20811510]

[5]. Chi EC, Allen GI, and Baraniuk RG, "Convex biclustering," Biometrics, vol. 73, no. 1, pp. 10–19, Mar. 2017. [PubMed: 27163413]

[6]. Wang B, Zhang Y, Sun WW, and Fang Y, "Sparse convex clustering," J. Comput. Graph. Statist, vol. 27, no. 2, pp. 393–403, Apr. 2018.

[7]. Brodinová Š, Filzmoser P, Ortner T, Breiteneder C, and Rohm M, "Robust and sparse K-means clustering for high-dimensional data," Adv. Data Anal. Classification, vol. 13, pp. 905–932, Mar. 2019.

[8]. Chakraborty S and Das S, "Detecting meaningful clusters from high-dimensional data: A strongly consistent sparse center-based clustering approach," IEEE Trans. Pattern Anal. Mach. Intell, vol. 44, no. 6, pp. 2894–2908, Jun. 2020.

[9]. Li Y, Zeng X, Lin C, and Tseng GC, "Simultaneous estimation of cluster number and feature sparsity in high-dimensional cluster analysis," Biometrics, vol. 78, no. 2, pp. 574–585, Jun. 2022. [PubMed: 33621349]

[10]. Nie F, Xu D, Tsang IW, and Zhang C, "Spectral embedded clustering," in Proc. IJCAI, 2009, pp. 1181–1186.

[11]. Shah SA and Koltun V, "Robust continuous clustering," Proc. Nat. Acad. Sci. USA, vol. 114, no. 37, pp. 9814–9819, 2017. [PubMed: 28851838]

[12]. Xie J, Girshick R, and Farhadi A, "Unsupervised deep embedding for clustering analysis," in Proc. Int. Conf. Mach. Learn., 2016, pp. 478–487.

[13]. McConville R, Santos-Rodriguez R, Piechocki RJ, and Craddock I, "N2D: (Not too) deep clustering via clustering the local manifold of an autoencoded embedding," in Proc. 25th Int. Conf. Pattern Recognit. (ICPR), Jan. 2021, pp. 5145–5152.

[14]. Ng AY, Jordan MI, and Weiss Y, "On spectral clustering: Analysis and an algorithm," in Proc. Adv. Neural Inf. Process. Syst, 2002, pp. 849–856.

[15]. Shi J and Malik J, "Normalized cuts and image segmentation," IEEE Trans. Pattern Anal. Mach. Intell, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[16]. John CR and Watson D, "Spectrum: Fast adaptive spectral clustering for single and multi-view data. R package version 1.1," 2020.

[17]. von Luxburg U, "A tutorial on spectral clustering," Statist. Comput, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[18]. Wang J, "Consistent selection of the number of clusters via crossvalidation," Biometrika, vol. 97, no. 4, pp. 893–904, Dec. 2010.

[19]. Fang Y and Wang J, "Selection of the number of clusters via the bootstrap method," Comput. Statist. Data Anal, vol. 56, no. 3, pp. 468–477, Mar. 2012.

[20]. Fan X, Yue Y, Sarkar P, and Wang YR, “On hyperparameter tuning in general clustering problemsm,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 2996–3007.

[21]. Preud’homme G, Duarte K, Dalleau K, Lacomblez C, Bresso E, Smaïl-Tabbone M, Couceiro M, Devignes M-D, Kobayashi M, Huttin O, Ferreira JP, Zannad F, Rossignol P, and Girerd N, “Head-to-head comparison of clustering methods for heterogeneous data: A simulation-driven benchmark,” Sci. Rep, vol. 11, no. 1, pp. 1–14, Dec. 2021. [PubMed: 33414495]

[22]. Dan P and Moore AW, “X-means: Extending K-means with efficient estimation of the number of clusters,” in Proc. ICML, vol. 1, Jun. 2000, pp. 727–734.

[23]. Guo J, Levina E, Michailidis G, and Zhu J, “Pairwise variable selection for high-dimensional model-based clustering,” Biometrics, vol. 66, no. 3, pp. 793–804, Sep. 2010. [PubMed: 19912170]

[24]. Bouveyron C, Girard S, and Schmid C, “High-dimensional data clustering,” Comput. Statist. Data Anal, vol. 52, no. 1, pp. 502–519, 2007.

