M3C: Monte Carlo reference-based consensus clustering

Christopher R. John[1*], David Watson[2], Dominic Russ[1], Katriona Goldmann[1], Michael Ehrenstein[3], Costantino Pitzalis[1], Myles Lewis[1], Michael Barnes[1]

[1]Experimental Medicine and Rheumatology, William Harvey Research Institute, Bart's and The London School of Medicine and Dentistry, Queen Mary University of London, Charterhouse Square, London, EC1M 6BQ, United Kingdom

[2]Oxford Internet Institute, University of Oxford, 1 St. Giles, Oxford, OX1 3JS

[3]Rayne Institute, University College London, 5 University Street, London, WC1E 6JF, United Kingdom

*To whom correspondence should be addressed

Abstract

Genome-wide data is used to stratify patients into classes for precision medicine using clustering algorithms. A common problem in this area is selection of the number of clusters (K). The Monti consensus clustering algorithm is a widely used method which uses stability selection to estimate K. However, the method has bias towards higher values of K and yields high numbers of false positives. As a solution, we developed Monte Carlo reference-based consensus clustering (M3C), which is based on this algorithm. M3C simulates null distributions of stability scores for a range of K values thus enabling a comparison with real data to remove bias and statistically test for the presence of structure. M3C corrects the inherent bias of consensus clustering as demonstrated on simulated and real expression data from The Cancer Genome Atlas (TCGA). For testing M3C, we developed clusterlab, a new method for simulating multivariate Gaussian clusters.

Contact: christopher.john@qmul.ac.uk or m.r.barnes@qmul.ac.uk

Introduction

Stratified medicine is the concept that patients may be clustered into classes to personalise patient therapy. Increasingly, patient genome-wide expression data is being used to perform clustering[1-6] . Cluster analysis of genome-wide data (e.g. transcriptomics, epigenomics, proteomics, and DNA copy number) has been shown to identify tumour subtypes with distinct clinical outcomes in cancer research[1-6], and is starting to be applied on other diseases as well[7-9]. Therefore, there is high demand for methods that deliver robust results. Broadly, the clustering problem may be broken down into two steps: select K and separate the data into K groups. The order of these steps varies by clustering algorithm – K must be defined upfront in *k*-means, for instance, while it is defined afterwards in hierarchical clustering. In this study, our primary focus was to develop a method for estimating the optimal K.

Numerous methods have been proposed for estimating K, such as: Monti et al. consensus clustering[10], the GAP-statistic[11], CLEST[12], and progeny clustering[13]. The concept behind consensus clustering is that the ideal clusters should be stable despite resampling. Therefore, the degree of cluster stability for each value of K can be measured to estimate the optimal K. Şenbabaoğlu et al. made a useful contribution by demonstrating that false positive structures could be found in K=1 null data using the Monti consensus clustering algorithm[14], this is a common problem in cluster analysis. The authors suggested to generate null datasets with the same gene-gene correlation structure as the real data to evaluate cluster strength. However, they did not provide a method for performing a formal hypothesis test. They developed a new metric that measures cluster stability called the proportion of ambiguous clustering (PAC) score, this is better able to estimate K than the original delta K metric[10] proposed by Monti et al. However, the PAC score does not take into account null

reference distributions, has inherant bias towards higher values of K, and does not test the null hypothesis K=1.

Our aim was to solve these problems by enhancing the Monti consensus clustering algorithm to include a Monte Carlo reference procedure to eliminate bias towards higher values of K and to test the null hypothesis K=1. This method we call M3C (https://www.bioconductor.org/packages/3.7/bioc/html/M3C.html). To introduce M3C, it is instructive to define the hypotheses that it tests. M3C calculates null distributions of PAC scores for each K (starting with K=2) by simulating K=1 null datasets. For each K, this allows us to formally test the following null hypothesis:

H_0: the PAC score comes from a single Gaussian cluster

The alternative hypothesis tested for each K is:

H_A: the PAC score does not come from a single Gaussian cluster

If no p values are significant along the range of K we accept the null hypothesis H_0 in every case, this means there is no significant evidence for clusters in the data. If a p value is significant, then we can reject the null hypothesis H_0, thereby accepting H_A, this is significant evidence for clusters in the data. M3C presented us with an opportunity to test two hypotheses on real data. First, that pre-existing high-profile publications contain results that declare evidence of structure when in fact there is none. Second, that not considering reference distributions when deciding K leads to systematic bias in the Monti consensus clustering method. The results in this manuscript imply a more rigorous approach is required.

Results

Systematic bias detected in two widely applied consensus clustering methods

Using clusterlab (see Methods for details), we first generated a null dataset where no genuine clusters are found (Fig. 1a). Next, we tested the Monti consensus clustering algorithm on this data, the cumulative distribution function (CDF) plot corresponding to the consensus matrices from K = 2 to K = 10 for the null dataset demonstrates that as K increases the consensus matrices inherently become more stable (indicated by a flatter line) (Fig. 1b). The PAC scores, which measure the CDF plot flatness, steadily decreased with increasing K estimating an optimal K of ten (Fig. 1c). A similar but reversed effect was observed in the cophenetic metric of Nonnegative Matrix Factorisation (NMF) consensus clustering[15], which estimates an optimal K of two (Fig. 1d). Therefore, consensus clustering and NMF consensus clustering show bias towards higher and lower values of K, respectively. Both methods also declare evidence of structure when it does not exist, due to not comparing against null reference distributions. To demonstrate the functionality of clusterlab, we generated a ring of four Gaussian clusters, four clusters with varying variance, and a more complex multi-ringed structure consisting of 25 Gaussian clusters (Supplementary Fig. 1).

M3C can find K and evaluate the significance of its decision

We provide an overview of our method in Figure 2a. For our initial investigations, we tested M3C on a negative control, a simulated dataset in which K = 1 (Fig. 2b). The Relative Cluster Stability Index (RCSI) could not distinguish real from false structure. In contrast, the calculation of Monte Carlo p values by M3C correctly suggested there was no structure in this negative control dataset (alpha = 0.05), and no bias towards higher values of K was observed. Next, M3C was tested on a positive control dataset with four simulated clusters (Fig. 2c). The PAC score and the RCSI correctly identified

four as the optimal value of K. A very low Monte Carlo p-value was found by M3C for K = 4 ($p = 9.95 \times 10^{-21}$), this correctly implies that this is the optimal K and means we can reject the null hypothesis H_0.

Next, we reanalysed a range of high-profile stratified medicine datasets where structure had been declared to test for false positive structures (Table 1 & Supplementary Table 1). Because of the ease of data availability, these were predominately, but not exclusively, from TCGA. Table 1 demonstrates the pervasive use of consensus clustering and NMF consensus clustering in the field. Using M3C, we identified two datasets in which no significant evidence against the null hypothesis could be detected. First, a systemic lupus erythematosus (SLE) microarray dataset was analysed where seven major subtypes were reported using hierarchical clustering and dendrogram cutting. However, none of the p-values along the range of K calculated by M3C reached statistical significance (the lowest was for K = 3, p = 0.15) (Fig. 2d). Second, a breast cancer miRNA-seq dataset was identified with no significant evidence of structure (the lowest p value was for K = 4, p = 0.27), whereas seven subtypes were originally reported using NMF (Fig. 2e). These findings imply that false positive structures exist in the literature through not comparing against reference datasets.

Table I: Datasets selected for assessment using M3C and optimal K decisions. HC refers to hierarchical clustering and CC to Monti consensus clustering.

Publication	Year	Data type	Original algorithm	Original K	M3C K
Glioblastoma[3]	2008	Microarray	CC	4	4
Ovarian carcinoma[4]	2011	Microarray	NMF	4	5
Lung cancer[5]	2012	RNA-seq	NMF	4	2
Breast cancer[16]	2012	miRNA-seq	NMF	7	1
Diffuse glioma[1]	2016	RNA-seq	CC	4	8
Lupus[9]	2016	Microarray	HC	7	1
Pheochromocytoma[2]	2017	RNA-seq	CC	4	6

Demonstration of the M3C method on TCGA gene expression data

114 Of those datasets that exhibited significant evidence of structure using M3C, we used this as an
115 opportunity to contrast the clarity of the M3C results with those from consensus clustering with the
116 PAC score, the NMF cophenetic coefficient[15], and the GAP-statistic[11]. Our intention in these analyses
117 was not to dispute the original reported K, but instead to test whether methods that do not consider
118 reference distributions along the range of K would lead to visible biases. In these analyses, it was
119 demonstrated that the GAP-statistic continuously increased, implying improving stability regardless
120 of the structure (Supplementary Fig. 2). These findings imply the GAP-statistic is not well suited to
121 analysing complex genome wide expression datasets. Across these datasets, we also demonstrate
122 why M3C fits a beta distribution to the data to estimate extreme tail values, as for K = 2, the beta
123 distribution fits the reference slightly better than a normal distribution (Supplementary Fig. 3 and 4).
124 This step is important as it removes the limitations on p-value derivation imposed by a finite number
125 of simulations (Supplementary Fig. 5).

126

127 The PAC score displayed the same bias towards higher K values observed earlier on simulated null
128 datasets, decreasing steadily regardless of the structure, implying increased stability (Figure 3a-e).
129 This effect is more of a problem in datasets where the clustering is not very clear. For the GBM
130 dataset[3], while a PAC elbow can be seen at K = 4, the global optimal value is K = 10 (Fig. 3a). The
131 problem with the PAC score resembles the problem encountered by Tibshirani, et al. (2001), when
132 the authors developed the GAP-statistic to overcome the subjective decision regarding the location
133 of the elbow. For the GBM case, the Monte Carlo p-values and the RCSI demonstrate a clear optimal
134 value of K = 4 (p = 0.00059), with additional evidence for structure at K = 5 (p = 0.0071).

135

136 For the ovarian dataset[4], a global optimal PAC value is observed at K = 2, which is supported by the
137 RCSI (Fig. 3b). However, when the Monte Carlo p-values are calculated, it is in fact K = 5 which is the
138 optimal K (p = 0.0078). This happens because some datasets have a skewed null distribution at K = 2,

resulting in lower PAC scores (Supplementary Fig. 3b). These are inherently favoured by the algorithm, a bias that is unaddressed by the PAC score or the RCSI. Only by calculating p-values for each value of K can we mitigate against these types of systematic biases.

