Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Consensus clustering for Bayesian mixture models

Stephen Coleman [1,*], Paul DW Kirk [1, 2] and Chris Wallace [1,2*]

[1]MRC Biostatistics Unit, University of Cambridge, Cambridge, CB2 0SR, United Kingdom and
[2]Department of Medicine, University of Cambridge, Cambridge, CB2 0AW, United Kingdom.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Bayesian mixture models have attractive features and been successfully applied in a diverse range of settings. Inference of these models is normally performed using Markov-chain Monte Carlo (MCMC) methods. In high dimensions MCMC methods often explore a limited range of partitions, with a lack of overlap between chains (i.e. a lack of convergence) frequently present.
Results: We extend the ensemble method, Consensus clustering (CC), to Bayesian mixture models. We compare CC to inference performed using MCMC methods and also to `mclust`, a popular mixture model R package. We show that CC can be extended to Bayesian integrative clustering models. CC is then demonstrated on real datasets in both the single dataset and multiple dataset context. CC is shown to capture more modes in the clustering distribution than either the maximum-likelihood estimate (MLE) or any individual Markov chain. CC also reduces the computation time required when a parallel environment is present compared to Bayesian inference.
Availability: None?
Contact: stephen.coleman@mrc-bsu.cam.ac.uk
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

From defining a taxonomy of disease to creating molecular sets, grouping items can help us to understand and make decisions about complex biological problems. For example, clustering patients based upon disease characteristics and personal omics data allows the reaction and progression of individuals within a cluster to inform treatment decisions about other members of the same group. In another setting, defining gene regulatory networks can provide valuable insights into the causal mechanisms driving molecular products and may be used in diagnosis or for drug targets (Emmert-Streib *et al.*, 2014).

Clustering data is about maximising the quantity of relevant data for each individual within a specific analysis, reducing from the complexity of the entire set without contracting to the individual level. The clustering approximates complex variation within the dataset, enabling local downstream analysis and decisions upon each group rather than at the global level.

The act of identifying such groups is referred to as "cluster analysis". Traditional methods such as k-means clustering (Lloyd, 1982; Forgy, 1965) or hierarchical clustering condition upon a user inputted choice of K, the number of occupied clusters present. Normally different choices of K are compared under some metric such as silhouette or based upon the within-cluster sum of squared errors as a function of K. There exists an alternative school of clustering, model-based clustering, which embeds the cluster analysis within a formal, statistical framework. This means that choice of K is a model selection problem with all the associated literature.

In many analyses or decision making processes, understanding how certain the clustering is van be vital. For example, in clustering patients there might be individuals with almost equal probability of being allocated between a number of clusters. Knowing which individuals have uncertain membership could strongly influence decisions about treatment. However in many cluster analyses only a point clustering is estimated and thus no one is no wiser about which individuals are boundary members of clusters. Bayesian mixture models provide an uncertainty quantification that allows one to include this uncertainty in a formal manner in downstream analyses and decisions.

Furthermore, one might believe that the number of clusters present might itself be uncertain, that it is a random variable that should be inferred from the data. Bayesian mixture models can treat K in this way, thereby incorporating uncertainty about K into the the allocation uncertainty and reducing some of the subjectivity of choosing K. In this way the number of components and the component parameters are modelled jointly, in

contrast with many clustering methods. Furthermore, if there is a prior belief about K, this may be included due to the Bayesian nature of the model.

