RESEARCH

Consensus clustering for Bayesian mixture models

Stephen Coleman[1*], Paul D.W. Kirk[1,2†] and Chris Wallace[1,2†]

Correspondence:
tephen.coleman@mrc-
su.cam.ac.uk
MRC Biostatistics Unit
niversity of Cambridge,
ambridge, UK
ull list of author information is
vailable at the end of the article
Equal contributor

Abstract

Background: Cluster analysis is an integral part of precision medicine and systems biology, used to define groups of patients or biomolecules. Consensus clustering is an ensemble approach that is widely used in these areas, which combines the output from multiple runs of a non-deterministic clustering algorithm. Here we consider the application of consensus clustering to a broad class of heuristic clustering algorithms that can be derived from Bayesian mixture models (and extensions thereof) by adopting an early stopping criterion when performing sampling-based inference for these models. While the resulting approach is non-Bayesian, it inherits the usual benefits of consensus clustering, particularly in terms of computational scalability and providing assessments of clustering stability/robustness.

Results: In simulation studies, we show that our approach can successfully uncover the target clustering structure, while also exploring different plausible clusterings of the data. We show that, when a parallel computation environment is available, our approach offers significant reductions in runtime compared to performing sampling-based Bayesian inference for the underlying model, while retaining many of the practical benefits of the Bayesian approach, such as exploring different numbers of clusters. We propose a heuristic to decide upon ensemble size and the early stopping criterion, and then apply consensus clustering to a clustering algorithm derived from a Bayesian integrative clustering method. We use the resulting approach to perform an integrative analysis of three 'omics datasets for budding yeast and find clusters of co-expressed genes with shared regulatory proteins. We validate these clusters using data external to the analysis. These clusters can help assign likely function to understudied genes, for example *GAS3* clusters with histones active in S-phase, suggesting a role in DNA replication.

Conclusions: Our approach can be used as a wrapper for essentially any existing sampling-based Bayesian clustering implementation, and enables meaningful clustering analyses to be performed using such implementations, even when computational Bayesian inference is not feasible, e.g. due to poor scalability. This enables researchers to straightforwardly extend the applicability of existing software to much larger datasets, including implementations of sophisticated models such as those that jointly model multiple datasets.

Keywords: cluster analysis; cell cycle; ensemble learning; integrative clustering

Background

From defining a taxonomy of disease to creating molecular sets, grouping items can help us to understand and make decisions using complex biological data. For example, grouping patients based upon disease characteristics and personal omics data may allow the identification of more homogeneous subgroups, enabling stratified medicine approaches. Defining and studying molecular sets can improve our understanding of biological systems as these sets are more interpretable than their constituent members (1), and study of their interactions and perturbations may have ramifications for diagnosis and drug targets (2, 3). The act of identifying such groups is referred to as *cluster analysis*. Many traditional methods such as K-means clustering (4, 5) condition upon a fixed choice of K, the number of clusters. These methods are often heuristic in nature, relying on rules of thumb to decide upon a final value for K. For example, different choices of K are compared under some metric such as silhouette (6) or the within-cluster sum of squared errors (**SSE**) as a function of K. Moreover, K-means clustering can exhibit sensitivity to initialisation, necessitating multiple runs in practice (7).

Another common problem is that traditional methods offer no measure of the stability or robustness of the final clustering. Returning to the stratified medicine example of clustering patients, there might be individuals that do not clearly belong to any one particular cluster; however if only a point estimate is obtained, this information is not available. Ensemble methods address this problem, as well as reducing sensitivity to initialisation. These approaches have had great success in supervised learning, most famously in the form of Random Forest (8) and boosting (9). In clustering, consensus clustering (10) is a popular method which has been implemented in `R` (11) and to a variety of methods (12, 13) and been applied to problems such as cancer subtyping (14, 15) and identifying subclones in single cell analysis (16). Consensus clustering uses W runs of some base clustering algorithm (such as K-means). These W proposed partitions are commonly compiled into a *consensus matrix*, the $(i, j)^{th}$ entries of which contain the proportion of model runs for which the i^{th} and j^{th} individuals co-cluster (for this and other definitions see section 1 of the Supplementary Material), although this step is not fundamental to consensus clustering and there is a large body of literature aimed at interpreting a collection of partitions (see, e.g., 17–19). This consensus matrix provides an assessment of the stability of the clustering. Furthermore, ensembles can offer reductions in computational runtime because the individual members of the ensemble are often computationally inexpensive to fit (e.g, because they are fitted using only a subset of the available data) and because the learners in most ensemble methods are independent of each other and thus enable use of a parallel environment for each of the quicker model runs (20).

Traditional clustering methods usually condition upon a fixed choice of K, the number of clusters with the choice of K being a difficult problem in itself. In consensus clustering, Monti *et al.* (10) proposed methods for choosing K using the consensus matrix and Ünlü *et al.* (21) offer an approach to estimating K given the collection of partitions, but each clustering run uses the same, fixed, number of clusters. An alternative clustering approach, mixture modelling, embeds the cluster analysis within a formal, statistical framework (22). This means that models can be compared formally, and problems such as the choice of K can be addressed as a model selection problem (23). Moreover, *Bayesian mixture models* can be used to try to directly infer K from the data. Such inference can be performed through use of a Dirichlet Process mixture model ~~(24, 25)~~(25–27), a mixture of finite mixture models (28, 29) or an over-fitted mixture model (30). These models and their extensions have a history of successful application to a diverse range of biological problems such as finding clusters of gene expression profiles (31), cell types in flow cytometry (32, 33) or scRNAseq experiments (34), and estimating protein localisation (35).

Bayesian mixture models can be extended to jointly model the clustering across multiple datasets (36, 37) (section 2 of the Supplementary Material).

Markov chain Monte Carlo (MCMC) methods are the most common tool for performing computational Bayesian inference. In Bayesian clustering, they are used to draw a collection of clustering partitions from the posterior distribution. However, in practice, chains can become stuck in local posterior modes preventing convergence (see, e.g., the Supplementary Materials of 38) and/or can require prohibitively long runtimes, particularly when analysing high-dimensional datasets. Some MCMC methods make efforts to overcome the problem of exploration, often at the cost of increased computational cost per iteration (39). There are MCMC methods that use parallel chains to improve the scalability or reduce the bias of the Monte Carlo estimate. However, these methods have various limitations. For instance, divide-and-conquer strategies such as Asymptotically Exact, Embarrassingly Parallel MCMC ~~(40)~~ (?) use subsamples of the dataset with each chain to improve scaling with the number of items being clustered. This assumes that each subsample is representative of the population, and is less helpful in situations where we have high-dimension but only moderate sample size, such as analysis of 'omics data. Alternative approaches, such as distributed MCMC (41) and coupling (42) have to account for burn-in bias; moreover, coupling further assumes the chains meet in finite time and then stay together. In practice, a further challenge associated with these methods is that their implementation may necessitate a substantial redevelopment of existing software.

Motivated by the lack of scalability of existing implementations of sampling-based Bayesian clustering (due to prohibitive computational runtimes, as well as poor exploration, as described above), here we aim to develop a general and straightforward procedure that exploits the flexibility of these methods, but extends their applicability. Specifically, we make use of existing sampling-based Bayesian clustering implementations, but only run them for a fixed (and relatively small) number of iterations, stopping before they have converged to their target stationary distribution. Doing this repeatedly, we obtain an ensemble of clustering partitions, which we use to perform consensus clustering. We propose a heuristic for deciding upon the ensemble size (the number of learners used, W) and the ensemble depth (the number of iterations, D), inspired by the use of scree plots in Principal Component Analysis (**PCA**; 43).