In cases where the clustering is very clear, the PAC score does perform well. In the lung cancer dataset[5], a global PAC optimal K can be seen at K = 2, which is supported by both the RCSI and the Monte Carlo p-value ($p = 0.0018$) (Fig. 3c). Although this conflicts with the original decision of K = 4, the M3C p-value for K = 4 was also significant ($p = 0.0032$), implying this would be another reasonable choice. However, the bias towards high K values of consensus clustering can be observed again on the diffuse glioma dataset[1] (Fig. 3d). Here the PAC score continuously decreases until it reaches a global optimum at K = 10. However, considering the reference distributions, M3C informs us that K = 8 is the most significant option ($p = 3.5\times10^{-9}$), which is also supported by the RCSI score. For the paraganglioma dataset[2], the RCSI estimates K = 6 and the Monte Carlo p-value supports this conclusion ($p = 1.6\times10^{-6}$), while the PAC score continually decreases, giving no clear choice of K (Fig. 3e). This is another example of why the reference distribution matters, as the RCSI method shows a local maximum for K = 2, while the Monte Carlo p-value does not support this. This is due to the uneven shape of the PG reference distribution for K = 2, which has positive kurtosis (Supplementary Fig. 4b). These findings imply results relying just on relative scores or mean comparisons with the reference can be potentially misleading.

In agreement with our findings on simulated null data, it was observed that the NMF cophenetic coefficient has a tendency towards calling K = 2 on real data (Fig. 3a-e). Only in the diffuse glioma dataset[1] did the maximum cophenetic coefficient suggest any other value of K. Although there are numerous variant decision rules for NMF in use[4,5,16], these do not compare against a null distribution. Instead of taking the most stable consensus matrix (highest cophenetic coefficient) as

the optimal K, local maxima are often selected[4,5]. Notably, for the ovarian dataset[4] a local maximum in the NMF cophenetic coefficient was observed at K = 5, which was supported by the M3C decision in this instance. Additional support was observed for the lung cancer optimal K, as an NMF global maximum cophenetic coefficient was detected for K = 2, and the M3C p-value also declared this K to be optimal (p = 0.0018). However, since a tendency in NMF towards K = 2 on null datasets has been observed in this study, it is unclear how confident we should be in this decision.

As a final step, we performed t-Distributed Stochastic Neighbor Embedding (t-SNE) on each dataset then calculated the silhouette width using either the original K or the M3C K to evaluate the relative strength of the M3C cluster assignments. t-SNE was performed first to reduce dimensionality, because the silhouette width has been shown to work poorly alone on high dimensional data in finding the true K[14]. This analysis demonstrated of the four datasets with differing K decisions to the original, the M3C decisions were better in three. These findings support the value of M3C's reference-based approach to deciding K.

Table II. Silhouette width of M3C optimal K assignments compared with original K decision assignments. Higher values of silhouette width (sil width) correspond to preferable clustering.

Dataset	Original K	Sil width	M3C K	Sil width
Glioblastoma[3]	4	0.28	4	0.28
Ovarian carcinoma[4]	4	0.30	5	0.27
Lung cancer[5]	4	0.26	2	0.27
Diffuse glioma[1]	4	0.041	8	0.18
Pheochromocytoma[2]	4	0.20	6	0.23

183 **M3C demonstrates good performance in finding K on simulated data**

184 Next, we sought to evaluate the performance of M3C on simulated data from K = 2 to K = 6 and
185 compare its performance to existing algorithms. In these tests, we varied the clusterlab alpha
186 parameter, which controls the distance between the clusters, and used algorithms which were able
187 to detect the true K from further apart cluster conditions (alpha = 2) to closer ones (alpha = 1) (Fig.
188 4a,b). Typically, in genome wide analyses many clusters will be overlapping and hard to distinguish
189 from one another. Therefore, sensitivity under these conditions is very valuable. This analysis found
190 that M3C using the RCSI score performed better than consensus clustering with the PAC score, M3C
191 using p-values, the GAP-statistic, CLEST, the original consensus clustering with the delta K score,
192 NMF, and progeny clustering. Notably, while M3C with the RCSI score was approximately 10% higher
193 in accuracy than M3C with p-values, the GAP-statistic, and consensus clustering with PAC, these
194 three methods performed similarly, within 4% of one another. CLEST was also a good performer in
195 this analysis. Overall, these simulations reinforce our findings on real data that M3C performs better
196 than other state-of-the-art methods.

197

198 **M3C can deal with complex structures using spectral clustering**

199 The performance of M3C is dependent on underlying clustering algorithm. Although k-means and
200 PAM perform well on the types of data generally encountered in genome-wide studies, they assume
201 the clusters are approximately spherical and equal in variance, which may not be true. Spectral
202 clustering is a widely applied technique due to its ability to cope with a broad range of structures[17].
203 Therefore, to increase the capabilities of the M3C software package, it includes self-tuning spectral
204 clustering[18]. We tested spectral clustering as M3C's inner algorithm versus PAM and k-means on two
205 synthetic datasets, one where the clusters were anisotropic (Fig. 5a), and a second where one
206 cluster had a far smaller variance than its neighbouring cluster (Fig. 5b). Under these conditions, it
207 was observed that M3C using PAM and k-means both had problems identifying the true K and

classifying the members of each cluster correctly. On the other hand, M3C using spectral clustering did not suffer these drawbacks. Using spectral clustering, M3C is also capable of recognising more complex non-Gaussian shapes, such as half-moons and concentric circles (Supplementary Fig. 6). The addition of spectral clustering to the M3C software package allows greater flexibility in the range of structures that may be examined.

M3C can quantify structural relationships between consensus clusters

An important question when the optimal K has been decided is, how do the discovered clusters relate to one another? Inherently, consensus clustering does not distinguish between flat versus hierarchical structure. To solve this, M3C performs hierarchical clustering on the medoids of each consensus cluster. To make the analysis statistically principled, M3C iteratively performs the SigClust method[19] on each pair of consensus clusters, then displays the pairwise p-values for each split of the dendrogram. Testing M3C on the PG dataset revealed a hierarchical relationship between the six clusters (Fig. 6a), with, for example, consensus clusters one and two grouping together ($p = 1.2 \times 10^{-80}$). In contrast, testing M3C on a null dataset without clusters demonstrated insignificant SigClust p-values and a flat dendrogram (Fig. 6b). The addition of a hierarchical clustering stage after choosing the optimal K should prove helpful in identifying structural relationships.

Sensitivity and complexity analysis of M3C

As a final step, we decided to evaluate M3C's internal parameters using the PAM algorithm, compare its runtimes with other methods, and calculate its complexity. A sensitivity analysis of the number of inner replications and outer simulations found M3C generally yielded stable results across six TCGA datasets with 100 inner replications and 100 outer simulations (Supplementary Fig. 7-8). We executed M3C on five datasets on a high-powered desktop computer using a single thread of an

Intel i7-5960X CPU @ 3.00GHz with 32GB of RAM. Runtimes ranged between 2-25 minutes, depending on dimensionality (Fig. 7a). We compared the runtime of M3C with other well performing methods from our earlier analysis on the same computer with a single thread (Fig. 7b-c). M3C, CLEST, and the GAP-statistic which all use Monte Carlo simulations as a reference were set to 25 reference iterations for comparative purposes. This analysis demonstrated that consensus clustering with the PAC score was the fastest method, followed by the GAP-statistic. CLEST and M3C were slower and similar in runtime for lower N (number of samples), but for N greater than 500, M3C performed more slowly than CLEST (Fig. 7b).

The complexity of the M3C algorithm is $O(BHA/C)$, where B is the number of Monte Carlo simulations, H is the number of consensus clustering resamples, and A is the complexity of the underlying clustering algorithm (see pseudo-code for M3C in Supplementary Note 1). The C denotes number of available processors, as M3C can be parallelized due to its independent simulations and subsampling subroutines. We empirically evaluated M3C's time complexity as a function of sample size N using the PAM algorithm, which has a complexity of $O(N^2)$. Calculating the slope of the log-log plot yielded an empirical complexity of $O(N^{2.4})$. This demonstrates that M3C is approximately quadratic in N.

Discussion

We report the advancement of the Monti consensus clustering algorithm to include a Monte Carlo simulation driven reference system for estimating the optimal K and testing the null hypothesis K=1, we call the method M3C. Our investigation into this consensus clustering algorithm demonstrated it has inherent bias towards higher values of K. These occur due to not considering the reference

256 distribution along the range of K when deciding on its value. Although considering these
257 distributions is a relatively straightforward procedure, as we have demonstrated, it has important
258 implications. To date, testing of the null hypothesis by TCGA has been conducted by SigClust after
259 deciding on the value of K using the standard methods[2,6,16]. SigClust tests the null hypothesis K=1 for
260 pairs of clusters, but it does not directly estimate K. The advantage of M3C is that it can both find K
261 and test the null hypothesis K=1.

262

263 Our reanalysis of high-profile stratified medicine studies, predominantly from TCGA[1-5,9,16], questions
264 the value of consensus clustering when used without considering the appropriate reference
265 distributions. The bias towards higher values of K, coupled with subjective decision making as to
266 what constitutes the optimal K, similar to the original elbow problem solved by the GAP-statistic[11],
267 may provide misleading results. We identified two cases in the literature where structure had been
268 declared despite M3C indicating no significant evidence against the null hypothesis. In the case of
269 the SLE study, seven subtypes were originally declared in a major transcriptomic analysis[9]. Within
270 the context of these new findings, it is perhaps better to describe these subtypes as existing within a
271 noisy spectrum of non-distinct states. This hints that there may be publication bias for positive
272 declaration of structures.