We write the basic mixture model for independent items $X = (x_1, \ldots, x_N)$ as

$$x_i \sim \sum_{k=1}^{K} \pi_k f(x_i|\theta_k) \qquad \text{independently for } i = 1, \ldots, N \quad (1)$$

where $f(\cdot|\theta)$ is some family of densities parametrised by θ. Here K, the number of subgroups in the population, $\{\theta_k\}_{k=1}^K$, the component parameters, and $\pi = (\pi_1, \ldots, \pi_K)$, the component weights are the objects to be inferred. In the context of *clustering*, such a model arises due to the belief that the population from which the random sample under analysis has been drawn consists of K unknown groups proportional to π. In this setting it is natural to include a latent *allocation variable*, $c = (c_1, \ldots, c_N)$, to indicate which group each item is drawn from. The model is then

$$\begin{aligned} p(c_i = k) &= \pi_k \quad \text{for } k = 1, \ldots, K, \\ x_i|c_i &\sim f(x_i|\theta_k) \quad \text{independently for } i = 1, \ldots, N. \end{aligned} \quad (2)$$

The joint model can then be written

$$p(X, c, K, \pi, \theta) = p(X|c, \pi, K, \theta)p(\theta|c, \pi, K)p(c|\pi, K)p(\pi|K)p(K)$$

In our case $f(\cdot|\theta)$ is the probability density function of a Gaussian distribution and $\theta = (\mu, \sigma^2)$. A common assumption is that the density of each feature is independent, with the $\theta_k = (\theta_{k1}, \ldots, \theta_{kP})$ for all $k = 1, \ldots, K$. Furthermore, conditional independence is assumed between certain parameters such that the model reduces to

$$p(X, c, \theta, \pi, K) = \prod_{i=1}^{N} p(x_i|c_i, \theta_{c_i}) \prod_{i=1}^{N} p(c_i|\pi, K)p(\pi|K)p(\theta|K)p(K). \quad (3)$$

Bayesian inference of mixture models is normally performed using Markov-chain Monte Carlo (MCMC) methods. However, MCMC methods in general are known to display problematic behaviour in high-dimensions (Robert *et al.*, 2018; Yao *et al.*, 2020; Chandra *et al.*, 2020). One of these problems is the difficulty in exploring a range of modes in the posterior distribution of the clustering. This means that there is a lack of convergence across chains and the distribution in any single chain is highly concentrated upon a very small number (often one) possible clustering(s). Another problem that emerges in large or high-dimensional datasets is that individual iterations of the MCMC sampler become slow and a long chain can become infeasible.

An alternative approach to clustering is through the use of ensembles of models. Ensembles are often better at representing modes within parameters than the individual learners Ghaemi *et al.* (2011). Ensembles also offer reductions in computational runtime because most ensemble methods enable use of a parallel environment to improve computation speed Ghaemi *et al.* (2009). Consensus clustering Monti *et al.* (2003) is an ensemble method for cluster analysis, previously implemented using k-means clustering (refer to original paper) in the R package `ConsensusClusterPlus` Wilkerson *et al.* (2010). This flavour of consensus clustering has been applied to cancer subtyping Lehmann *et al.* (2011); Verhaak *et al.* (2010) and identifying subclones in single cell analysis Kiselev *et al.* (2017). Consensus clustering applies R independent runs of the clustering algorithm to peturbed versions of the dataset (i.e. sampled with replacement) and combines the R final partitions in a *Consensus matrix* (CM) to infer a final clustering. The consensus matrix is a symmetric matrix with the $(i, j)^{th}$ entry being the proportions of model runs for which the i^{th} and j^{th} items are clustered together. For a single partition the *coclustering matrix* represents this information, being a binary matrix with the $(i, j)^{th}$ entry indicating if items i and j are allocated to the same cluster.