We show via simulation that our approach can successfully identify meaningful clustering structures. We then illustrate the use of our approach to extend the applicability of existing Bayesian clustering implementations, using as a case study the Multiple Dataset Interation (**MDI**; 36) model for Bayesian integrative clustering applied to real data. While the simulation results serve to validate our method, it is important to also evaluate methods on real data which may represent more challenging problems. For our real data, we use three 'omics datasets relating to the cell cycle of *Saccharomyces cerevisiae* with the aim of inferring clusters of genes across datasets. As there is no ground truth available, we then validate these clusters using knowledge external to the analysis.

Material and methods

Consensus clustering for Bayesian mixture models

We apply consensus clustering to MCMC based Bayesian clustering models using the method described in algorithm 1. Our application of consensus clustering has two main parameters at the ensemble level, the chain depth, D, and ensemble width, W. We infer a point clustering from the consensus matrix using the `maxpear` function (44) from the R package `mcclust` (45) which maximises the posterior expected adjusted Rand index between the true clustering and point estimate if the matrix is composed of samples drawn from the posterior distribution (section 3 of the Supplementary Material for details). There are alternative choices of methods to infer a point estimate which minimise different loss functions (see, e.g., 46–48).

```
Data: $X = (x_1, \ldots, x_N)$
Input:
The number of chains to run, $W$
The number of iterations within each chain, $D$
A clustering method that uses MCMC methods to generate samples of clusterings
of the data Cluster$(X, d)$
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 $w = 1$ to $W$ do
        /* set the random seed controlling initialisation and MCMC
           moves                                                             */
        $set.seed(w)$;
        /* initialise a random partition on $X$ drawn from the
           prior distribution                                                */
        $Y_{(0,w)} \leftarrow Initialise(X)$;
        for $d = 1$ to $D$ do
            /* generate a markov chain for the membership vector             */
            $Y_{(d,w)} \leftarrow Cluster(X, d)$;
        end
        /* create a coclustering matrix from the $D^{th}$ sample             */
        $\mathbf{B}^{(w)} \leftarrow Y_{(D,w)}$;
        $\mathbf{M} \leftarrow \mathbf{M} + \mathbf{B}^{(w)}$;
    end
    $\mathbf{M} \leftarrow \frac{1}{W}\mathbf{M}$;
    $\hat{Y} \leftarrow$ partition $X$ based upon $\mathbf{M}$;
end
```

Algorithm 1: Consensus clustering for Bayesian mixture models.

Determining the ensemble depth and width

As our ensemble sidesteps the problem of convergence within each chain, we need an alternative stopping rule for growing the ensemble in chain depth, D, and number of chains, W. We propose a heuristic based upon the consensus matrix to decide if a given value of D and W are sufficient. We suspect that increasing W and D might continuously improve the performance of the ensemble, but we observe in our simulations that these changes will become smaller and smaller for greater values, eventually converging for each of W and D. We notice that this behaviour is analogous to PCA in that where for consensus clustering some improvement might always be expected for increasing chain depth or ensemble width, more variance will be captured by increasing the number of components used in PCA. However, increasing this number beyond some threshold has diminishing returns, diagnosed in PCA by a scree plot. Following from this, we recommend, for some set of ensemble parameters, $D' = \{d_1, \ldots, d_I\}$ and $W' = \{w_1, \ldots, w_J\}$, find the mean absolute difference of the consensus matrix for the d_i^{th} iteration from w_j chains to that for the $d_{(i-1)}^{th}$ iteration from w_j chains and plot these values as a function of chain depth, and the analogue for sequential consensus matrices for increasing ensemble width and constant depth.

If this heuristic is used, we believe that the consensus matrix and the resulting inference should be stable (see, e.g., 49, 50), providing a robust estimate of the clustering. In contrast, if there is still strong variation in the consensus matrix for varying chain length or number, then we believe that the inferred clustering is influenced significantly by the random initialisation and that the inferred partition is unlikely to be stable for similar datasets or reproducible for a random choice of seeds.

Simulation study

We use a finite mixture with independent features as the data generating model within the simulation study. Within this model there exist "irrelevant features" (51) that have global parameters rather than cluster specific parameters. The generating model is

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

for data $X = (x_1, \ldots, x_N)$, cluster label or allocation variable $c = (c_1, \ldots, c_N)$, cluster weight $\pi = (\pi_1, \ldots, \pi_K)$, K clusters and the relevance variable, $\phi \in \{0, 1\}$ with $\phi_p = 1$ indicating that the p^{th} feature is relevant to the clustering. We used a *Gaussian* density, so $\theta_{kp} = (\mu_{kp}, \sigma^2_{kp})$. We defined three scenarios and simulated 100 datasets in each (figure 1 and table 1). Additional details of the simulation process and additional scenarios are included in section 4.1 of the Supplementary Materials.

Table 1 Parameters defining the simulation scenarios as used in generating data and labels. $\Delta\mu$ is the distance between neighbouring cluster means within a single feature. 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	$\Delta\mu$	σ^2	π
2D	100	2	0	5	3.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Small N, large P	50	500	0	5	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	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$

In each of these scenarios we apply a variety of methods (listed below) and compare the inferred point clusterings to the generating labels using the Adjusted Rand Index (**ARI**, 52).

- `Mclust`, a maximum likelihood implementation of a finite mixture of Gaussian densities (for a range of modelled clusters, K),
- 10 chains of 1 million iterations, thinning to every thousandth sample for the overfitted Bayesian mixture of Gaussian densities, and

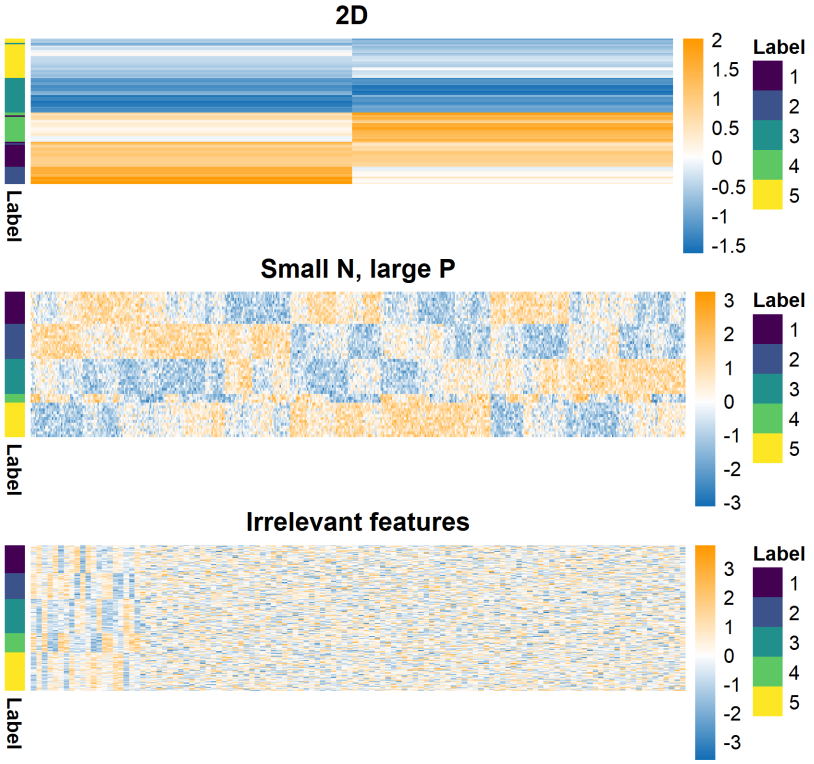

Figure 1 Example of generated datasets. Each row is an item being clustered and each column a feature of generated data. The 2D dataset (which is ordered by hierarchical clustering here) should enable proper mixing of chains in the MCMC. The small N, large P case has clear structure (observable by eye). This is intended to highlight the problems of poor mixing due to high dimensions even when the generating labels are quite identifiable. In the irrelevant features case, the structure is clear in the relevant features (on the left-hand side of this heatmap). This setting is intended to test how sensitive each approach is to noise.