273

274 It is necessary to remark on the limitations of the approach. The M3C method can allow testing of
275 the null hypothesis K=1 and mitigate bias. However, this method does not allow, for example, the
276 formal statistical comparison of selecting K=2 compared with other values of K. The relative
277 magnitude of the p values can be used to estimate the optimal K by comparing against the null K=1
278 scenario like using the RCSI, however, this is not formal hypothesis testing. A second limitation is
279 that M3C is computationally expensive, however, extreme tail estimation and multi-core ability

280 mitigate this problem. Finally, just because the p-value or RCSI supports a given K gives no guarantee
281 the identified clusters or their number will be reproducible in an independent validation dataset.

282

283 Other types of consensus clustering methods include Infinite Ensemble Clustering[20] (IEC) and
284 Entropy-based consensus clustering[21] (ECC), which can be used for patient stratification. IEC
285 incorporates marginalized denoising auto-encoder with dropout noises to generate the expectation
286 representation for infinite basic partitions. ECC employs an entropy-based utility function to fuse
287 many basic partitions into a single consensus structure. A future challenge is to systematically
288 evaluate the performance of a wider range of consensus clustering methods on genome wide
289 expression data.

290

291 We benchmarked the performance of M3C against a number of alternatives, including: Monti
292 consensus clustering, the GAP-statistic, progeny clustering, and CLEST. Several cluster validity indices
293 were not tested, however, such as: the Silhouette index[22], the Calinski Harabasz index[23], the Jaccard
294 index[24], and the Davies-Bouldin index[25]. It would be interesting to determine if any of these indices
295 perform well in determining the optimal K when applied on consensus matrices produced by the
296 consensus clustering algorithm, our study indicates they will be subject to bias without a reference
297 procedure. It is also relevant to mention that there are other methods that could be applied to
298 investigate the significance of dendrogram splits, such as the inheritance procedure[26].

299

300 Lastly, it is important to mention the methodological contributions of clusterlab. Clusterlab is a
301 flexible new method for generating Gaussian clusters. Unlike prior methods[14,27,28], it is able to
302 generate and position Gaussain clusters in a highly customisable manner with specified variance,
303 spacing, and size. Clusterlab can generate data similar in nature to cancer gene expression datasets,

which are typically high-dimensional and Gaussian[19]. The method should appeal to researchers in a range of disciplines for testing methods for finding K and clustering algorithms.

Methods

M3C. The method uses a Monte Carlo simulation, which generates random data with each iteration, to repeat the Monti et al. consensus clustering algorithm many times over. Then, the real algorithm is run just once to compare the real cluster stabilities along the range of K with those expected using random Gaussian data (K = 1). Pseudo-code is given in Supplementary Note 1. This gives a new method for choosing K after consensus clustering that removes bias towards high values of K and allows one to statistically test for the presence of structure. The specific details are now given.

Simulation of the reference dataset. There are a range of options for the generation of reference datasets in M3C's Monte Carlo simulation. We use an approach first proposed by Tibshirani et al., which preserves covariance structure via principal component analysis (PCA). With an input matrix, $T \in \mathbb{R}^{S*F}$ we can compute the input data's eigenvector matrix $A \in \mathbb{R}^{F*S}$ and its principal component score matrix, $Y \in \mathbb{R}^{S*S}$, where F is the number of features in the provided matrix, and S is the number of samples. The steps taken to generate random data are repeated $b = 1 \ldots B$ times:

1. Conduct PCA to obtain the orthogonal matrix of eigenvectors, A of the input data T:

$$Y_{S*S} = T_{S*F} * A_{F*S} \quad (1)$$

2. Next, a random PC score matrix is generated, $Y^b \in \mathbb{R}^{S*S}$, where the ith column is filled with random values from a normal distribution with mean zero and standard deviation equal to the ith column in Y. Let, D_i be the standard deviation of Y_{*i} and for $i = 1 \ldots S$:

$$Y_{*i}^{b} \sim N(0, D_i) \quad (2)$$

3. Multiplying Y^b with the transpose of A yields $Q^b \in \mathbb{R}^{S*F}$, a single simulated null dataset with the same feature correlation structure as T, but without clusters.

$$Q_{S*F}^{b} = Y_{S*S}^{b} * A_{S*F}^{b} \quad (3)$$

Steps 1-3 are repeated by M3C for each Monte Carlo reference simulation for $b = 1 \ldots B$, and for the bth simulation one random dataset, Q^b is passed into the consensus clustering algorithm (described below) to calculate null reference stability scores for $K = 2, \ldots, maxK$. After B simulations, the consensus clustering algorithm is run just once on the input data for comparison using procedures we will go on to detail. M3C is set to use $B = 100$ and this was the parameter setting used for the simulations in this study.

Consensus clustering. The Monti et al. consensus clustering algorithm subsamples the input data sample-wise, H times, and with each resampling iteration clusters the perturbed dataset using a user defined inner clustering algorithm (e.g., PAM) for each value of K. It then measures the stability of the sample cluster assignments over all resampling iterations to decide K. M3C includes PAM, k-means, and spectral clustering as options, with PAM set by default due to its superior speed. Let, $D^{(1)}, D^{(2)}, \ldots, D^{(H)}$ be the list of H perturbed datasets, and let $M^{(h)} \in \{0,1\}^{N*N}$ be the connectivity matrix resulting from clustering dataset $D^{(h)}$, the entries of $M^{(h)}$ are then defined as:

$$M^{(h)}(i,j) = \begin{cases} 1 & \text{if samples i and j are in the same cluster} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

To keep count of the number of times samples i and j are resampled together in the perturbed dataset $D^{(h)}$ an indicator matrix $I^{(h)} \in \{0,1\}^{N*N}$ is defined:

$$I^{(h)}(i,j) = \begin{cases} 1 & \text{if samples i and j are in dataset } D^{(h)} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The consensus matrix, $M \in [0,1]^{N*N}$, is defined as the normalised sum of all the connectivity matrices of all H perturbed datasets:

$$M(i,j) = \frac{\sum_{h=1}^{H} M^{(h)}(i,j)}{\sum_{h=1}^{H} I^{(h)}(i,j)} \quad (6)$$

The entry (i,j), or consensus index, is the number of times that two samples cluster together divided by the total number of times they were sampled together across all the perturbed datasets. A value of 1 would correspond to a perfect score as the two samples are always found in the same cluster across all resampling runs, while a value of 0 would correspond to the worst score as the two samples never are found in the same cluster. A consensus matrix is generated for every value of K and then the stability of each matrix quantified using an empirical cumulative distribution (CDF) plot. For any given consensus matrix M, the CDF is calculated and is defined over the range $[0,1]$ as follows:

$$CDF(c) = \frac{\sum_{i<j} 1\{M(i,j) \leq c\}}{N(N-1)/2} \quad (7)$$

Where $1\{\ldots\}$ denotes the indicator function, $M(i,j)$, denotes entry (i,j) of the consensus matrix M, N is the number of rows (and columns) of M, and c is the consensus index value.

Calculation of the PAC score. The CDF plot has consensus index values on the x axis and CDF values on the y axis. A perfectly stable cluster solution will have a flat CDF plot representing a matrix purely of 0s and 1s, therefore the degree of CDF flatness for each K is a measure of the stability of K. To quantify this, M3C uses the PAC score, a metric shown to perform well in simulations[14]. PAC is defined as the fraction of sample pairs with consensus index values falling in the intermediate sub-interval $(x_1, x_2) \in [0,1]$. For a given value of K, $CDF(c)$ corresponds to the fraction of sample pairs with consensus index values less than or equal to c and PAC is defined as:

$$PAC_K(x_1, x_2) = CDF_K(x_2) - CDF_K(x_1) \quad (8)$$

M3C calculates the PAC score with $x_1 = 0.1$ and $x_2 = 0.9$. Although the PAC window is a user defined parameter, we have found these settings to perform well in our experience.

Calculation of the RCSI. To account for the reference PAC scores from $b = 1 \ldots B$, where B is the total number of Monte Carlo simulations, M3C uses the RCSI. Let, $Pref_{Kb}$ be the reference PAC score from the bth Monte Carlo simulation for a given K, and, $Preal_K$ the real PAC score for that K, then the $RCSI_K$ is defined as:

$$RCSI_K = \log_{10}\left(\frac{1}{B}\sum_{b=1}^{B} Pref_{Kb}\right) - \log_{10}(Preal_K) \quad (9)$$

Calculation of the Monte Carlo p value. To improve the selection of the optimal K, M3C derives Monte Carlo p values by testing the real PAC score for each K against the null PAC distribution, generated using simulated structureless data. Let o_K be the number of observed PAC scores in the reference less than or equal to the real PAC score, let B be the total number of Monte Carlo simulations, and the p value for that value of K, P_K is then defined as:

$$P_K = \frac{o_K + 1}{B + 1} \quad (10)$$

Where 1 is added the numerator and denominator to avoid p values of zero[29].

Interpretation of the p-values. For each K the method will test the null hypothesis H_0 that the PAC score, $Preal_K$, came from a single Gaussian cluster ($K = 1$) versus the alternative hypothesis H_A that $Preal_K$ did not come from a single Gaussian cluster ($K \neq 1$). If a p value for a K reaches significance (alpha=0.05) it should be viewed as evidence that the data is not a single Gaussian cluster. If no p values along the range of K reaches significance (alpha=0.05) it should be viewed as evidence that the data is a single Gaussian cluster. The relative significance of the p-values can be used to suggest the most preferable K, although we caution that the method does not formally test the selection of one value of K versus another.