Data: $X = (x_1, \ldots, x_N)$
Input: A resampling scheme *Resample*
A clustering algorithm *Cluster*
Number of resampling iterations S
Set of cluster numbers to try $\mathcal{K} = \{K_1, \ldots, K_{max}\}$
Output: A predicted clustering, $\hat{Y}$
The predicted number of clusters present $\hat{K}$
begin
  for $K \in \mathcal{K}$ **do**
    `/* initialise an empty Consensus Matrix */`
    $\mathbf{M}^{(K)} \leftarrow \mathbf{0}_{N \times N}$;
    for $s = 1$ **to** S **do**
      $X^{(s)} \leftarrow Resample(X)$;
      `/* Cluster the peturbed dataset, represented in a coclustering matrix */`
      $\mathbf{B}^{(s)} \leftarrow Cluster(X^{(s)}, K)$;
      $\mathbf{M}^{(K)} \leftarrow \mathbf{M}^{(K)} + \mathbf{B}^{(s)}$;
    end
    $\mathbf{M}^{(K)} \leftarrow \frac{1}{S}\mathbf{M}^{(K)}$;
  end
  $\hat{K} \leftarrow$ best $K \in \mathcal{K}$ based upon all $\mathbf{M}^{(K)}$;
  $\hat{Y} \leftarrow$ partition X based upon $\mathbf{M}^{(\hat{K})}$;
end

Algorithm 1: Consensus Clustering algorithm

We extend consensus clustering to Bayesian mixture models. We show via simulation that ensembles consisting of short chains are sufficient to uncover meaningful structure in a number of scenarios including some within which a Gibbs sampler becomes trapped in individual modes for any reasonable length of runtime. The chains are both short and independent, thus their individual runtime is far shorter than the chains traditionally used for Bayesian inference and may also be run in parallel. This means that consensus clustering of Bayesian mixture models offers significant reductions in runtime without sacrifices in performance. As the ensemble can describe multiple modes, the uncertainty present in the consensus matrix can be more representative of the data than the individual modes captured by any single chain.

We then apply our algorithm to the multiple dataset setting and the extension of Bayesian mixture models, Multiple Dataset Integration (MDI). We show on three datasets from the original MDI paper that Consensus clustering performs similarly to Bayesian inference of this model, and then using more modern, larger data that Consensus clustering enables implementation of such models in scenarios where the problem of long runtimes and poor mixing previously discouraged this.

2 Methods

2.1 Consensus clustering for Bayesian mixture models

We extend CC to use Bayesian mixture models as the underlying model. This offers the ability to include a prior distribution on parameters and

inference of the number of occupied clusters present, maintaining two of the key attractions of Bayesian model-based clustering while losing the asymptotic guarantees of Bayesian inference. In this case the algorithm is simplified as it is not necessary to try a range of possible clusters present. Furthermore, the dataset is not perturbed as described in algorithm 1 for two reasons. First, the overfit mixture can capture individuals in singletons and thus is more robust to outliers than k-means. Secondly, the MCMC method driving each model offers diversity of partitions when combined with different initialisations. The method is described in algorithm 2.

Data: $X = (x_1, \ldots, x_N)$
Input: A Bayesian mixture model with membership vector
$c = (c_1, \ldots, c_N)$
A clustering algorithm that generates samples *Cluster*
The number of chains to run, S
The number of iterations within each chain, R
Output: A predicted clustering, $\hat{Y}$
The consensus matrix $\mathbf{M}$
begin
  /* initialise an empty Consensus Matrix */
  $\mathbf{M} \leftarrow \mathbf{0}_{N \times N}$;
  for $s = 1$ **to** S **do**
    /* set the random seed controlling initialisation and MCMC moves */
    $set.seed(s)$;
    /* initialise a random partition on X drawn from the prior distribution */
    $Y_{(0,s)} \leftarrow Initialise(X)$;
    for $r = 1$ **to** R **do**
      /* generate a markov chain for the membership vector */
      $Y_{(r,s)} \leftarrow Cluster(c, r)$;
    end
    /* create a coclustering matrix from the R^{th} sample */
    $\mathbf{B}^{(s)} \leftarrow Y_{(R,s)}$;
    $\mathbf{M} \leftarrow \mathbf{M} + \mathbf{B}^{(s)}$;
  end
  $\mathbf{M} \leftarrow \frac{1}{S}\mathbf{M}$;
  $\hat{Y} \leftarrow$ partition X based upon $\mathbf{M}$;
end