- A variety of consensus clustering ensembles defined by inputs of W chains and D iterations within each chain (see algorithm 1) with $W \in \{1, 10, 30, 50, 100\}$ and $D \in \{1, 10, 100, 1000, 10000\}$ where the base learner is an overfitted Bayesian mixture of Gaussian densities.

Note that none of the applied methods include a model selection step and as such there is no modelling of the relevant variables. This and the unknown value of K is what separates the models used and the generating model described in equation 1. More specifically, the likelihood of a point X_n for each method is

$$p(X_n|\mu, \Sigma, \pi) = \sum_{k=1}^{K} \pi_k p(X_n|\mu_k, \Sigma_k), \tag{2}$$

where $p(X_n|\mu_k, \Sigma_k)$ is the probability density function of the multivariate Gaussian distribution parameterised by a mean vector, μ_k, and a covariance matrix, Σ_k, and π_k is the component weight such that $\sum_{k=1}^{K} \pi_k = 1$. The implementation of the Bayesian mixture model restricts Σ_k to be a diagonal matrix while `Mclust` models a number of different covariance structures. Note that while we use the overfitted Bayesian mixture model, this is purely from convenience and we expect that a true Dirichlet Process mixture or a mixture of mixture models would display similar behaviour in an ensemble.

The ARI is a measure of similarity between two partitions,c_1, c_2, corrected for chance, with 0 indicating c_1 is no more similar to c_2 than a random partition would be expected to be and a value of 1 showing that c_1 and c_2 perfectly align. Details of the methods in the simulation study can be found in sections 4.2, 4.3 and 4.4 of the Supplementary Material.

Mclust

`Mclust` (53) is a function from the R package `mclust`. It estimates Gaussian mixture models for K clusters based upon the maximum likelihood estimator of the parameters. It initialises upon a hierarchical clustering of the data cut to K clusters. A range of choices of K and different covariance structures are compared and the "best" selected using the Bayesian information criterion, (54) (details in section 4.2 of the Supplementary Material).

Bayesian inference

To assess within-chain convergence of our Bayesian inference we use the Geweke Z-score statistic (55). Of the chains that appear to behave properly we then asses across-chain convergence using $\hat{R}$ (56) and the recent extension provided by (57). If a chain has reached its stationary distribution the Geweke Z-score statistic is expected to be normally distributed. Normality is tested for using a Shapiro-Wilks test (58). If a chain fails this test (i.e., the associated p-value is less than 0.05), we assume that it has not achieved stationarity and it is excluded from the remainder of the analysis. The samples from the remaining chains are then pooled and a posterior similarity matrix (**PSM**) constructed. We use the `maxpear` function to infer a point clustering. For more details see section 4.3 of the Supplementary Material.

Analysis of the cell cycle in budding yeast

Datasets

The cell cycle is crucial to biological growth, repair, reproduction, and development (59–61) and is highly conserved among eukaryotes (61). . This means that understanding of the cell cycle of *S. cerevisiae* can provide insight into a variety of cell cycle perturbations including those that occur in human cancer (60, 62) and ageing (63). We aim to create clusters of genes that are co-expressed, have common regulatory proteins and share a biological function. To achieve this, we use three datasets that were generated using different 'omics technologies and target different aspects of the molecular biology underpinning the cell cycle process.

- Microarray profiles of RNA expression from (64), comprising measurements of cell-cycle-regulated gene expression at 5-minute intervals for 200 minutes (up to three cell division cycles) and is referred to as the **time course** dataset. The cells are synchronised at the START checkpoint in late G1-phase using alpha factor arrest (64). We include only the genes identified by (64) as having periodic expression profiles.
- Chromatin immunoprecipitation followed by microarray hybridization (**ChIP-chip**) data from (65). This dataset discretizes p-values from tests of association between 117 DNA-binding transcriptional regulators and a set of yeast genes.

Based upon a significance threshold these p-values are represented as either a 0 (no interaction) or a 1 (an interaction).

- Protein-protein interaction (**PPI**) data from BioGrid (66). This database consists of of physical and genetic interactions between gene and gene products, with interactions either observed in high throughput experiments or computationally inferred. The dataset we used contained 603 proteins as columns. An entry of 1 in the $(i, j)^{th}$ cell indicates that the i^{th} gene has a protein product that is believed to interact with the j^{th} protein.

The datasets were reduced to the 551 genes with no missing data in the PPI and ChIP-chip data, as in (36).

Multiple dataset integration

We applied consensus clustering to MDI for our integrative analysis. Details of MDI are in section 2.2 of the Supplementary Material, but in short MDI jointly models the clustering in each dataset, inferring individual clusterings for each dataset. These partitions are informed by similar structure in the other datasets, with MDI learning this similarity as it models the partitions. The model does not assume global structure. This means that the similarity between datasets is not strongly assumed in our model; individual clusters or genes that align across datasets are based solely upon the evidence present in the data and not due to strong modelling assumptions. Thus, datasets that share less common information can be included without fearing that this will warp the final clusterings in some way.

The datasets were modelled using a mixture of Gaussian processes in the time course dataset and Multinomial distributions in the ChIP-chip and PPI datasets.

Results

Simulated data

We use the ARI between the generating labels and the inferred clustering of each method to be our metric of predictive performance. In figure 2, we see `Mclust`