395 *Calculation of the beta distribution p-value.* To estimate p-values beyond the range of the Monte
396 Carlo simulation, M3C fits a beta distribution. This distribution is more flexible than the normal
397 alternative, which is especially helpful when K = 2, which tends to result in null distributions with
398 nonzero skew and kurtosis. Moreover, the PAC score is bound on the interval $[0,1]$, as is the beta
399 distribution, providing the correct range for computation. The α and β shape parameters required
400 for the beta distribution are derived using maximum likelihood estimates for the mean, μ, and
401 variance, σ^2, of the reference PAC scores for any given K:

402 $$\mu = \frac{1}{N}\sum_{n=1}^{N} Pref_{Kn} \quad (11)$$

403 $$\sigma^2 = \frac{1}{N}\left(\sum_{n=1}^{N} Pref_{Kn} - \mu\right)^2 \quad (12)$$

404 $$\alpha = \left(\frac{1-\mu}{\sigma^2} - \frac{1}{\mu}\right)\mu^2 \quad (13)$$

405 $$\beta = \alpha\left(\frac{1}{\mu} - 1\right) \quad (14)$$

406 These α and β shape parameters are then used by M3C to generate the reference distribution for K.
407 The real PAC score is used as a test statistic for derivation of the estimated p value. Let x denote the
408 reference PAC score. Then the beta probability density function (PDF) is defined as:

409 $$PDF(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\beta(\alpha,\beta)} \quad (15)$$

410 **Simulating NXN dimensional Gaussian clusters in a precise manner.** We found that current
411 Gaussian cluster simulation methods were inadequate for systematic testing of M3C. MixSim[27],
412 generates Gaussian clusters, however, it is not possible to precisely control their positioning. The
413 Python scikit-learn machine learning module contains a Gaussian cluster simulator, but it generates
414 clusters randomly and controlled positioning is not possible. Another method allows controlled
415 spacing[14], but does not generate Gaussian clusters, instead the clusters resemble triangular slices
416 and the variance and size cannot be set. Therefore, we developed clusterlab (https://cran.r-
417 project.org/web/packages/clusterlab/index.html). Clusterlab is a novel method that allows

simulation of Gaussian clusters with controlled spacing, size, and variance. It works by generating cluster centres or points on the circumference of a circle in 2D space because this is easier to work in mathematically than higher dimensional space. The specific details are now given.

Generating evenly spaced points on the perimeter of a circle. To control the spacing, size, and variance of synthetic clusters, clusterlab works within a 2D Cartesian coordinate system with an origin at $(0,0)$. First, the algorithm generates a set $S = \{w_i \in \mathbb{R}^2, i = 1, \dots X\}$ of X evenly spaced pairs of coordinates, where $w_i = (x_i, y_i)$, on the perimeter of a circle. Each of these coordinates later will be the centre of a Gaussian cluster, therefore, X is also the number of clusters to be generated. Let, r be the radius of the circle, then, for the ith cluster centre from $i = 1 \dots X$ we need to set $i = 0$ for the first cluster centre, so for $i = 0 \dots X - 1$, the coordinate pairs are calculated as follows:

$$x_i = \cos\frac{2\pi}{X \cdot i} r \quad (21)$$

$$y_i = \sin\frac{2\pi}{X \cdot i} r \quad (22)$$

This naturally leaves the r parameter as a means of controlling the spacing of the cluster centres. However, at this point, we also introduce an additional parameter for moving the ith cluster centre, α_i. α_i is a scalar that can be used to push each coordinate pair (or vector) away from its starting point, yielding the transformed coordinates (x_i', y_i'). In the case of a cluster being left stationary, $\alpha_i = 1$. More specifically, for all pairs in set S, from $i = 1 \dots X$:

$$(x_i', y_i') = \alpha_i (x_i, y_i) \quad (23)$$

We also leave the option to add a final coordinate to S at (0,0), to allow a central cluster within the middle of the ring to be generated later.

Generation of more complex multi-ringed structures. As an optional next step to extend the single ring system, clusterlab can create multiple rings or concentric circles of 2D coordinates. After simulating the qth ring, as described above, from $q = 1 \dots Q$, the qth rings 2D coordinates are

pushed away from the origin using vector multiplication with a scalar, let this scalar be β_q, let the newly transformed coordinates be (x_i'', y_i''), and so for $i = 1 \dots X$:

$$(x_i'', y_i'') = \beta_q (x_i', y_i') \quad (24)$$

Our new total number of samples, T, will be, $T = X * Q$. With each iteration from $q = 1 \dots Q$, the ith transformed coordinates, $d_i = (x_i'', y_i'')$, are added to a new set, $R = \{d_i \in \mathbb{R}^2, i = 1, \dots T\}$. Optionally, another layer of complexity may be added by using vector rotations of the qth rings coordinate pairs from $i = 1 \dots X$, by setting $\theta_q \neq 0$ in the following equation. To calculate each of the rings new coordinates (x_i''', y_i''') from $i = 1 \dots X$, the following calculation is performed for every pair:

$$x_i''' = x_i'' \cos(\theta_q) - y_i'' \sin(\theta_q) \quad (25)$$

$$y_i''' = x_i'' \sin(\theta_q) + y_i'' \cos(\theta_q) \quad (26)$$

Generation of Gaussian clusters. At this point we will assume that multiple rings have not been generated and we are working with, S, a set of (x_i', y_i') coordinates described by equation 23. However, the method that generates the Gaussian cluster multi-ringed system is identical to the single ringed system described below, except we start with the multiplied (x_i'', y_i'') or multiplied and rotated set of (x_i''', y_i''') points from the multi ring 2D coordinate set, R.

To form X Gaussian clusters of size M_i per cluster, we add Gaussian noise from a normal distribution, $N(0, D_i)$, to the ith pair of cluster centre 2D coordinates, $k_i = (x_i', y_i')$, to create the new coordinates, $t_i = (x_j, y_j)$. Performing this M_i times for each cluster centre, giving a total of $Z = \sum_{i=1}^{Z} M_i$ coordinate pairs, yields the final set, $J = \{t_i \in \mathbb{R}^2, i = 1, \dots Z\}$. The number of samples in each cluster may be set by varying M_i, and the clusters variance, by setting D_i. The new coordinate pairs, (x_j, y_j), to be added to, J, for all samples are calculated as follows:

$$(x_j, y_j) = (x_i' + N(0, D_i), y_i' + N(0, D_i)) \quad (27)$$

465 *Projection of the final 2D coordinates into N dimensions.* We transform the cluster sample
466 coordinates into N dimensions with a previously explained method which uses a reverse PCA[14]. First,
467 two random vectors are generated of length V, where V will equal the number of features in the
468 final matrix, from a normal distribution $N(0,0.1)$, let these be v_1 and v_2. The SD of 0.1 was chosen
469 empirically after examination of the scale of the simulated PC plots compared to those from real
470 expression datasets. The v_1 and v_2 vectors are treated as fixed eigenvectors in this method, and
471 each of our previously simulated coordinate pairs are treated as 2D PC scores. The final matrix,
472 $F \in \mathbb{R}^{Z*V}$, comprised of Z rows (samples) and V columns (features), is formed by linear
473 combinations of the fixed eigenvectors with the pairs of PC scores. Let, x_i and y_i be the PC scores of
474 the ith sample, from $i = 1 \ldots Z$ from set J, then the ith row of the output matrix F is given by:

475 $$F_{i*} = x_i * v_1 + y_i * v_2 \quad (28)$$

476 *Non-Gaussian structures.* For generating structures used in the spectral clustering analysis, the CRAN
477 clusterSim package version 0.47 was used[30]. For the anisotropic and unequal variance clusters, 90
478 samples were simulated with two dimensions with the cluster.Gen function using the default
479 settings. For the half-moon clusters, the shapes.two.moon function was used with 90 samples, and
480 for the concentric circles the shapes.two.circles function was used with 180 samples, both using
481 default settings. The sample number was increased in the latter to prevent gaps forming in the
482 concentric circles.

483

484 **Real test datasets.** All test datasets, apart from the SLE dataset, were already normalised and
485 downloaded directly through TCGA publication page (https://tcga-
486 data.nci.nih.gov/docs/publications/) during the period of April to June 2017, further details are
487 provided in Supplementary Table 1. We chose RNA-seq or microarray data from the TCGA where the
488 data was already normalised. The diffuse glioma (DG) dataset is a RNA-seq matrix consisting of 2266
489 features and 667 samples[1] (https://tcga-data.nci.nih.gov/docs/publications/lgggbm_2015/LGG-

490 GBM.gene_expression.normalized.txt). The GBM dataset, is a microarray matrix consisting of 1740
491 features and 206 samples[3] (https://tcga-
492 data.nci.nih.gov/docs/publications/gbm_exp/unifiedScaledFiltered.txt), the feature list used was
493 taken from a later publication on the same dataset[6]. The lung cancer (LC) dataset[5] used was a RNA-
494 seq matrix consisting of 178 samples and 2257 features (https://tcga-
495 data.nci.nih.gov/docs/publications/lusc_2012/gaf.gene.rpkm.20111213.csv.zip), the feature list used
496 to filter this dataset was from an earlier publication where four subtypes had been identified
497 (http://cancer.unc.edu/nhayes/publications/scc/wilkerson.scc.tgz). The paraganglioma (PG) dataset
498 downloaded was a RNA-seq matrix consisting of 173 samples and 3000 features (https://tcga-
499 data.nci.nih.gov/docs/publications/pcpg_2017/PCPG_mRNA_expression_naRM.log2.csv.zip), the
500 gene wise filtering scheme used was the same as described as in the corresponding publication[2]. The
501 ovarian cancer (OV) dataset[4] was a RNA-seq matrix of 489 samples and 800 features (https://tcga-
502 data.nci.nih.gov/docs/publications/ov_2011/TCGA_489_UE.zip), and the gene list used for
503 subsequent filtering was obtained from an earlier publication that detected four subtypes[31]. The SLE
504 dataset[9] used was a microarray matrix of 82 samples and 48 features, the data was obtained from
505 GEO (GSE65391), normalised, and filtered in the manner described in the associated publication.

506

507 **References**

508 1 Ceccarelli, M. *et al.* Molecular profiling reveals biologically discrete subsets and pathways of
509 progression in diffuse glioma. *Cell* **164**, 550-563 (2016).
510 2 Fishbein, L. *et al.* Comprehensive molecular characterization of pheochromocytoma and
511 paraganglioma. *Cancer cell* **31**, 181-193 (2017).
512 3 Network, C. G. A. R. Comprehensive genomic characterization defines human glioblastoma
513 genes and core pathways. *Nature* **455**, 1061 (2008).