[25]. Armstrong SA, Staunton JE, Silverman LB, Pieters R, den Boer ML, Minden MD, Sallan SE, Lander ES, Golub TR, and Korsmeyer SJ, “MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia,” Nature Genet., vol. 30, no. 1, p. 41, 2002. [PubMed: 11731795]

[26]. Maaten LVD and Hinton G, “Visualizing data using t-SNE,” J. Mach. Learn. Res, vol. 9, no. 11, pp. 1–27, 2008.

[27]. McInnes L, Healy J, and Melville J, “UMAP: Uniform manifold approximation and projection for dimension reduction,” 2018, arXiv:1802.03426.

[28]. Dempster AP, Laird NM, and Rubin DB, “Maximum likelihood from incomplete data via the EM algorithm,” J. Roy. Stat. Soc. B, Methodol, vol. 39, pp. 1–38, Sep. 1977.

[29]. Schwarz G, “Estimating the dimension of a model,” Ann. Statist, vol. 6, no. 2, pp. 461–464, 1978.

[30]. McLachlan G and Peel D, Finite Mixture Models. Hoboken, NJ, USA: Wiley, 2004.

[31]. Scrucca L, Fop M, Murphy TB, and Raftery AE, “Mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models,” R J, vol. 8, no. 1, pp. 289–317, 2016. [PubMed: 27818791]

[32]. Fraley C and Raftery AE, “Model-based clustering, discriminant analysis, and density estimation,” J. Amer. Statist. Assoc, vol. 97, no. 458, pp. 611–631, 2002.

[33]. Cover TM and Thomas JA, “Entropy, relative entropy and mutual information,” Elements Inf. Theory, vol. 2, no. 1, pp. 12–13, 1991.

[34]. Vinh NX, Epps J, and Bailey J, “Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance,” J. Mach. Learn. Res, vol. 11, pp. 2837–2854, Oct. 2010.

[35]. Tian L, Dong X, Freytag S, Cao K-AL, Su S, JalalAbadi A, Amann-Zalcenstein D, Weber TS, Seidi A, Jabbari JS, Naik SH, and Ritchie ME, “Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments,” Nature Methods, vol. 16, no. 6, pp. 479–487, Jun. 2019. [PubMed: 31133762]

[36]. de Souto MCP, Costa IG, de Araujo DSA, Ludermir TB, and Schliep A, “Clustering cancer gene expression data: A comparative study,” BMC Bioinform., vol. 9, no. 1, p. 497, 2008.

[37]. de Souto MCP, de Araujo DSA, Costa IG, Soares RGF, Ludermir TB, and Schliep A, “Comparative study on normalization procedures for cluster analysis of gene expression datasets,” in Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intelligence), Jun. 2008, pp. 2792–2798.

[38]. Arthur D and Vassilvitskii S, “K-means++: The advantages of careful seeding,” in Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms, 2007, pp. 1027–1035.

[39]. Bezdek JC, Pattern Recognition With Fuzzy Objective Function Algorithms. New York, NY, USA: Springer, 2013.

[40]. Comaniciu D and Meer P, “Mean shift: A robust approach toward feature space analysis,” IEEE Trans. Pattern Anal. Mach. Intell, vol. 24, no. 5, pp. 603–619, Aug. 2002.

[41]. Zhao D and Tang X, “Cyclizing clusters via zeta function of a graph,” in Proc. Adv. Neural Inf. Process. Syst, 2009, pp. 1953–1960.

[42]. Yang Y, Xu D, Nie F, Yan S, and Zhuang Y, “Image clustering using local discriminant models and global integration,” IEEE Trans. Image Process, vol. 19, no. 10, pp. 2761–2773, Oct. 2010. [PubMed: 20423802]

[43]. Zhang W, Zhao D, and Wang X, “Agglomerative clustering via maximum incremental path integral,” Pattern Recognit., vol. 46, no. 11, pp. 3056–3065, Nov. 2013.

[44]. Elhamifar E and Vidal R, “Sparse subspace clustering: Algorithm, theory, and applications,” IEEE Trans. Pattern Anal. Mach. Intell, vol. 35, no. 11, pp. 2765–2781, Mar. 2013. [PubMed: 24051734]

[45]. Frey BJ and Dueck D, “Clustering by passing messages between data points,” Science, vol. 315, no. 5814, pp. 972–976, Feb. 2007. [PubMed: 17218491]

[46]. Tibshirani R, Walther G, and Hastie T, “Estimating the number of clusters in a data set via the gap statistic,” J. Roy. Stat. Soc., B, Stat. Methodol, vol. 63, no. 2, pp. 411–423, 2001.