Algorithm 2: Consensus Clustering for Bayesian mixture models

2.2 Predicting a clustering from CMs

We use the `maxpear` function Fritsch *et al.* (2009) from the R package `mcclust` Fritsch (2012) to infer a point clustering from CMs. This function was designed to perform inference upon the posterior similarity matrix (**PSM**) from the samples of a single long chain (this is analogous to a CM, except the partitions are all drawn from a single Markov chain), predicting a clustering that has maximum posterior expected adjusted Rand index Hubert and Arabie (1985) with the true clustering. In the context of the CM, the function does not have this interpretation. However, the method produces a kind of "sample average clustering" based upon all sampled partitions. This appears a sensible method for predicting a point clustering from the CM, averaging over each learner in the ensemble.

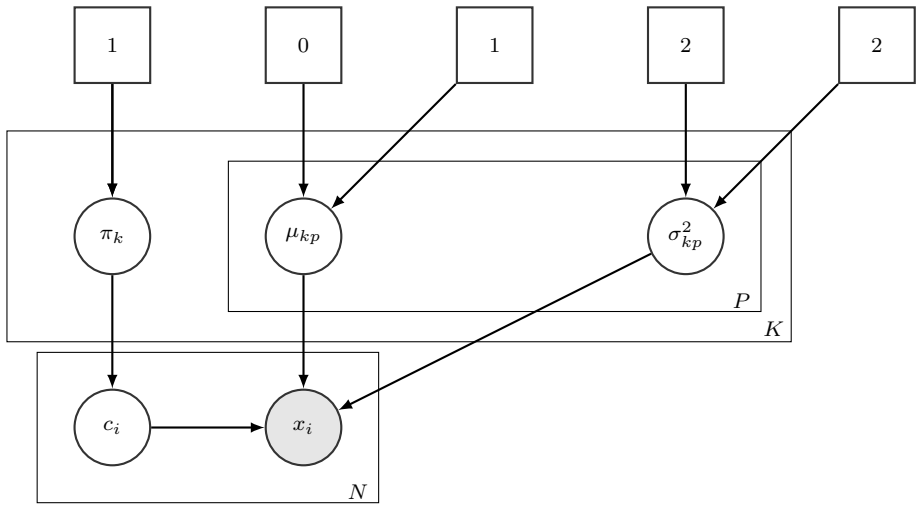

Fig. 1. Directed acyclic graph for the Bayesian mixture of Gaussians used.

2.3 Data generating mechanism

The data generating model is a finite mixture model (as per equation 2) with independent features. Within this model there exist "irrelevant features" (Law *et al.*, 2003) that have global parameters rather than component specific parameters. The generating model is then, extending from equation 3 with the change that K is now fixed,

$$p(X, c, \theta, \pi|K) = \prod_{i=1}^{N}\prod_{p=1}^{P} p(x_{ip}|c_i, \theta_{c_i p})^{(1-\phi_p)} p(x_{ip}|\theta_p)^{\phi_p} \times \prod_{i=1}^{N} p(c_i|\pi, K) p(\pi|K) p(\theta|K)$$

with $\phi_p = 1$ indicating that the p^{th} feature is relevant.

In the simulation study described here, the model is a mixture of *Gaussian* distribution and thus $\theta_{kp} = (\mu_{kp}, \sigma^2_{kp})$. The prior distributions used on the mixture parameters are

$$\pi \sim \text{Dirichlet}(\alpha, \ldots, \alpha), \qquad \mu_{kp} \sim \mathcal{N}(\xi, \kappa), \qquad \sigma^2 \sim \Gamma^{-1}(a, b).$$

The conditional independences and the priors used are shown in the directed acyclic graph (**DAG**) in figure 1. The total number of occupied and empty components is set to $K_{max} = 50$. This and the choice of priors are the defaults in the software provided by Mason *et al.* (2016).

Data is generated using algorithm 3.