514 4 Network, C. G. A. R. Integrated genomic analyses of ovarian carcinoma. *Nature* **474**, 609
515 (2011).
516 5 Network, C. G. A. R. Comprehensive genomic characterization of squamous cell lung cancers.
517 *Nature* **489**, 519 (2012).
518 6 Verhaak, R. G. *et al.* Integrated genomic analysis identifies clinically relevant subtypes of
519 glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer cell*
520 **17**, 98-110 (2010).
521 7 Lefaudeux, D. *et al.* U-BIOPRED clinical adult asthma clusters linked to a subset of sputum
522 omics. *Journal of Allergy and Clinical Immunology* **139**, 1797-1807 (2017).
523 8 Ottoboni, L. *et al.* An RNA profile identifies two subsets of multiple sclerosis patients
524 differing in disease activity. *Science translational medicine* **4**, 153ra131-153ra131 (2012).
525 9 Banchereau, R. *et al.* Personalized immunomonitoring uncovers molecular networks that
526 stratify lupus patients. *Cell* **165**, 551-565 (2016).
527 10 Monti, S., Tamayo, P., Mesirov, J. & Golub, T. Consensus clustering: a resampling-based
528 method for class discovery and visualization of gene expression microarray data. *Machine*
529 *learning* **52**, 91-118 (2003).
530 11 Tibshirani, R., Walther, G. & Hastie, T. Estimating the number of clusters in a data set via the
531 gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**,
532 411-423 (2001).
533 12 Dudoit, S. & Fridlyand, J. A prediction-based resampling method for estimating the number
534 of clusters in a dataset. *Genome biology* **3**, research0036. 0031 (2002).
535 13 Hu, C. W., Kornblau, S. M., Slater, J. H. & Qutub, A. A. Progeny clustering: a method to
536 identify biological phenotypes. *Scientific reports* **5** (2015).
537 14 Şenbabaoğlu, Y., Michailidis, G. & Li, J. Z. Critical limitations of consensus clustering in class
538 discovery. *Scientific reports* **4** (2014).

539 15 Gaujoux, R. & Seoighe, C. A flexible R package for nonnegative matrix factorization. *BMC*
540 *bioinformatics* **11**, 367 (2010).
541 16 Network, C. G. A. Comprehensive molecular portraits of human breast tumours. *Nature* **490**,
542 61 (2012).
543 17 Ng, A. Y., Jordan, M. I. & Weiss, Y. in *Advances in neural information processing systems.*
544 849-856.
545 18 Zelnik-Manor, L. & Perona, P. in *Advances in neural information processing systems.* 1601-
546 1608.
547 19 Liu, Y., Hayes, D. N., Nobel, A. & Marron, J. Statistical significance of clustering for high-
548 dimension, low–sample size data. *Journal of the American Statistical Association* **103**, 1281-
549 1293 (2008).
550 20 Liu, H., Shao, M., Li, S. & Fu, Y. Infinite ensemble clustering. *Data Mining and Knowledge*
551 *Discovery* **32**, 385-416 (2018).
552 21 Liu, H. *et al.* Entropy-based consensus clustering for patient stratification. *Bioinformatics* **33**,
553 2691-2698 (2017).
554 22 Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster
555 analysis. *Journal of computational and applied mathematics* **20**, 53-65 (1987).
556 23 Caliński, T. & Harabasz, J. A dendrite method for cluster analysis. *Communications in*
557 *Statistics-theory and Methods* **3**, 1-27 (1974).
558 24 Jaccard, P. J. B. S. V. S. N. Étude comparative de la distribution florale dans une portion des
559 Alpes et des Jura. **37**, 547-579 (1901).
560 25 Davies, D. L. & Bouldin, D. W. A cluster separation measure. *IEEE transactions on pattern*
561 *analysis and machine intelligence*, 224-227 (1979).
562 26 Goeman, J. J. & Finos, L. The inheritance procedure: multiple testing of tree-structured
563 hypotheses. *Statistical Applications in Genetics and Molecular Biology* **11**, 1-18 (2012).

564 27 Melnykov, V., Chen, W.-C. & Maitra, R. MixSim: An R package for simulating data to study
565 performance of clustering algorithms. *Journal of Statistical Software* **51**, 1 (2012).
566 28 Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of machine learning*
567 *research* **12**, 2825-2830 (2011).
568 29 Phipson, B. & Smyth, G. K. Permutation P-values should never be zero: calculating exact P-
569 values when permutations are randomly drawn. *Statistical applications in genetics and*
570 *molecular biology* **9** (2010).
571 30 Walesiak, M., Dudek, A. & Dudek, M. clusterSim: Searching for optimal clustering procedure
572 for a data set. *R package version 0.36-1* (2008).
573 31 Verhaak, R. G. *et al.* Prognostically relevant gene signatures of high-grade serous ovarian
574 carcinoma. *The Journal of clinical investigation* **123** (2012).
575 32 Wilkerson, M. D. *et al.* Lung squamous cell carcinoma mRNA expression subtypes are
576 reproducible, clinically-important and correspond to different normal cell types. *Clinical*
577 *cancer research*, clincanres. 0199.2010 (2010).

578

579 **Author contributions**

580 C.R.J conceived and designed the approach. C.R.J and D.W wrote the manuscript. C.R.J and D.W
581 wrote the code. C.R.J, D.W, K.G, and D.R performed data analyses. All authors reviewed and edited
582 the manuscript. M.B, C.P, M.E, and M.L supervised the project.

583

584 **Additional information**

585 The authors declare that they have no competing interests.

586

587 **Figure legends**

588

589 **Figure 1. Bias in the estimation of K using Monti and NMF consensus clustering.** (A) A PCA plot of a
590 simulated null dataset where only one cluster should be declared. (B) Monti consensus clustering
591 yields a CDF plot implying improved stability with increased K. (C) The PAC score to measure the
592 stability of K decreases with its value, demonstrating a strong preference towards estimating higher
593 optimal values of K. (C) NMF consensus clustering yields a cophenetic coefficient plot which implies
594 lower values of K are preferable using this method.

595

596 **Figure 2. Overview of the M3C method and an initial demonstration.** (A) A schematic of the M3C
597 method and software. After exploratory PCA to investigate structure, the M3C function may be run
598 which includes two functions; M3C-ref and M3C-real. The M3C-ref function runs consensus
599 clustering with simulated random data sets that maintain the same gene-gene correlation structure
600 of the input data. While, the M3C-real function runs the same algorithm for the input data.
601 Afterwards, the relative cluster stability index (RCSI), Monte Carlo p values, and beta p values are
602 calculated. Structural relationships are then analysed using hierarchical clustering of the consensus
603 cluster medoids with SigClust to calculate significance of the dendrogram branch points. (B) Results
604 from running M3C on a simulated null dataset, it can be clearly seen that the p values do not reach
605 significance along the range of K, therefore the correct result is suggested, K=1. (C) Results from
606 running M3C on a simulated dataset where four clusters are found, the correct decision is made by
607 M3C. (D) Using M3C, a systemic lupus erythematosus dataset was detected with no significant
608 evidence of structure. (E) Similarly, a breast cancer dataset was identified with no significant
609 evidence of structure.

610

611 **Figure 3. Further evidence of bias existing in widely applied consensus clustering algorithms.** (A)
612 Results from running M3C on a glioblastoma dataset[3] found the optimal K was four. Consensus
613 clustering using the PAC-score shows an optimal K of ten, and NMF of two. (B) Results from running
614 M3C on an ovarian cancer dataset[4] found the optimal K was five. Consensus clustering using the
615 PAC-score shows an optimal K of two, and NMF also of two. (C) Results from running M3C on a lung
616 cancer dataset[32] found the optimal K was two. Consensus clustering using the PAC-score shows an
617 optimal K of two, and NMF also of two. (D) Results from running M3C on a diffuse glioma dataset[1]
618 found the optimal K was eight. Consensus clustering using the PAC-score shows an optimal K of ten,
619 and NMF of four. (E) Results from running M3C on a paraganglioma dataset[2] found the optimal K
620 was six. Consensus clustering using the PAC-score shows an optimal K of ten, and NMF of two. It can
621 be observed, consensus clustering using the PAC-score and NMF both tend towards K=10 or K=2,
622 respectively, on real data.

623

624 **Figure 4. M3C demonstrates good performance in finding K on simulated data.** (A) A sensitivity
625 analysis was conducted for every algorithm for K=2 to K=6 while varying the alpha parameter of
626 clusterlab (degree of Gaussian cluster separation). Accuracy was calculated as the fraction of correct
627 optimal K decisions, and for each alpha, with 25 iterations performed at each step. CC(original)
628 refers to the Monti et al. (2003) consensus clustering method, GAP-STAT refer to the GAP-statistic,
629 CC(PAC) refers to consensus clustering with the PAC-score. (B) Performance was calculated across
630 the range of K tested for each algorithm as the mean accuracy.