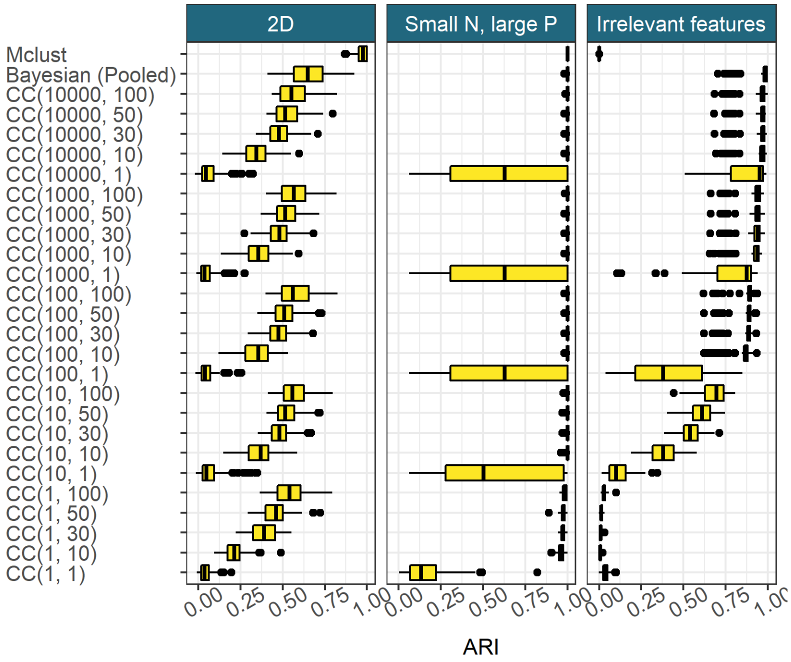

Figure 2 Model performance in the 100 simulated datasets for each scenario, defined as the ARI between the generating labels and the inferred clustering. $CC(d, w)$ denotes consensus clustering using the clustering from the d^{th} iteration from w different chains.

performs very well in the 2D and Small N, large P scenarios, correctly identifying the true structure. However, the irrelevant features scenario sees a collapse in performance, `Mclust` is blinded by the irrelevant features and identifies a clustering of $K = 1$.

The pooled samples from multiple long chains performs very well across all scenarios and appears to act as an upper bound on the more practical implementations of consensus clustering.

Consensus clustering does uncover some of the generating structure in the data, even using a small number of short chains. With sufficiently large ensembles and chain depth, consensus clustering is close to the pooled Bayesian samples in predictive performance. It appears that for a constant chain depth increasing the ensemble width used follows a pattern of diminishing returns. There are strong initial gains for a greater ensemble width, but the improvement decreases for each successive chain. A similar pattern emerges in increasing chain length for a constant number of chains (figure 2).

~~We see very little difference between the similarity matrix from the pooled samples and the consensus clustering (figure 3). Similar clusters emerge, and we see comparable confidence in the pairwise clusterings.~~ For the PSMs from the individual chains, all entries are 0 or ~~1.~~ 1 (figure 3). This means only a single clustering is sampled within each chain, implying very little uncertainty in the partition. However, three different ~~modes~~ clustering solutions emerge across the chains~~showing that the chains are~~, indicating that each individual chain is failing to explore the full support of the posterior distribution of the clustering~~and are each unrepresentative of the uncertainty in the final clustering. This shows that consensus clustering is exploring more possible clusterings than any individual chain and, as it explores a similar space to the pooled samples which might be considered more representative~~. In general, while MCMC convergence theorems hold as the number of iterations tend to infinity, any finite chain might suffer in representing the full support of the posterior distribution~~than any one chain, it suggests it better describes the true uncertainty present than any single chain~~, as we observe here. Moreover, the mixing of each chain can be poor as well (i.e. it may take a long time to reach the stationary distribution from an arbitrary initialisation). In our empirical study, we find that using many short runs provide similar point and interval estimates to running a small number of long chains (figure 3), while being computationally less expensive (figure 4), and hence more convenient for our applications. ~~It also shows that pooling chains offers robustness to multi-modality (as expected for an ensemble)and the ARI for the pooled samples is an upper bound on the performance for the individual long chains .~~

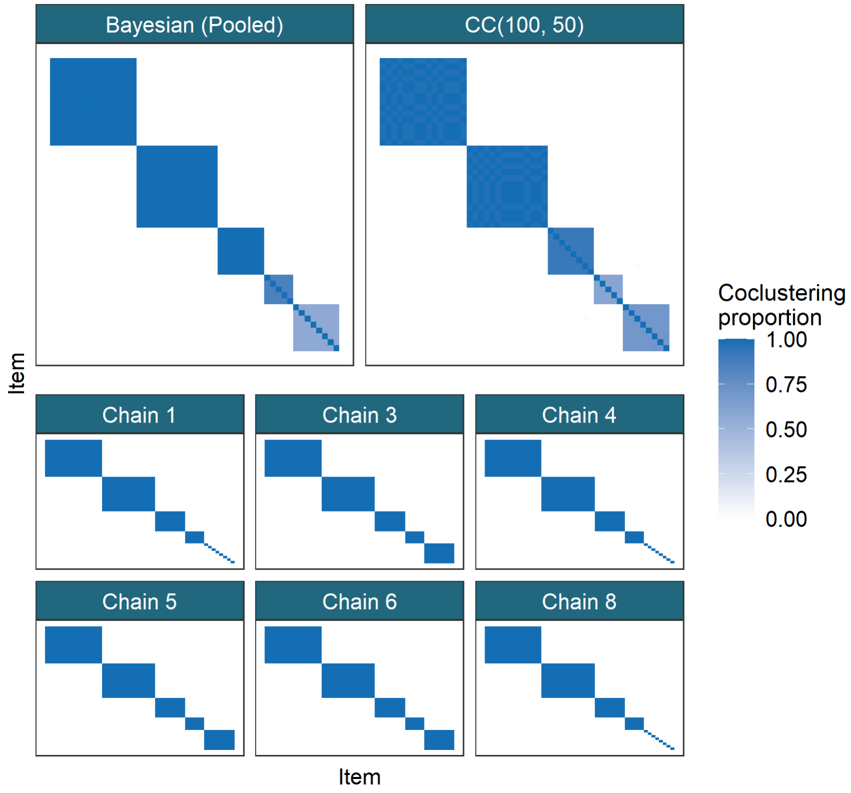

Figure 3 Comparison of similarity matrices from a dataset for the Small N, large P scenario. In each matrix, the $(i, j)^{th}$ entry is the proportion of clusterings for which the i^{th} and j^{th} items co-clustered for the method in question. In the first row the PSM of the pooled Bayesian samples is compared to the CM for CC(100, 50), with a common ordering of rows and columns in both heatmaps. In the following rows, 6 of the long chains that passed the tests of convergence are shown.

Figure 4 shows that chain length is directly proportional to the time taken for the chain to run. This means that using an ensemble of shorter chains, as in consensus clustering, can offer large reductions in the time cost of analysis when a parallel environment is available compared to standard Bayesian inference. Even on a laptop of 8 cores running an ensemble of 1,000 chains of length 1,000 will require approximately half as much time as running 10 chains of length 100,000 due to parallelisation, and the potential benefits are far greater when using a large computing cluster.