[47]. Rahman S, Valen Johnson E, Rao S, and Shudde R, “RJcluster: A fast clustering algorithm for high dimensional data based on the gram matrix decomposition. R package version 3.2.4,” 2021.

[48]. Friedman JH and Meulman JJ, “Clustering objects on subsets of attributes (with discussion),” J. Roy. Stat. Soc., B, Stat. Methodol, vol. 66, no. 4, pp. 815–849, Nov. 2004.

[49]. Bodenhofer U, Kothmeier A, and Hochreiter S, “APCluster: An R package for affinity propagation clustering,” Bioinformatics, vol. 27, no. 17, pp. 2463–2464, 2011. [PubMed: 21737437]

[50]. Scrucca L and Raftery AE, “Clustvarsel: A package implementing variable selection for Gaussian model-based clustering in R,” J. Stat. Softw, vol. 84, no. 1, pp. 1–28, 2018. [PubMed: 30450020]

[51]. Maechler M, Rousseeuw P, Struyf A, Hubert M, and Hornik K, “Cluster: Cluster analysis basics and extensions. R package version 2.1.3,” 2022.

[52]. Bergé L, Bouveyron C, and Girard S, “HDclassif: An R package for model-based clustering and discriminant analysis of high-dimensional data,” J. Statist. Softw, vol. 46, no. 6, pp. 1–29, 2012.

[53]. Witten MD and Tibshirani R, “SPARCL: Perform sparse hierarchical clustering and sparse K-means clustering,” Tech. Rep., R package version 1.0.4, 2018.

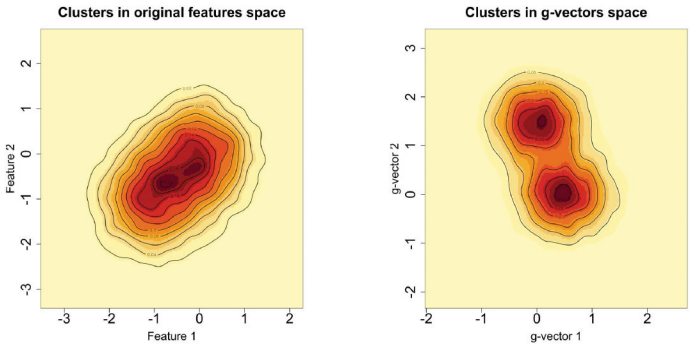


FIGURE 1.

(a) Contour plot of the simulated distribution of the original feature vectors for object 1, $\boldsymbol{x}_1^T = (\boldsymbol{x}_{11}, \boldsymbol{x}_{12})$ drawn from $\mathcal{N}$ ([0, 0], $0.5I_2$) and object 2, $\boldsymbol{x}_2^T = (\boldsymbol{x}_{21}, \boldsymbol{x}_{22})$ drawn from $\mathcal{N}$ ([−1, −1], $0.5I_2$). (b) The contour plot of the distribution of the simulated *g-vectors* corresponding to object 1 and 2.