631

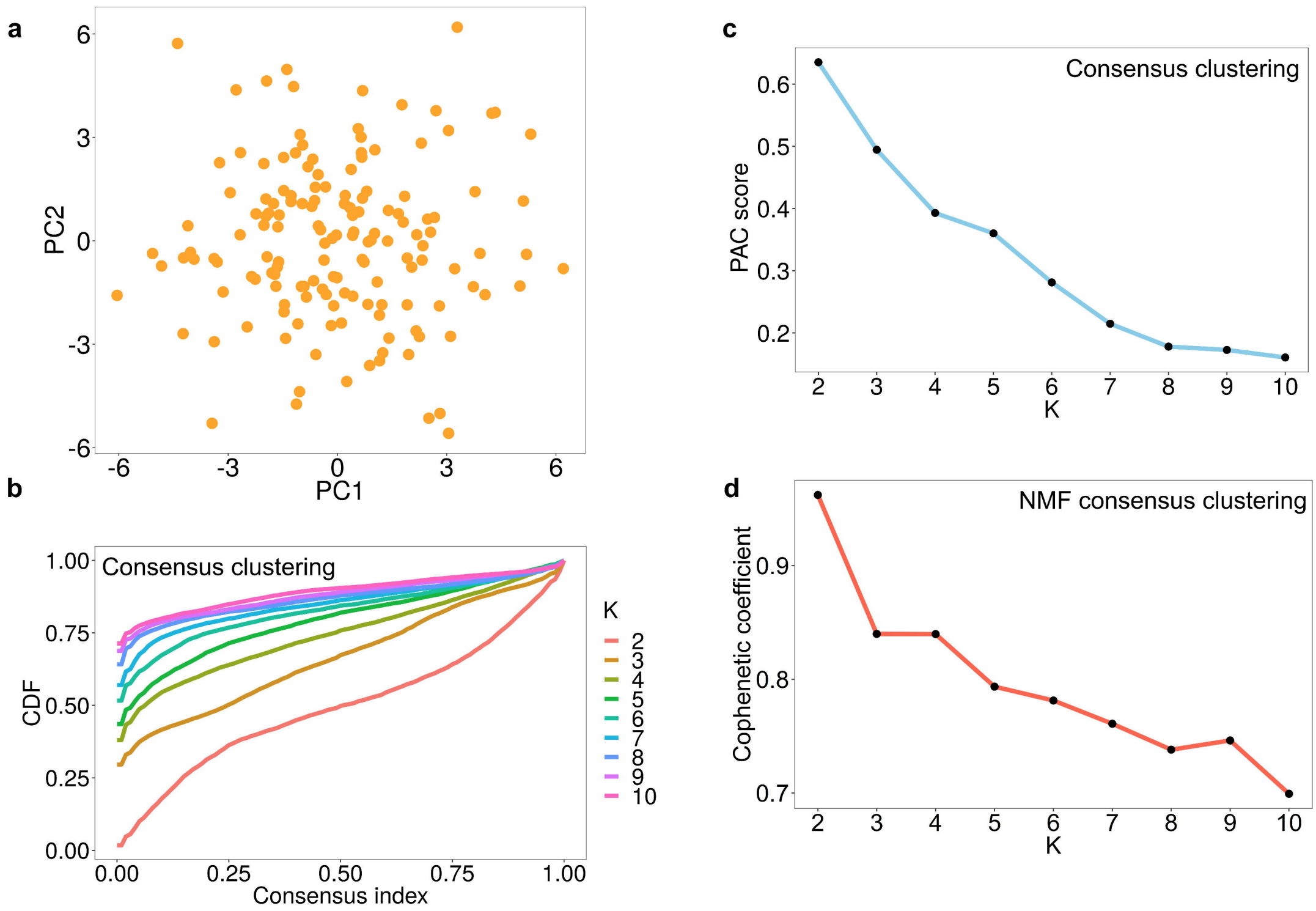
632 **Figure 5. M3C uses spectral clustering to deal with complex structures.** (A) Results from running
633 M3C using either spectral, PAM, or k-means clustering on anisotropic structures. The results for K=2
634 for each inner algorithm are shown in all cases, in the corner of the plots are the optimal K decisions

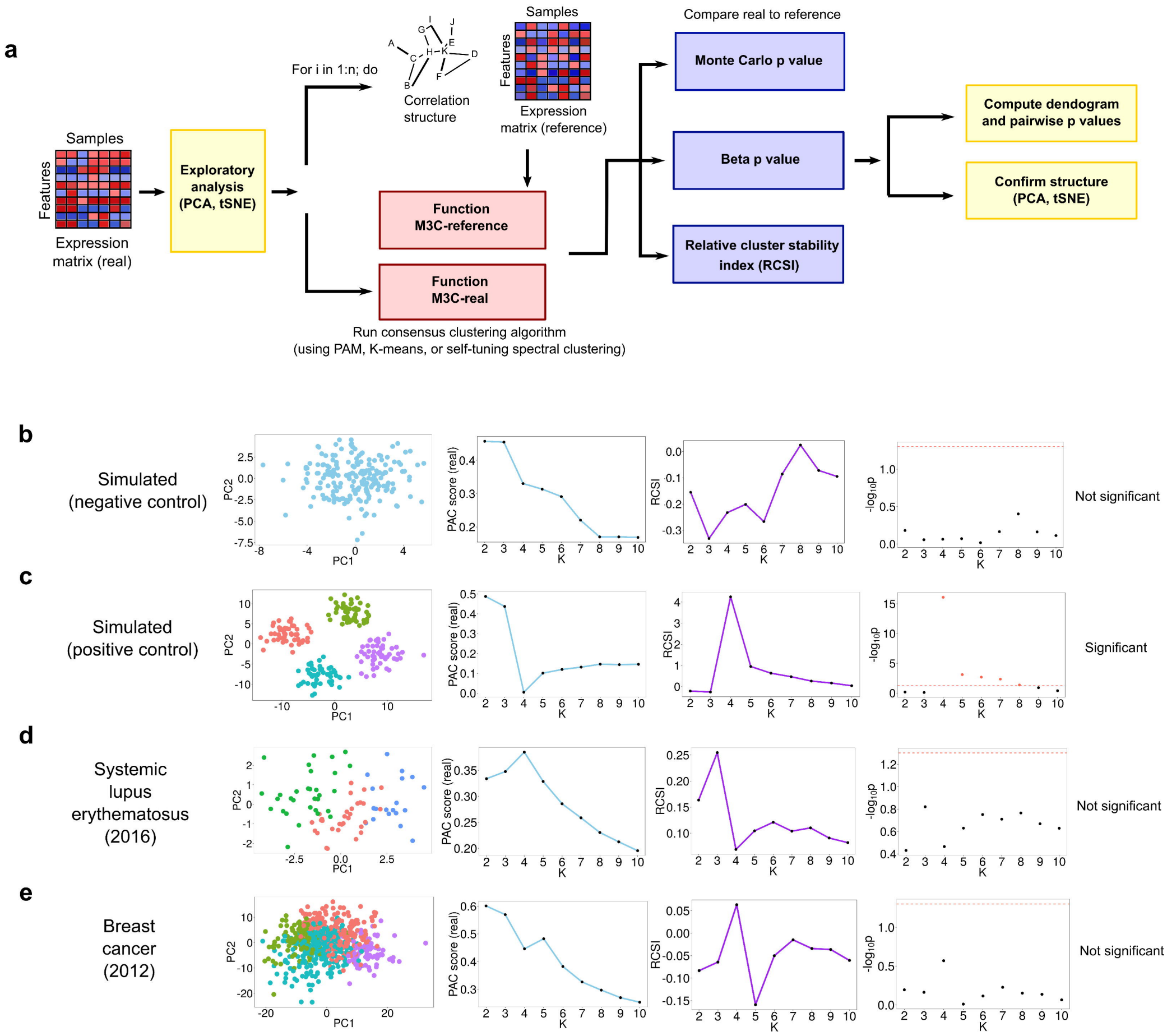
using the RCSI. (B) Similarly, results from testing different internal algorithms on structures of unequal variance.

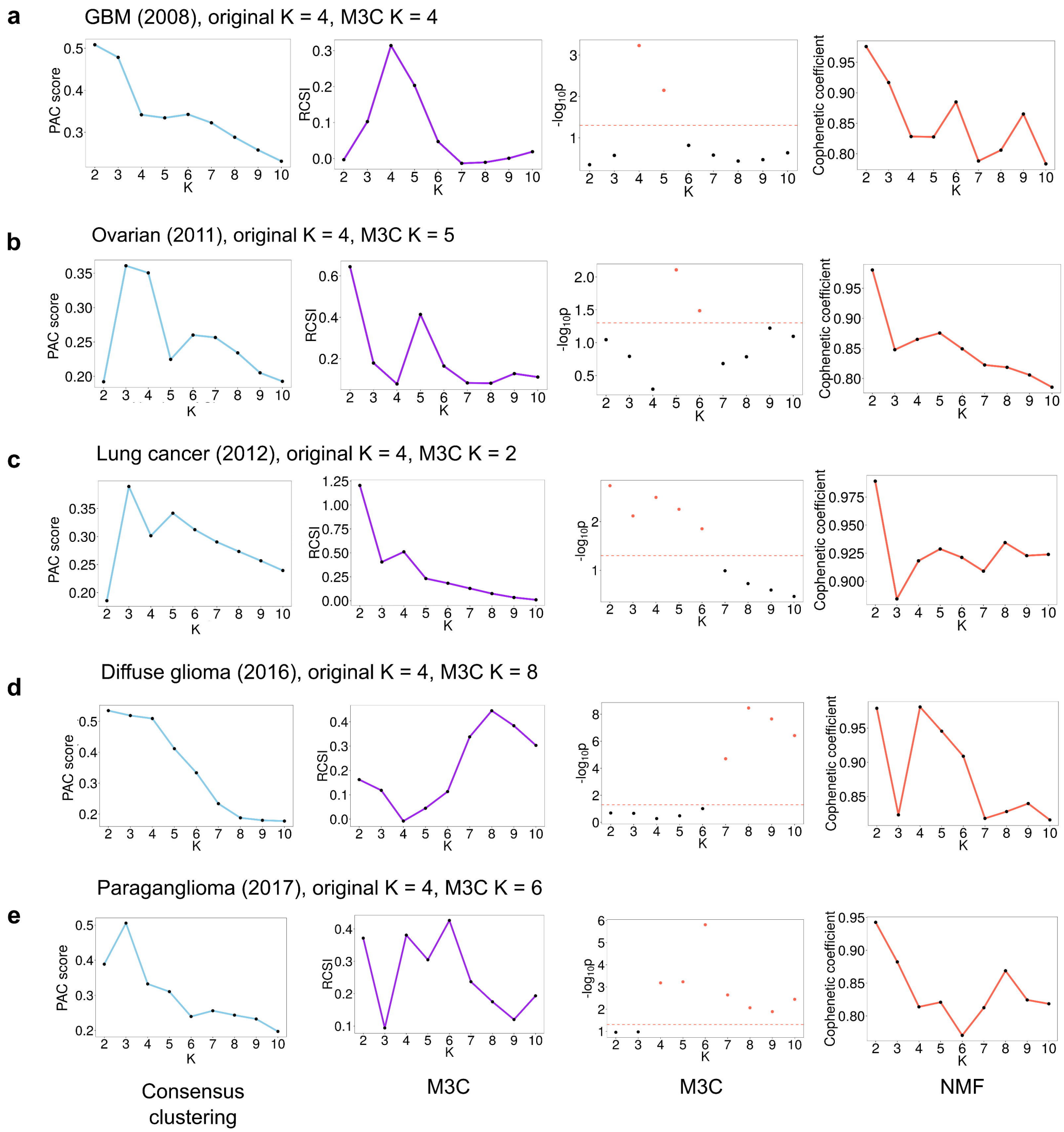
Figure 6. M3C can investigate structural relationships between consensus clusters. M3C calculates the medoids of each consensus cluster, then hierarchical clustering is performed on these, SigClust is run to detect the significance of each branch point. (A) Results from M3C structural analysis of the six clusters obtained from the paraganglioma dataset analysis[2], all p values were strongly significant, supporting the M3C decision of the declaration of structure. (B) Results from the same analysis run on a simulated null dataset of the same dimensions, no p values were significant.