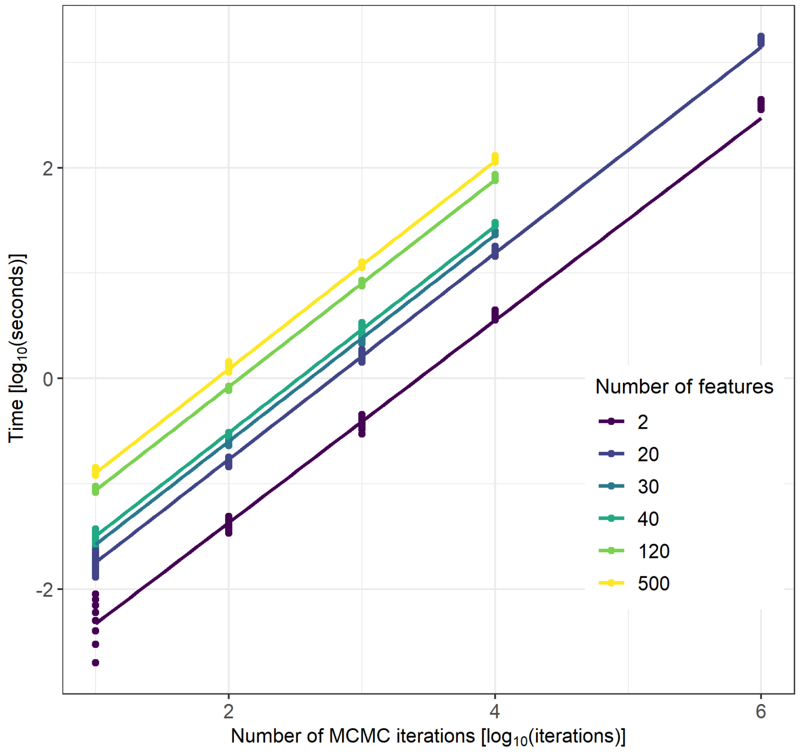

Figure 4 The time taken for different numbers of iterations of MCMC moves in $\log_{10}(seconds)$. The relationship between chain length, D, and the time taken is linear (the slope is approximately 1 on the $\log_{10}$ scale), with a change of intercept for different dimensions. The runtime of each Markov chain was recorded using the terminal command `time`, measured in milliseconds.

Additional results for these and other simulations are in section 4.4 of the Supplementary Material.

Multi-omics analysis of the cell cycle in budding yeast

We use the stopping rule proposed in to determine our ensemble depth and width. In figure 5, we see that the change in the consensus matrices from increasing the ensemble depth and width is diminishing in keeping with results in the simulations. We see no strong improvement after $D = 6,000$ and increasing the number of learners from 500 to 1,000 has small effect. We therefore use the largest ensemble available, a depth $D = 10001$ and width $W = 1000$, believing this ensemble is stable (additional evidence in section 5.1 of the Supplementary Material).

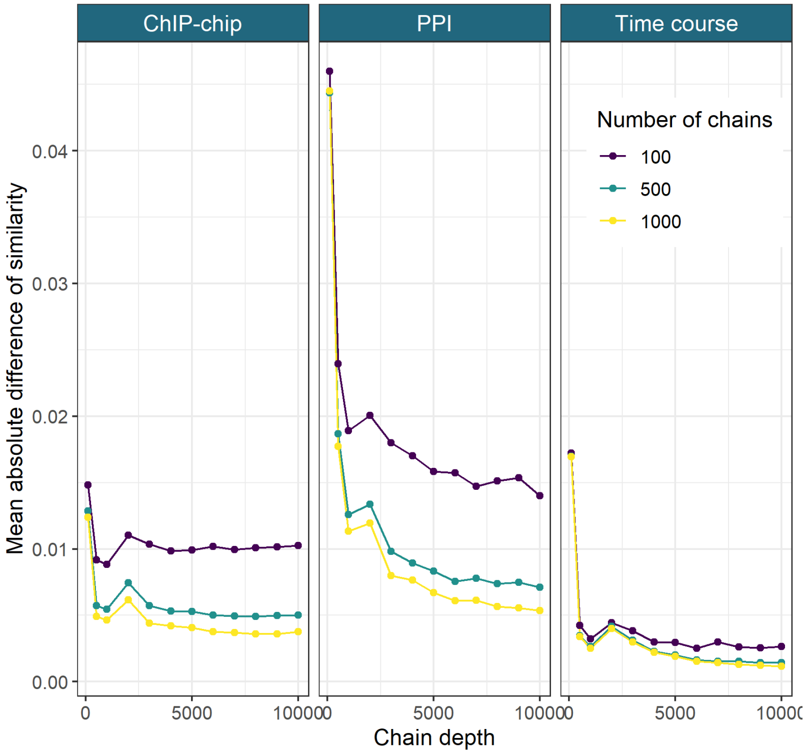

Figure 5 The mean absolute difference between the sequential Consensus matrices. For a set of chain lengths, $D' = \{d_1, \ldots, d_I\}$ and number of chains, $W' = \{w_1, \ldots, w_J\}$, we take the mean of the absolute difference between the consensus matrix for (d_i, w_j) and (d_{i-1}, w_j) (here $D' = \{101, 501, 1001, 2001, \ldots, 10001\}$ and $W' = \{100, 500, 1000\}$).

We focus upon the genes that tend to have the same cluster label across multiple datasets. More formally, we analyse the clustering structure among genes for which $\hat{P}(c_{nl} = c_{nm}) > 0.5$, where c_{nl} denotes the cluster label of gene n in dataset l. In our analysis it is the signal shared across the time course and ChIP-chip datasets that is strongest, with 261 genes (nearly half of the genes present) in this pairing tending to have a common label, whereas only 56 genes have a common label across all three datasets. Thus, we focus upon this pairing of datasets in the results of the analysis performed using all three datasets. We show the gene expression and regulatory proteins of these genes separated by their cluster in figure 6. In figure 6, the clusters in the time series data have tight, unique signatures (having different periods, amplitudes, or both) and in the ChIP-chip data clusters are defined by a small number of well-studied transcription factors (**TFs**) (see table 2 of the Supplementary Material for details of these TFs, many of which are well known to regulate cell cycle expression, 67).

As an example, we briefly analyse clusters 9 and 16 in greater depth. Cluster 9 has strong association with MBP1 and some interactions with SWI6, as can be seen in figure 6. The Mbp1-Swi6p complex, MBF, is associated with DNA replication (68). The first time point, 0 minutes, in the time course data is at the START checkpoint, or the G1/S transition. The members of cluster 9 begin highly expressed at this point before quickly dropping in expression (in the first of the 3 cell cycles). This suggests that many transcripts are produced immediately in advance of S-phase, and thus are required for the first stages of DNA synthesis. These genes' descriptions (found using `org.Sc.sgd.db`, 69, and shown in table 3 of the Supplementary Material) support this hypothesis, as many of the members are associated with DNA replication, repair and/or recombination. Additionally, *TOF1*, *MRC1* and *RAD53*, members of the replication checkpoint (70, 71) emerge in the cluster as do members of the cohesin complex. Cohesin is associated with sister chromatid cohesion which is established during the S-phase of the cell cycle (72) and also contributes to transcription regulation, DNA repair, chromosome condensation, homolog pairing (73), fitting the theme of cluster 9.