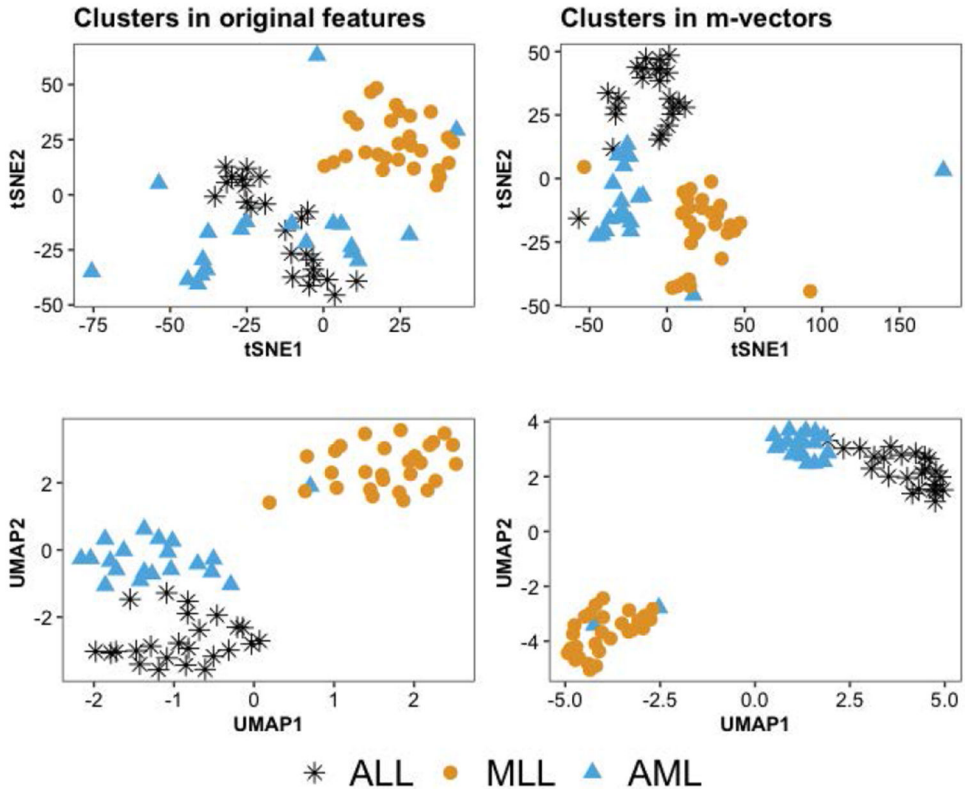


FIGURE 2.

Application of two popular non-linear dimension reduction techniques, t-SNE (top) and UMAP (bottom) to original gene expression vectors (left) and *m-vectors* (right). The tumor clusters labels, ALL, AML, and MLL, are well-established and validated in [25].

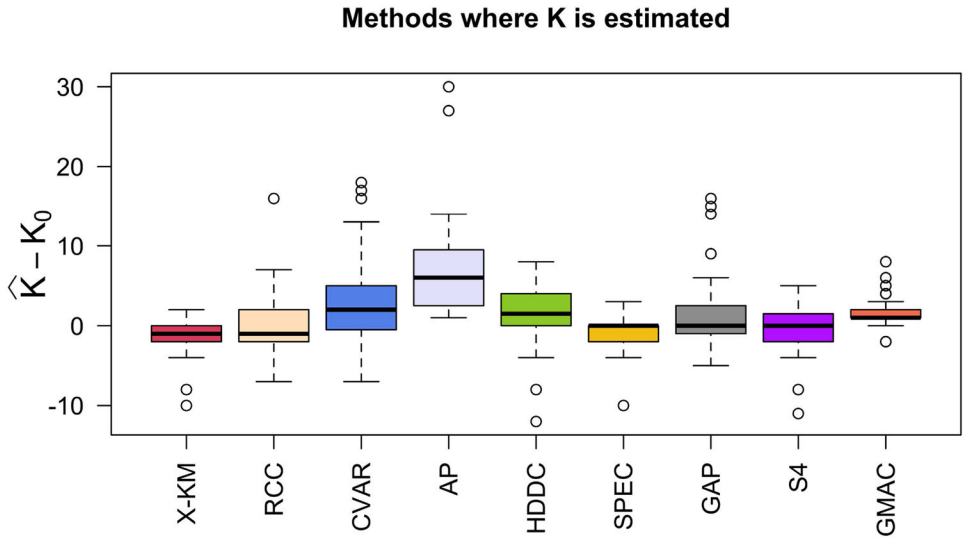


FIGURE 3.

Boxplot displays of the distribution of $(\widehat{\boldsymbol{K}} - \boldsymbol{K_0})$ for each of the clustering methods considered in Table 2. Here, $\widehat{\boldsymbol{K}}$ is the estimated number of clusters and K_0 the validated number of clusters. Detailed information of K_0 and $\widehat{\boldsymbol{K}}$ for each method is provided in Table 2. In addition to GMAC, X-KM, CVAR, and HDDC (defined in Section V-A) also used BIC to select the number of clusters. From the boxplot and Table 2, we see that GMAC provided the best estimates of K_0, followed closely by X-KM.