Figure 7. M3C can perform quickly across a range of datasets. (A) M3C runtimes (in minutes) for five datasets used in the analysis. Performance was measured on an Intel Core i7-5960X CPU running at 3.00GHz using a single thread with 32GB of RAM. M3C was run using 25 outer Monte Carlo simulations and 100 inner iterations using the PAM algorithm. (B) M3C and other method runtimes in minutes for a series of simulated datasets with the number of samples (N) ranging from 100-1000 for datasets of 1000 features. CLEST and the GAP-statistic, which also use a Monte Carlo reference procedure, were set to run with 25 Monte Carlo simulations, the same as M3C for comparison. (C) Log-log plot of the same data shown in B.

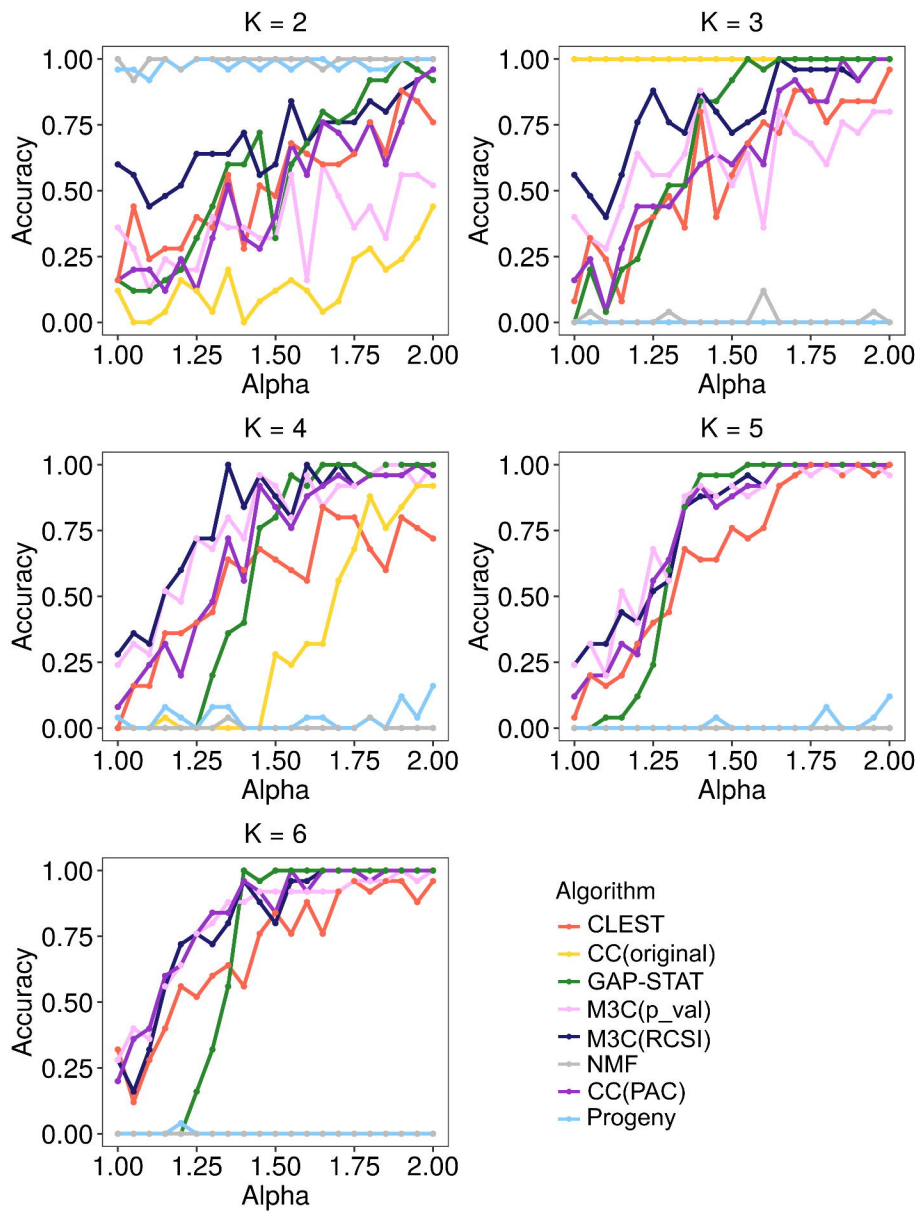
Figure 1


a
PC2
PC1
6
3
0
-3
-6
b
Consensus clustering
CDF
Consensus index
1.00
0.75
0.50
0.25
0.00
K
2
3
4
5
6
7
8
9
10
c
Consensus clustering
PAC score
K
0.6
0.5
0.4
0.3
0.2
d
NMF consensus clustering
Cophenetic coefficient
K
0.9
0.8
0.7

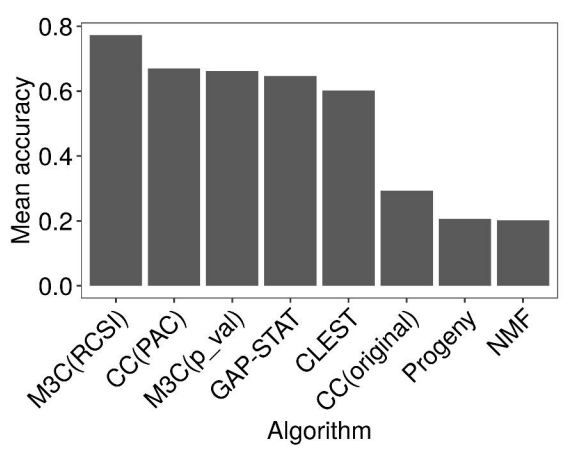

a
Samples
Features
Expression matrix (real)
Exploratory analysis (PCA, tSNE)
For i in 1:n; do
Correlation structure
Samples
Features
Expression matrix (reference)
Function M3C-reference
Function M3C-real
Run consensus clustering algorithm (using PAM, K-means, or self-tuning spectral clustering)
Compare real to reference
Monte Carlo p value
Beta p value
Relative cluster stability index (RCSI)
Compute dendogram and pairwise p values
Confirm structure (PCA, tSNE)
b
Simulated (negative control)
Not significant
c
Simulated (positive control)
Significant
d
Systemic lupus erythematosus (2016)
Not significant
e
Breast cancer (2012)
Not significant
PC1
PC2
PAC score (real)
RCSI
-log10p
K

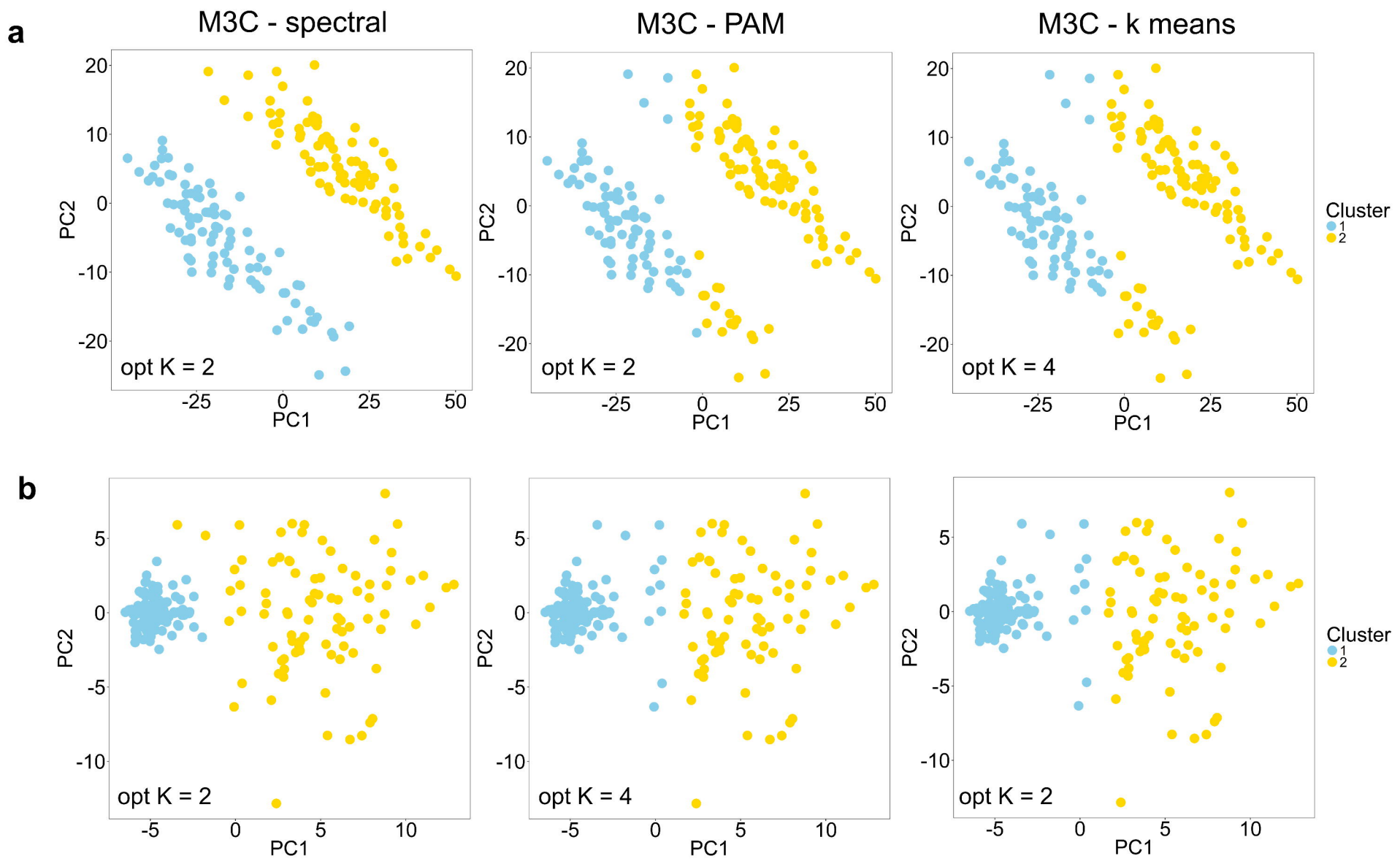

a
GBM (2008), original K = 4, M3C K = 4
b
Ovarian (2011), original K = 4, M3C K = 5
c
Lung cancer (2012), original K = 4, M3C K = 2
d
Diffuse glioma (2016), original K = 4, M3C K = 8
e
Paraganglioma (2017), original K = 4, M3C K = 6
PAC score
RCSI
-log10p
Cophenetic coefficient
K
Consensus clustering
M3C
M3C
NMF