Cluster 16 appears to be a cluster of S-phase genes, consisting of *GAS3*, *NRM1* and *PDS1* and the genes encoding the histones H1, H2A, H2B, H3 and H4. Histones are the chief protein components of chromatin (74) and are important contributors to gene regulation (75). They are known to peak in expression in S-phase (64), which matches the first peak of this cluster early in the time series. Of the other members, *NRM1* is a transcriptional co-repressor of MBF-regulated gene expression acting at the transition from G1 to S-phase (76, 77). Pds1p binds to and inhibits the Esp1 class of sister separating proteins, preventing sister chromatids separation

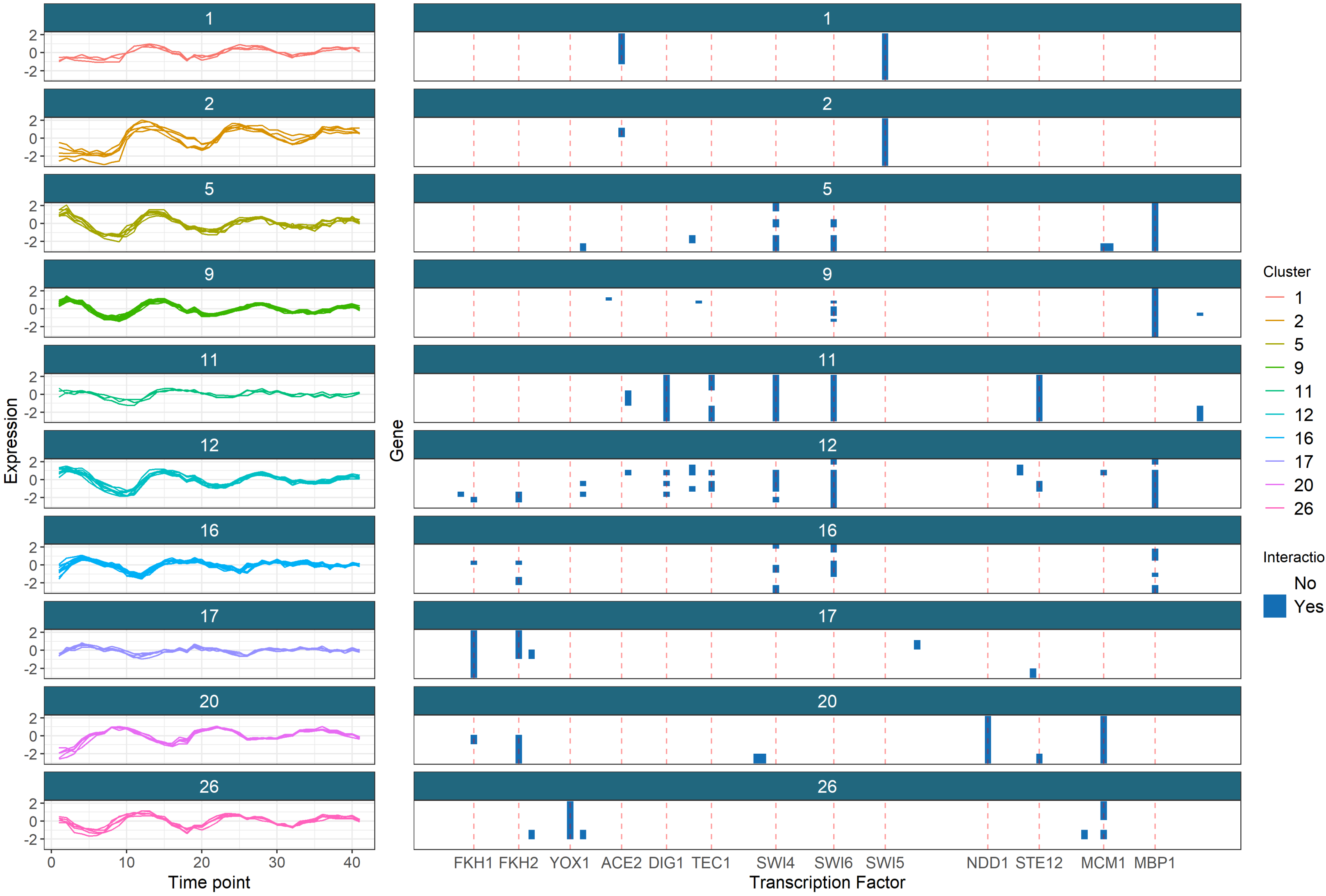

Figure 6 The gene clusters which tend to have a common label across the time course and ChIP-chip datasets, shown in these datasets. We include only the clusters with more than one member and more than half the members having some interactions in the ChIP-chip data. Red lines for the most common transcription factors are included.

before M-phase (72, 78). *GAS3*, is not well studied. It interacts with *SMT3* which regulates chromatid cohesion, chromosome segregation and DNA replication (among other things). Chromatid cohesion ensures the faithful segregation of chromosomes in mitosis and in both meiotic divisions (79) and is instantiated in S-phase (72). These results, along with the very similar expression profile to the histone genes in the time course data, suggest that *GAS3* may be more directly involved in DNA replication or chromatid cohesion than is currently believed.

We attempt to perform a similar analysis using traditional Bayesian inference of MDI, but after 36 hours of runtime there is no consistency or convergence across chains. We use the Geweke statistic and $\hat{R}$ to reduce to the five best behaved chains (none of which appear to be converged, see section 5.2 of the Supplementary Material for details). If we then compare the distribution of sampled values for the ϕ parameters for these long chains, the final ensemble used (D = 10001, W = 1000) and the pooled samples from the 5 long chains, then we see that the distribution of the pooled samples from the long chains (which might be believed to sampling different parts of the posterior distribution) is closer in appearance to the distributions sampled by the consensus clustering than to any single chain (figure 7). Further disagreement between chains is shown in the Gene Ontology term over-representation analysis in section 5.3 of the Supplementary Material.

Discussion

Our proposed method has demonstrated good performance on simulation studies, uncovering the generating structure in many cases and performing comparably to `Mclust` and long chains in many scenarios. We saw that when the chains are sufficiently deep that the ensemble approximates Bayesian inference, as shown by the similarity between the PSMs and the CM in the 2D scenario where the individual chains do not become trapped in a single mode. ~~However, we have shown that if a finite Markov chainfails to describe the full posterior distribution, our method~~

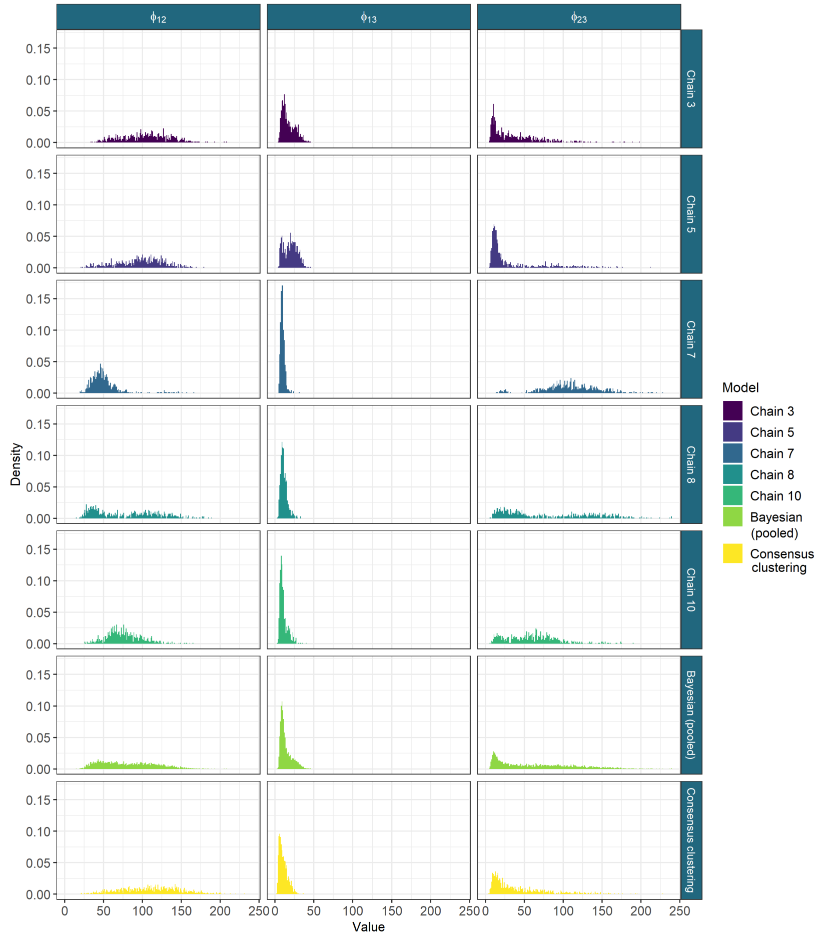

Figure 7 The sampled values for the ϕ parameters from the long chains, their pooled samples and the consensus using 1000 chains of depth 10,001. The long chains display a variety of behaviours. Across chains there is no clear consensus on the nature of the posterior distribution. The samples from any single chain are not particularly close to the behaviour of the pooled samples across all three parameters. It is the consensus clustering that most approaches this pooled behaviour.