2.4 Performance quantification

We use the Adjusted Rand Index as our metric for the quality of the point clustering inferred by each method, comparing this estimate with the generating labels produced by algorithm 3. This is a measure of "predictive performance", the ability of the methods to infer a single partition that and its similarity to the truth. We also attempt to summarise the uncertainty quantification from each by computing the Frobenius Norm between the true coclustering matrix and the

- consensus matrix for consensus clustering,
- posterior similarity matrix for the Bayesian inference, and
- coclustering matric for `mclust`.

Input: Distance between means Δ_μ
A common standard deviation σ^2
A number of clusters K
The number of items to generate in total N
The number of features to generate in total P
An indicator vector of feature relevance $\phi = (\phi_1, \ldots, \phi_P)$
The expected proportion of items in each cluster
$\pi = (\pi_1, \ldots, \pi_K)$
A method for sampling x times from the array y, with weights π: *Sample*(y, x)
A method for permuting a vector x: *Permute*(x)
A method for generating a value from a univariate Gaussian distribution with mean μ and standard deviation σ^2: *Gaussian*(μ, σ^2)
Output: A dataset, X
The generating cluster labels $c = (c_1, \ldots, c_N)$
begin
  `/* initialise the empty data matrix */`
  $X \leftarrow 0_{N \times P}$;
  `/* create a matrix of` K `means */`
  $\mu \leftarrow (\Delta_\mu, \ldots, K\Delta_\mu)$;
  `/* generate the allocation vector */`
  $c \leftarrow$ *Sample*$(1:K, N, \pi)$;
  $\mathbf{M} \leftarrow \mathbf{0}_{N \times N}$;
  for $p = 1$ **to** P **do**
    `/* Test if the feature is relevant, if relevant generate data from a mixture of univariate Gaussians, otherwise draw all items from the same distribution */`
    if $\phi_p = 1$ **then**
      $\nu \leftarrow$ *Permute*(μ);
      for $n = 1$ **to** N **do**
        $X(n, p) \leftarrow$ *Gaussian*(ν_{c_n}, σ^2)
      end
    end
    if $\phi_p = 0$ **then**
      for $n = 1$ **to** N **do**
        $X(n, p) \leftarrow$ *Gaussian*$(0, \sigma^2)$
      end
    end
  end
  `/* Mean centre and scale the data */`
  $X \leftarrow Normalise(X)$
end

Algorithm 3: Data generation for a mixture of Gaussian with independent features. This algorithm is implemented in the `generateSimulationDataset` function from the `mdiHelpR` R package.

The Frobenius Norm will provide some information if the above matrices correspond at all to the true coclustering matrix, but if no method has performed well then this Norm will reward the *singleton solution* wherein all items are allocated to individual clusters. This means that a visual inspection of the PSMs and CMs is also required. As `mclust` provides only a point estimate the ARI between this and the truth will contain the required information.

The runtime of each MCMC chain is calculated using the terminal command `time`, measured in milliseconds.

2.5 Bayesian model convergence

To assess within-chain convergence of our Bayesian inference we use the Geweke statistic Geweke *et al.* (1991). Of the chains that appear to behave properly we then asses across chain convergence using $\hat{R}$ Gelman *et al.* (1992) and the recent extension provided by Vats and Knudson, 2018.

If a chain has reached its stationary distribution the Geweke statistic is expected to be normally distributed. Normality is tested for using a Shapiro-Wilks test Shapiro and Wilk (1965).

The Vats and Knudson extension of $\hat{R}$ is a summary statistic for the entire chain; this is the primary indicator of failure for convergence, but a visualisation of the original $\hat{R}$ diagnostic is also considered.

3 Examples

3.1 Simulations

We compare consensus clustering of Bayesian mixture models to a traditional inference using several long chains of 1 million iterations (thinning to every thousandth) and an maximum-likelihood estimator as implement in the R package `mclust` Scrucca *et al.*, 2016. These are compared within a range of simulations described in table 1. In this article we describe the results of three of these in more detail with the remainder described in the supplementary material.