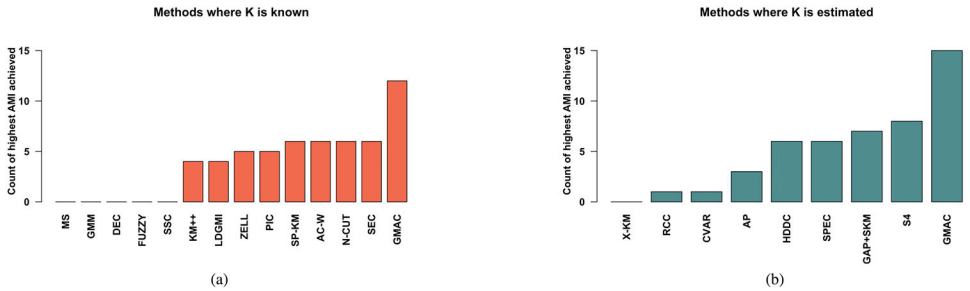


FIGURE 4.

(a) Barplot summaries of the AMI results for the clustering algorithms displayed in Table 1 where the number of clusters, K_0, was assumed to be known. The plot shows how frequently each of the 14 clustering algorithms obtained the highest AMI across 32 gene expression data sets. When K_0 is known, GMAC achieved the highest AMI for 12 out of 32 datasets. Abbreviations for algorithms are provided in Section V-A. (b) Barplot summaries of the AMI results for the clustering algorithms displayed in Table 2 when K_0 is estimated. The plot shows how frequently each of the 10 clustering algorithms obtained the highest AMI across 32 gene expression data sets. When K_0 is unknown, GMAC achieved the highest AMI for 15 out of 32 datasets.

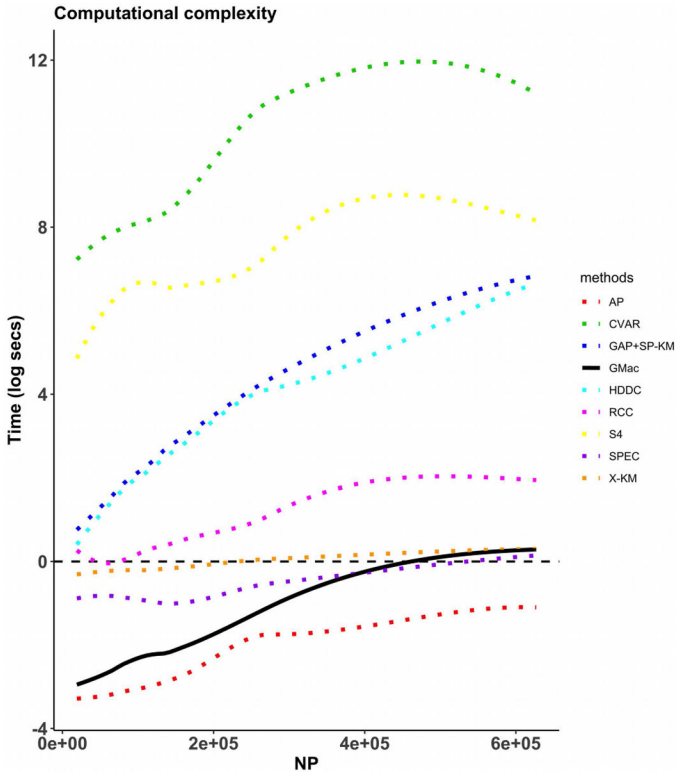


FIGURE 5.

This plot illustrates the average computational speed of the clustering algorithms that estimate the number of clusters. Computational times are displayed in *log seconds* (with base e) as a function of NP.

TABLE 1.

Adjusted Mutual Information (AMI) displayed for 9 clustering algorithms compared over 32 different microarray datasets. For brevity, results from 5 additional clustering algorithms are not shown because none of them produced the highest AMI for any of the datasets considered here. Here, the clustering algorithms used the true number of clusters K_0. For each dataset, the maximum achieved AMI is highlighted in bold. The number of times each method achieves the highest AMI is reported in the last row.