~~frequently has better ability to represent several modes in the data than individual chains and thus offers a more consistent and reproducible analysis.~~ We have shown cases where many short runs are computationally less expensive than one long chain and give meaningful point and interval estimates; estimates that are very similar to those from the limiting case of a Markov chain. Thus if individual chains are suffering from mixing problems or are too computationally expensive to run, consensus clustering may provide a viable option. We also showed that the ensemble of short chains is more robust to irrelevant features than Mclust. ~~Furthermore, an ensemble of short chains is significantly faster in a parallel environment than inference using individual long chains.~~

We proposed a method of assessing ensemble stability and deciding upon ensemble size which we used when performing an integrative analysis of yeast cell cycle data using MDI, an extension of Bayesian mixture models that jointly models multiple datasets. We uncovered many genes with shared signal across several datasets and explored the meaning of some of the inferred clusters using data external to the analysis. We found biologically meaningful results as well as signal for possibly novel biology. We also showed that individual chains for the existing implementation of MDI do not converge in a practical length of time, having run 10 chains for 36 hours with no consistent behaviour across chains. This means that Bayesian inference of the MDI model is not practical on this dataset with the software currently available.

However, consensus clustering does lose the theoretical framework of true Bayesian inference. We attempt to mitigate this with our assessment of stability in the ensemble, but this diagnosis is heuristic and subjective, and while there is empirical evidence for its success, it lacks the formal results for the tests of model convergence for Bayesian inference.

More generally, we have benchmarked the use of an ensemble of Bayesian mixture models, showing that this approach can infer meaningful clusterings and overcomes the problem of multi-modality in the likelihood surface even in high dimensions, thereby providing more stable clusterings than individual long chains that are prone to becoming trapped in individual modes. We also show that the ensemble can be significantly quicker to run. In our multi-omics study we have demonstrated that the method can be applied as a wrapper to more complex Bayesian clustering methods using existing implementations and that this provides meaningful results even when individual chains fail to converge. This enables greater application of complex Bayesian clustering methods without requiring re-implementation using more clever MCMC methods, a process that would involve a significant investment of human time.

We expect that researchers interested in applying some of the Bayesian integrative clustering models such as MDI and Clusternomics (37) will be enabled to do so, as consensus clustering overcomes some of the unwieldiness of existing implementations of these complex models. More generally, we expect that our method will be useful to researchers performing cluster analysis of high-dimensional data where the runtime of MCMC methods becomes too onerous and multi-modality is more likely to be present.

Abbreviations

ARI: Adjusted Rand Index
ChIP-chip: Chromatin immunoprecipitation followed by microarray hybridization
CM: Consensus Matrix
MCMC: Markov chain Monte Carlo
MDI: Multiple Dataset Integration
PCA: Principal Component Analysis
PPI: Protein-Protein Interaction
PSM: Posterior Similarity Matrix
SSE: Sum of Squared Errors
TF: Transcription Factor

Declarations

Not applicable.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

The code and datasets supporting the conclusions of this article are available in the github repository, `https://github.com/stcolema/ConsensusClusteringForBayesianMixtureModels`.

Competing interests

The authors declare that they have no competing interests.

Funding

This work was funded by the MRC (MC UU 00002/4, MC UU 00002/13) and supported by the NIHR Cambridge Biomedical Research Centre (BRC-1215-20014). The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health and Social Care. This research was funded in whole, or in part, by the Wellcome Trust [WT107881]. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

Authors' contributions

SC designed the simulation study with contributions from PK and CW, performed the analyses and wrote the manuscript. PK and CW provided an equal contribution of joint supervision, directing the research and provided suggestions such as the stopping rule. All contributed to interpreting the results of the analyses. All authors revised and approved the final manuscript.

Acknowledgements

Not applicable.

Author details

[1]MRC Biostatistics Unit University of Cambridge, Cambridge, UK. [2]Cambridge Institute of Therapeutic Immunology & Infectious Disease, University of Cambridge, Cambridge, UK.

References

1. Hejblum BP, Skinner J, Thiébaut R. Time-course gene set analysis for longitudinal gene expression data. PLoS computational biology. 2015;11(6):e1004310.
2. Bai JP, Alekseyenko AV, Statnikov A, Wang IM, Wong PH. Strategic applications of gene expression: from drug discovery/development to bedside. The AAPS journal. 2013;15(2):427–437.
3. Emmert-Streib F, Dehmer M, Haibe-Kains B. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. Frontiers in cell and developmental biology. 2014;2:38.
4. Lloyd S. Least squares quantization in PCM. IEEE transactions on information theory. 1982;28(2):129–137.
5. Forgy EW. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. biometrics. 1965;21:768–769.
6. Rousseeuw PJ. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computational and Applied Mathematics. 1987 Nov;20:53–65.
7. Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding. Stanford; 2006.
8. Breiman L. Random forests. Machine learning. 2001;45(1):5–32.
9. Friedman JH. Stochastic gradient boosting. Computational statistics & data analysis. 2002;38(4):367–378.
10. Monti S, Tamayo P, Mesirov J, Golub T. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Machine learning. 2003;52(1-2):91–118.
11. Wilkerson, D M, Hayes, Neil D. ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. Bioinformatics. 2010;26(12):1572–1573.
12. John CR, Watson D, Russ D, Goldmann K, Ehrenstein M, Pitzalis C, et al. M3C: Monte Carlo reference-based consensus clustering. Scientific reports. 2020;10(1):1–14.
13. Gu Z, Schlesner M, Hübschmann D. cola: an R/Bioconductor package for consensus partitioning through a general framework. Nucleic Acids Research. 2020 12;Gkaa1146. Available from: https://doi.org/10.1093/nar/gkaa1146.

14. Lehmann BD, Bauer JA, Chen X, Sanders ME, Chakravarthy AB, Shyr Y, et al. Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies. The Journal of clinical investigation. 2011;121(7):2750–2767.

15. Verhaak RG, Hoadley KA, Purdom E, Wang V, Qi Y, Wilkerson MD, et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. Cancer cell. 2010;17(1):98–110.

16. Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, et al. SC3: consensus clustering of single-cell RNA-seq data. Nature methods. 2017;14(5):483–486.

17. Li T, Ding C. Weighted Consensus Clustering. In: Proceedings of the 2008 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics; 2008. p. 798–809.