We test different scenarios that change various parameters in this model. The scenarios tested and there defining parameters are shown in table 1. In this case the number of relevant features (P_s) is $\sum_p \phi_p$, and $P_n = P - P_s$.

Table 1. Parameters defining the simulation scenarios as used in generating data and labels. Results for the Simple 2D, the first Small N, large P and final Irrelevant features scenarios are shown in this report, please see the supplementary material for additional results. The number of relevant features (P_s) is $\sum_p \phi_p$, and $P_n = P - P_s$.

Scenario	N	P_s	P_n	K	Δ_μ	σ^2	π
Simple 2D	100	2	0	5	3.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
No structure	100	0	2	1	0.0	1	1
Base Case	200	20	0	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Standard deviation	200	20	0	5	1.0	3	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Standard deviation	200	20	0	5	1.0	5	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Irrelevant features	200	20	10	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Irrelevant features	200	20	20	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Irrelevant features	200	20	100	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Varying proportions	200	20	0	5	1.0	1	$(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16})$
Varying proportions	200	20	0	5	0.4	1	$(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16})$
Small N, large P	50	500	0	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Small N, large P	50	500	0	5	0.2	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$

The examples included in this article are

- a low-dimensional dataset,
- a wide dataset representative of the *small N, large P* paradigm prevalent in genetics, and
- a dataset with a large number of irrelevant features.

The first of these is expected to be the setting where `mclust` and the individual long chains behave well. In the other simulations increasing dimensionality means that mixing problems can emerge and the chains become liable to being trapped in individual modes. Within each simulation 100 datasets are generated following algorithm 3. To each of these datasets

- `mclust` (for a range of possible K),

- 10 chains of 1 million iterations, thinning to every thousandth sample for the overfitted Bayesian mixture model, and
- a variety of consensus clustering ensembles defined by inputs of S chains and R iterations within each chain (see algorithm 2) with $S \in \{1, 10, 30, 50, 100\}$ and $R \in \{1, 10, 100, 1000, 10000\}$,

are applied.

Results

3.2 Simulations

A summary of the results for a selection of the simulation scenarios are shown in table 2 and figure 2. In the strong signal scenarios (i.e. noise free and clearly distinguishable generating clusters), `mclust` performs very well, correctly identifying the true structure However, a large number of irrelevant features completely erodes the ability of `mclust` to uncover meaningful structure. The pooled samples from multiple long chains performs consistently very well. This is not surprising as the pooling effect means that any multi-modality present in the data does not present any degree of problem. In this case the pooled samples, themselves a consensus of 10 models, acts as an upper bound on the more realistic implementations of consensus clustering. Consensus clustering does consistently uncover structure in the data. With sufficiently large ensembles and chain depth, consensus clustering is close to the pooled Bayesian samples in predictive performance. In terms of the Frobenius norm, many of the consensus clustering results have significant overlap with the pooled long chains.

Table 2. Mean ARI for 100 datasets within three simulation scenarios between the generating labels and the predicted clustering for a subset of methods. Consensus (r, s) indicates consensus clustering using the r^{th} sample from s chains.

Model	Simple 2D	Small N, large P	Irrelevant features 100
Bayesian (Pooled)	0.669	0.999	0.000
Maximum likelihood (Mclust)	0.970	1.000	0.873
Consensus (100, 50)	0.517	0.999	0.944
Consensus (10000, 100)	0.570	0.999	0.385
Consensus (10, 10)	0.362	0.997	

Figure 3 shows an example of different chains becoming trapped in different modes and failing to explore a common partition space.