				K_0 is fixed								
Datasets	**N**	**P**	K_0	**KM++**	**SP-KM**	**AC-W**	**N-Cuts**	**ZELL**	**SEC**	**LDGMI**	**PIC**	**GMAC**
Alizadeh-v1	42	1097	2	0.340	−0.015	0.101	0.096	0.250	0.238	0.123	0.033	**0.660**
Alizadeh-v2	62	2095	3	0.568	0.872	0.922	0.922	0.922	0.922	0.738	0.922	**1**
Alizadeh-v3	62	2095	4	0.586	0.649	0.616	0.601	**0.702**	0.574	0.582	0.625	0.649
Armstrong-v1	72	1083	2	**0.372**	**0.370**	0.308	**0.372**	0.308	0.323	0.355	0.308	**0.372**
Armstrong-v2	72	2196	3	**0.891**	0.375	0.746	0.83	0.802	**0.891**	0.509	0.802	0.678
Bhattacharjee	203	1545	5	0.444	0.296	**0.601**	0.563	0.496	0.570	0.378	0.378	0.395
Bittner	38	2203	2	−0.012	**0.195**	0.002	0.042	0.115	−0.002	0.014	0.115	0.002
Bredel	28	1072	3	0.297	0.000	**0.384**	0.203	0.278	0.259	0.295	0.278	0.000
Chowdary	104	184	2	0.764	0.595	**0.859**	**0.859**	**0.859**	**0.859**	**0.859**	**0.859**	**0.859**
Dyrskjot	40	1205	5	0.507	0.760	0.474	0.303	0.269	0.389	0.385	0.177	**0.821**
Garber	66	4555	4	0.242	0.026	0.210	0.204	0.246	0.200	0.191	0.246	0.063
Golub-v1	72	1870	2	0.688	0.701	**0.831**	0.650	0.615	0.615	0.615	0.615	0.620
Golub-v2	72	1870	3	0.680	0.617	**0.737**	0.693	0.689	0.703	0.600	0.689	0.495
Gordon	181	1628	2	0.651	**0.937**	0.483	0.681	−0.005	0.791	0.669	0.664	**0.937**
Laiho	37	2204	2	0.007	0.062	−0.007	0.030	0.073	−0.007	0.093	0.044	**0.203**
Lapointe-v1	69	1627	3	0.088	0.012	0.151	**0.179**	0.151	0.088	0.149	0.151	0.158
Lapointe-v2	110	2498	4	0.008	0.097	0.033	0.153	0.147	0.028	0.118	**0.171**	0.151
Liang	37	1413	3	**0.301**	**0.301**	**0.301**	**0.301**	**0.301**	**0.301**	**0.301**	**0.301**	**0.301**
Nutt-v1	50	1379	4	0.171	**0.311**	0.159	0.156	0.109	0.086	0.078	0.113	0.188
Nutt-v2	28	1072	2	−0.025	**0.000**	−0.024	−0.025	−0.031	−0.025	−0.027	−0.030	**0.000**
Nutt-v3	22	1154	2	0.063	0.000	0.004	0.080	0.059	0.080	**0.174**	0.059	**0.174**
Pomeroy-v1	34	859	2	0.012	−0.032	−0.020	−0.006	−0.020	**0.008**	−0.026	−0.032	0.061

				K_0 is fixed							
Datasets	**N**	**P**	K_0	**KM++**	**SP-KM**	**AC-W**	**N-Cuts**	**ZELL**	**SEC**	**LDGMI**	**PIC**	**GMAC**
Pomeroy-v2	42	1381	5	0.502	0.576	0.591	**0.617**	0.568	0.577	0.602	0.568	0.450
Ramaswamy	190	1365	14	0.618	0.401	0.623	0.651	0.618	0.620	**0.663**	0.639	0.493
Risinger	42	1773	4	0.210	0.162	0.297	0.223	0.201	0.258	0.153	0.201	**0.393**
Shipp-v1	77	800	2	**0.264**	0.035	0.208	0.132	−0.002	0.168	0.203	−0.002	0.042
Singh	102	341	2	0.048	0.037	0.019	0.033	−0.003	0.069	−0.003	**0.066**	0.016
Su	174	1573	10	0.666	0.672	0.662	**0.738**	0.687	0.650	0.667	0.660	**0.739**
Tomlins-v1	104	2317	5	0.396	0.382	0.454	0.409	**0.647**	0.469	0.419	0.590	0.352
Tomlins-v2	92	1290	4	0.368	0.222	0.215	0.292	0.226	**0.383**	0.354	0.311	0.177
West	49	1200	2	0.489	0.403	0.489	0.442	**0.515**	0.489	0.442	**0.515**	0.401
Yeohv2	248	2528	6	0.385	0.002	0.383	0.479	0.530	**0.550**	0.337	0.442	0.137
Count of Highest AMI				**4**	**6**	**6**	**6**	**5**	**6**	**4**	**5**	**12**

TABLE 2.