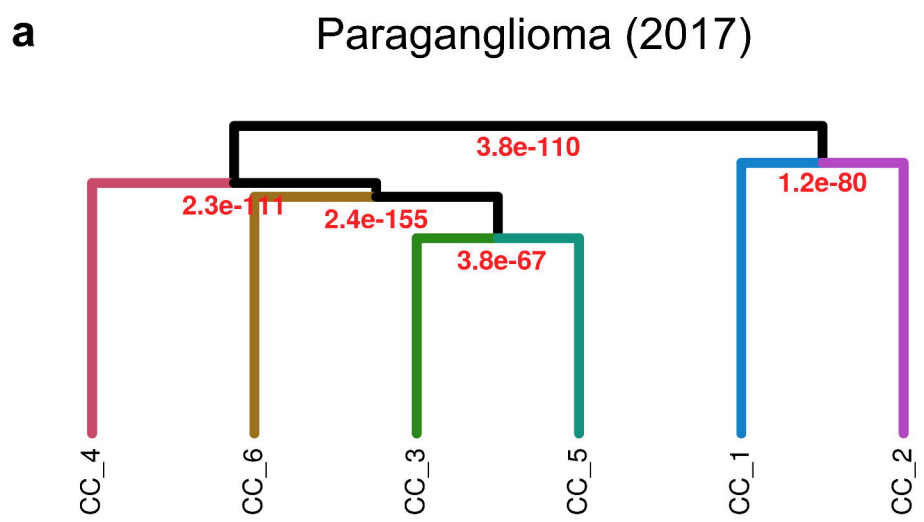
a

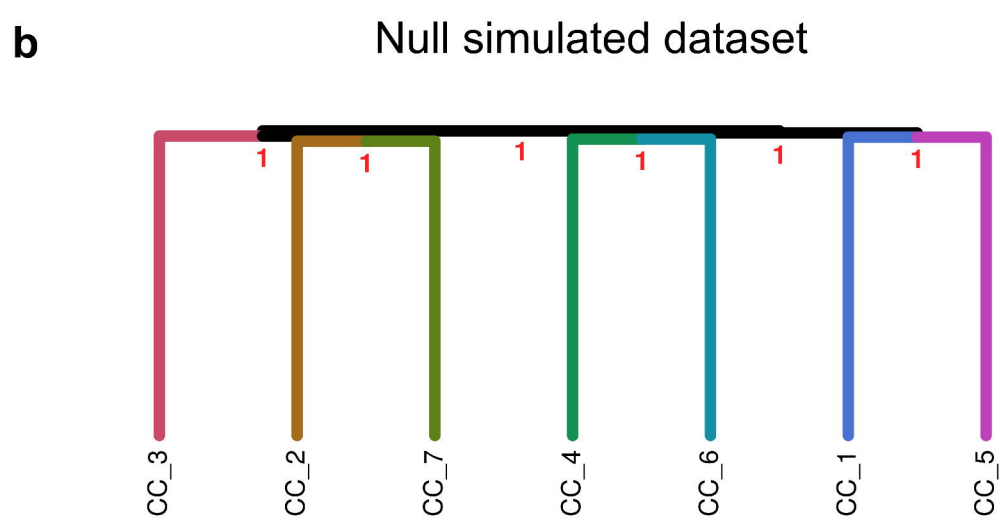

K = 2
K = 3
K = 4
K = 5
K = 6
Accuracy
Alpha
1.00
0.75
0.50
0.25
0.00
1.00 1.25 1.50 1.75 2.00
Algorithm
CLEST
CC(original)
GAP-STAT
M3C(p_val)
M3C(RCSI)
NMF
CC(PAC)
Progeny

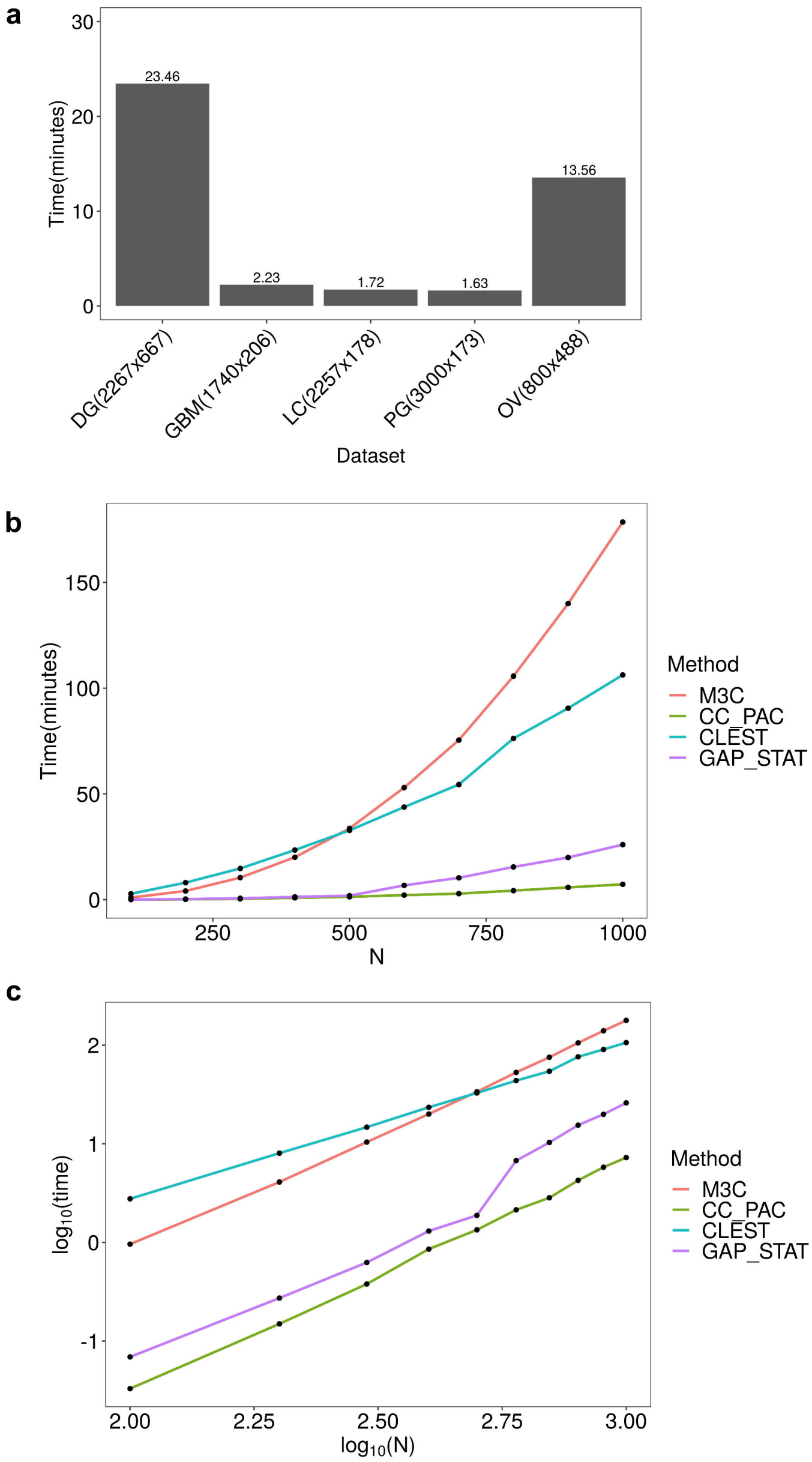
b


Mean accuracy
0.8
0.6
0.4
0.2
0.0
M3C(RCSI)
CC(PAC)
M3C(p_val)
GAP-STAT
CLEST
CC(original)
Progeny
NMF
Algorithm


a
M3C - spectral
M3C - PAM
M3C - k means
opt K = 2
opt K = 2
opt K = 4
PC1
PC2
Cluster
1
2
b
opt K = 2
opt K = 4
opt K = 2
PC1
PC2
Cluster
1
2


a
Paraganglioma (2017)
3.8e-110
2.3e-111
2.4e-155
3.8e-67
1.2e-80
CC_4
CC_6
CC_3
CC_5
CC_1
CC_2


b
Null simulated dataset
1
1
1
1
1
1
CC_3
CC_2
CC_7
CC_4
CC_6
CC_1
CC_5


a
Time(minutes)
30
20
10
0
23.46
2.23
1.72
1.63
13.56
DG(2267x667)
GBM(1740x206)
LC(2257x178)
PG(3000x173)
OV(800x488)
Dataset
b
Time(minutes)
150
100
50
0
250
500
750
1000
N
Method
M3C
CC_PAC
CLEST
GAP_STAT
c
log10(time)
2
1
0
-1
2.00
2.25
2.50
2.75
3.00
log10(N)
Method
M3C
CC_PAC
CLEST
GAP_STAT