4 Discussion

In this article, we have extended Consensus clustering to Bayesian mixture models. This overcomes the problem of mixing for MCMC based clustering methods and also improves the speed at which an analysis can be performed. The proposed method has demonstrated good performance on simulation studies, having similar ability to uncover structure as more traditional approaches. It has better ability to represent several modes in the data than individual chains and is significantly faster, while retaining the ability to infer K, the number of occupied components present.

We expect that our method will be useful to researchers analysing high-dimensional data where the runtime of MCMC methods becomes too onerous and multi-modality is more likely to be present.

5 Conclusion

(Table **??**) Text Text Text Text Text Text Text Text Text

1. this is item, use enumerate
2. this is item, use enumerate
3. this is item, use enumerate

Acknowledgements

Text Text Text Text Text Text Text Text. nt to know about text text text text

Funding

This work has been supported by the... Text Text Text Text.

References

Chandra, N. K. *et al.* (2020). Bayesian clustering of high-dimensional data. *arXiv preprint arXiv:2006.02700*.

Emmert-Streib, F. *et al.* (2014). Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Frontiers in cell and developmental biology*, **2**, 38.

Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, **21**, 768–769.

Fritsch, A. (2012). *mcclust: Process an MCMC Sample of Clusterings*. R package version 1.0.

Fritsch, A. *et al.* (2009). Improved criteria for clustering based on the posterior similarity matrix. *Bayesian analysis*, **4**(2), 367–391.

Gelman, A. *et al.* (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, **7**(4), 457–472.

Geweke, J. *et al.* (1991). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*, volume 196. Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN.

Ghaemi, R. *et al.* (2009). A survey: clustering ensembles techniques. *World Academy of Science, Engineering and Technology*, **50**, 636–645.

Ghaemi, R. *et al.* (2011). A review: accuracy optimization in clustering ensembles using genetic algorithms. *Artificial Intelligence Review*, **35**(4), 287–318.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, **2**(1), 193–218.

Kiselev, V. Y. *et al.* (2017). Sc3: consensus clustering of single-cell rna-seq data. *Nature methods*, **14**(5), 483–486.

Law, M. H. *et al.* (2003). Feature selection in mixture-based clustering. In *Advances in neural information processing systems*, pages 641–648.