Adjusted Mutual Information and the number of estimated clusters displayed as AMI ($\widehat{K}$) for 9 clustering algorithms compared over 32 different microarray datasets. Here, the clustering algorithms estimated the true number of clusters K_0. For each dataset, the maximum achieved AMI is highlighted in bold. The number of times each method achieves the highest AMI is reported in the penultimate row. The last row displays the square root of the average squared error in estimating the number of clusters.

				Methods (K_0 is estimated)								
Datasets	**N**	**P**	K_0	**X-KM**	**RCC**	**CVAR**	**AP**	**HDDC**	**SPEC**	**GAP+S-KM**	**S4**	**GMAC**
Alizadeh-v1	42	1097	2	0.000 (1)	0.003 (2)	−0.006 (1)	0.211 (6)	0.133 (8)	0.157 (2)	0.000 (1)	0.123 (7)	**0.660** (2)
Alizadeh-v2	62	2095	3	0.757 (4)	0.608 (6)	0.533 (3)	0.563 (6)	0.571 (6)	0.753 (4)	**0.872** (3)	0.856 (2)	0.818 (4)
Alizadeh-v3	62	2095	4	0.636 (2)	0.496 (6)	0.295 (3)	0.540 (6)	0.548 (6)	0.609 (4)	**0.648** (4)	**0.649** (2)	**0.649** (4)
Armstrong-v1	72	1083	2	0.338 (2)	0.000 (1)	0.302 (9)	0.381 (8)	0.461 (3)	0.617 (3)	**0.611** (3)	0.355 (2)	**0.612** (3)
Armstrong-v2	72	2196	3	0.660 (2)	0.000 (1)	0.513 (7)	0.586 (8)	−0.01 (2)	0.693 (3)	**0.803** (3)	0.718 (2)	0.611 (4)
Bhattacharjee	203	1545	5	0.401 (3)	0.000 (1)	0.17 (15)	0.38 (17)	0.269 (2)	0.505 (3)	**0.542** (8)	0.372 (3)	0.475 (9)
Bittner	38	2203	2	0.016 (2)	0.156 (4)	−0.02 (3)	0.243 (9)	0.288 (6)	0.013 (2)	0.000 (1)	**0.453** (7)	0.125 (5)
Bredel	28	1072	3	0.000 (1)	0.060 (2)	−0.03 (4)	0.139 (4)	0.227 (6)	**0.356** (3)	0.116 (3)	0.203 (5)	0.144 (4)
Chowdary	104	184	2	0.646 (3)	0.583 (6)	0.448 (5)	0.44 (11)	**0.859** (2)	0.575 (5)	0.43 (16)	0.649 (3)	0.483 (7)
Dyrskjot	40	1205	5	−0.002 (3)	0.000 (1)	0.40 (10)	0.558 (9)	0.607 (9)	0.629 (3)	0.403 (2)	0.403 (2)	**0.766** (6)
Garber	66	4555	4	0.037 (2)	0.10 (20)	0.175 (2)	0.27 (10)	0.164 (5)	0.137 (2)	0.35 (19)	**0.531** (3)	0.063 (4)
Golub-v1	72	1870	2	0.065 (2)	0.000 (1)	0.62 (15)	0.43 (11)	0.478 (3)	0.137 (2)	0.044 (7)	**0.706** (5)	0.620 (3)
Golub-v2	72	1870	3	0.097 (2)	0.000 (1)	0.14 (20)	0.52 (11)	0.478 (3)	0.352 (2)	0.000 (7)	**0.617** (5)	0.463 (3)
Gordon	181	1628	2	0.657 (3)	0.009 (5)	0.248 (3)	0.30 (29)	0.003 (2)	**0.937** (2)	0.44 (11)	**0.937** (2)	0.466 (8)
Laiho	37	2204	2	−0.02 (2)	0.000 (1)	0.03 (18)	0.061 (6)	**0.16** (10)	0.036 (2)	0.000 (1)	0.031 (6)	**0.165** (3)
Lapointe-v1	69	1627	3	0.012 (2)	−0.004 (4)	0.158 (3)	0.152 (5)	**0.253** (9)	0.012 (2)	0.216 (3)	0.216 (3)	0.158 (5)
Lapointe-v2	110	2498	4	−0.005 (2)	0.002 (3)	0.133 (3)	0.210 (7)	0.138 (2)	−0.006 (2)	0.216 (6)	−0.006 (2)	**0.249** (8)
Liang	37	1413	3	0.294 (3)	0.239 (3)	0.432 (6)	**0.481** (4)	0.301 (3)	**0.481** (4)	**0.481** (4)	**0.481** (4)	**0.481** (4)
Nutt-v1	50	1379	4	0.113 (2)	0.075 (2)	0.112 (6)	0.116 (6)	0.284 (9)	0.215 (3)	0.000 (1)	0.363 (7)	**0.459** (6)
Nutt-v2	28	1072	2	−0.024 (2)	0.060 (2)	−0.016 (4)	0.15 (10)	**0.250** (3)	−0.027 (4)	0.116 (3)	**0.255** (5)	**0.256** (4)
Nutt-v3	22	1154	2	0.430 (2)	0.000 (1)	0.225 (4)	−0.002 (4)	0.321 (5)	0.511 (2)	0.000 (1)	0.259 (2)	**0.645** (4)
Pomeroy-v1	34	859	2	0.021 (2)	0.000 (1)	0.06 (20)	0.061 (4)	**0.118** (7)	−0.014 (2)	−0.032 (2)	−0.032 (2)	0.061 (4)

				Methods (K_0 is estimated)								
Datasets	**N**	**P**	K_0	**X-KM**	**RCC**	**CVAR**	**AP**	**HDDC**	**SPEC**	**GAP+S-KM**	**S4**	**GMAC**
Pomeroy-v2	42	1381	5	0.327 (2)	0.000 (1)	**0.606** (8)	**0.606** (8)	0.513 (9)	0.544 (3)	0.362 (2)	0.473 (4)	0.396 (3)
Ramaswamy	190	1365	14	0.401 (4)	**0.68** (20)	0.184 (7)	0.59 (25)	0.246 (2)	0.547 (4)	0.43 (13)	0.495 (3)	0.44 (15)
Risinger	42	1773	4	0.036 (2)	−0.005 (2)	0.148 (3)	0.309 (6)	**0.377** (9)	0.308 (3)	0.000 (1)	0.039 (2)	**0.378** (4)
Shipp-v1	77	800	2	0.037 (2)	0.000 (1)	0.03 (13)	0.11 (10)	0.038 (4)	0.069 (4)	0.089 (3)	0.098 (3)	**0.133** (3)
Singh	102	341	2	0.038 (4)	0.034 (2)	0.030 (5)	0.08 (12)	0.000 (1)	0.029 (3)	**0.10** (18)	0.050 (2)	**0.10** (10)
Su	174	1573	10	0.311 (2)	0.002 (3)	0.59 (11)	0.66 (20)	0.381 (2)	**0.824** (6)	0.73 (16)	0.424 (2)	0.70 (11)
Tomlins-v1	104	2317	5	0.034 (2)	0.06 (12)	0.182 (2)	0.37 (19)	0.000 (1)	**0.485** (3)	0.000 (1)	0.356 (4)	0.352 (6)
Tomlins-v2	92	1290	4	0.129 (2)	0.006 (7)	0.132 (3)	0.19 (18)	0.051 (8)	0.166 (2)	0.000 (1)	0.157 (2)	**0.206** (6)
West	49	1200	2	−0.001 (2)	0.000 (1)	0.097 (2)	0.26 (10)	0.202 (8)	**0.413** (2)	0.000 (1)	0.403 (3)	**0.413** (3)
Yeohv2	248	2528	6	0.006 (2)	0.000 (1)	0.027 (11)	**0.41** (36)	0.229 (6)	0.172 (2)	0.000 (1)	0.001 (2)	0.042 (4)
Count of Highest AMI				**0**	**1**	**1**	**3**	**6**	**6**	**7**	**8**	**15**
$\sqrt{\text{Avg}(\widehat{K}-K_0)^2}$				2.77	4.11	6.83	9.92	4.29	**2.52**	5.43	3.26	**2.57**