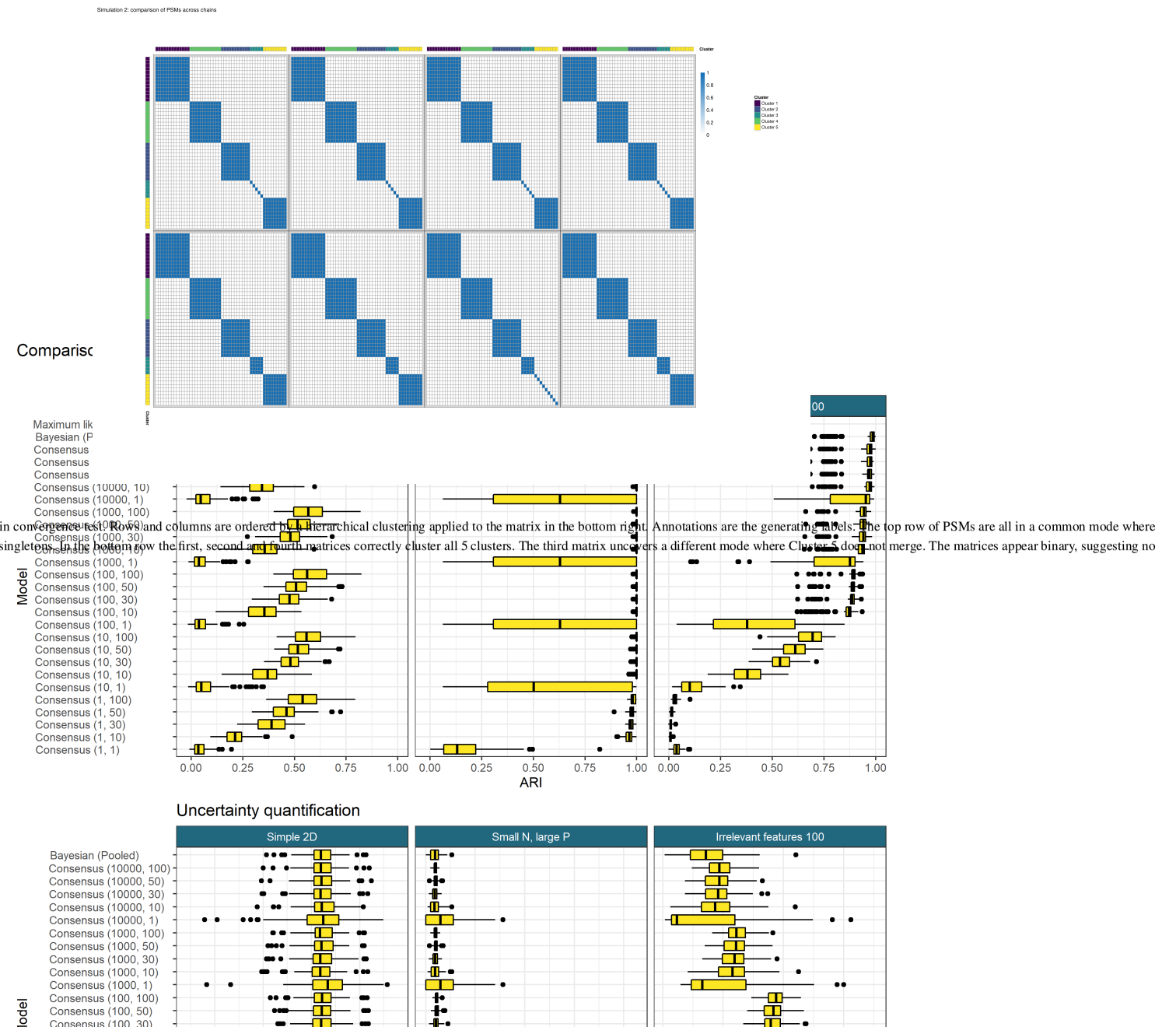

Fig. 3. PSMs for 8 chains that pass within chain convergence test. Rows and columns are ordered by hierarchical clustering applied to the matrix in the bottom right. Annotations are the generating labels. The top row of PSMs are all in a common mode where Cluster 3 does not merge instead remaining as singletons. In the bottom row the first, second and fourth matrices correctly cluster all 5 clusters. The third matrix uncovers a different mode where Cluster 5 does not merge. The matrices appear binary, suggesting no diversity in the partitions being sampled.

Lehmann, B. D. *et al.* (2011). Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies. *The Journal of clinical investigation*, **121**(7), 2750–2767.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, **28**(2), 129–137.

Mason, S. A. *et al.* (2016). Mdi-gpu: accelerating integrative modelling for genomic-scale data using gp-gpu computing. *Statistical applications in genetics and molecular biology*, **15**(1), 83–86.

Monti, S. *et al.* (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, **52**(1-2), 91–118.

Robert, C. P. *et al.* (2018). Accelerating mcmc algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, **10**(5), e1435.

Scrucca, L. *et al.* (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, **8**(1), 289–317.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, **52**(3/4), 591–611.

Vats, D. and Knudson, C. (2018). Revisiting the gelman-rubin diagnostic. *arXiv preprint arXiv:1812.09384*.

Verhaak, R. G. *et al.* (2010). Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh1, egfr, and nf1. *Cancer cell*, **17**(1), 98–110.

Wilkerson *et al.* (2010). Consensusclusterplus: a class discovery tool with confidence assessments and item tracking. *Bioinformatics*, **26**(12), 1572–1573.

Yao, Y. *et al.* (2020). Stacking for non-mixing bayesian computations: The curse and blessing of multimodal posteriors. *arXiv preprint arXiv:2006.12335*.