18. Carpineto C, Romano G. Consensus Clustering Based on a New Probabilistic Rand Index with Application to Subtopic Retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012 Dec;34(12):2315–2326.

19. Strehl A, Ghosh J. Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. Journal of Machine Learning Research. 2002;3:583–617.

20. Ghaemi R, Sulaiman MN, Ibrahim H, Mustapha N, et al. A survey: clustering ensembles techniques. World Academy of Science, Engineering and Technology. 2009;50:636–645.

21. Ünlü R, Xanthopoulos P. Estimating the Number of Clusters in a Dataset via Consensus Clustering. Expert Systems with Applications. 2019 Jul;125:33–39.

22. Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. Journal of the American statistical Association. 2002;97(458):611–631.

23. Fraley C. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. The Computer Journal. 1998 Aug;41(8):578–588.

24. Ferguson TS. A Bayesian analysis of some nonparametric problems. The annals of statistics. 1973;p. 209–230.

25. Antoniak CE. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. The Annals of Statistics. 1974 Nov;2(6):1152–1174.

26. Ferguson TS. Bayesian Density Estimation by Mixtures of Normal Distributions. In: Rizvi MH, Rustagi JS, Siegmund D, editors. Recent Advances in Statistics. Academic Press; 1983. p. 287–302.

27. Lo AY. On a Class of Bayesian Nonparametric Estimates: I. Density Estimates. The Annals of Statistics. 1984;12(1):351–357.

28. Richardson S, Green PJ. On Bayesian analysis of mixtures with an unknown number of components. Journal of the Royal Statistical Society: series B. 1997;59(4):731–792.

29. Miller JW, Harrison MT. Mixture models with a prior on the number of components. Journal of the American Statistical Association. 2018;113(521):340–356.

30. Rousseau J, Mengersen K. Asymptotic behaviour of the posterior distribution in overfitted mixture models. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2011;73(5):689–710.

31. Medvedovic M, Sivaganesan S. Bayesian infinite mixture model based clustering of gene expression profiles. Bioinformatics. 2002;18(9):1194–1206.

32. Chan C, Feng F, Ottinger J, Foster D, West M, Kepler TB. Statistical mixture modeling for cell subtype identification in flow cytometry. Cytometry Part A: The Journal of the International Society for Analytical Cytology. 2008;73(8):693–701.

33. Hejblum BP, Alkhassim C, Gottardo R, Caron F, Thiébaut R, et al. Sequential Dirichlet process mixtures of multivariate skew t-distributions for model-based clustering of flow cytometry data. The Annals of Applied Statistics. 2019;13(1):638–660.
34. Prabhakaran S, Azizi E, Carr A, Pe'er D. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. In: International Conference on Machine Learning; 2016. p. 1070–1079.
35. Crook OM, Mulvey CM, Kirk PD, Lilley KS, Gatto L. A Bayesian mixture modelling approach for spatial proteomics. PLoS computational biology. 2018;14(11):e1006516.
36. Kirk P, Griffin JE, Savage RS, Ghahramani Z, Wild DL. Bayesian correlated clustering to integrate multiple datasets. Bioinformatics. 2012;28(24):3290–3297.
37. Gabasova E, Reid J, Wernisch L. Clusternomics: Integrative context-dependent clustering for heterogeneous datasets. PLoS computational biology. 2017;13(10):e1005781.
38. Strauss ME, Kirk PD, Reid JE, Wernisch L. GPseudoClust: deconvolution of shared pseudo-profiles at single-cell resolution. Bioinformatics. 2020;36(5):1484–1491.
39. Robert CP, Elvira V, Tawn N, Wu C. Accelerating MCMC algorithms. Wiley Interdisciplinary Reviews: Computational Statistics. 2018;10(5):e1435.
40. Neiswanger W, Wang C, Xing EP. Asymptotically Exact, Embarrassingly Parallel MCMC. In: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence. UAI'14. Arlington, Virginia, USA: AUAI Press; 2014. p. 623–632.
41. Murray L. Distributed Markov Chain Monte Carlo. In: Proceedings of Neural Information Processing Systems workshop on learning on cores, clusters and clouds. vol. 11; 2010. .
42. Jacob PE, O'Leary J, Atchadé YF. Unbiased Markov Chain Monte Carlo Methods with Couplings. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2020;82(3):543–600.
43. Wold S, Esbensen K, Geladi P. Principal component analysis. Chemometrics and intelligent laboratory systems. 1987;2(1-3):37–52.
44. Fritsch A, Ickstadt K, et al. Improved criteria for clustering based on the posterior similarity matrix. Bayesian analysis. 2009;4(2):367–391.
45. Fritsch A. mcclust: process an MCMC sample of clusterings; 2012. R package version 1.0. Available from: https://CRAN.R-project.org/package=mcclust.
46. Wade S, Ghahramani Z. Bayesian Cluster Analysis: Point Estimation and Credible Balls (with Discussion). Bayesian Analysis. 2018 Jun;13(2):559–626.
47. Lourenço A, Rota Bulò S, Rebagliati N, Fred ALN, Figueiredo MAT, Pelillo M. Probabilistic Consensus Clustering Using Evidence Accumulation. Machine Learning. 2015 Jan;98(1):331–357.
48. Dahl DB, Johnson DJ, Mueller P. Search Algorithms and Loss Functions for Bayesian Clustering. arXiv:210504451 [stat]. 2021 May;.
49. Von Luxburg U, Ben-David S. Towards a statistical theory of clustering. In: Pascal workshop on statistics and optimization of clustering. Citeseer; 2005. p. 20–26.
50. Meinshausen N, Bühlmann P. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2010;72(4):417–473.
51. Law MH, Jain AK, Figueiredo M. Feature selection in mixture-based clustering. In: Advances in neural information processing systems; 2003. p. 641–648.
52. Hubert L, Arabie P. Comparing partitions. Journal of classification. 1985;2(1):193–218.
53. Scrucca L, Fop M, Murphy BT, Raftery AE. mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. The R Journal. 2016;8(1):289–317. Available from:

https://doi.org/10.32614/RJ-2016-021.

54. Schwarz G, et al. Estimating the dimension of a model. The annals of statistics. 1978;6(2):461–464.

55. Geweke J, et al. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. vol. 196. Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN; 1991.

56. Gelman A, Rubin DB, et al. Inference from iterative simulation using multiple sequences. Statistical science. 1992;7(4):457–472.

57. Vats D, Knudson C. Revisiting the Gelman-Rubin diagnostic. arXiv preprint arXiv:181209384. 2018;.

58. Shapiro SS, Wilk MB. An analysis of variance test for normality (complete samples). Biometrika. 1965;52(3/4):591–611.

59. Tyson JJ, Chen KC, Novák B. Cell Cycle, Budding Yeast. In: Dubitzky W, Wolkenhauer O, Cho KH, Yokota H, editors. Encyclopedia of Systems Biology. New York, NY: Springer New York; 2013. p. 337–341.

60. Chen KC, Calzone L, Csikasz-Nagy A, Cross FR, Novak B, Tyson JJ. Integrative analysis of cell cycle control in budding yeast. Molecular biology of the cell. 2004;15(8):3841–3862.

61. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P. The cell cycle and programmed cell death. Molecular biology of the cell. 2002;4:983–1027.

62. Ingalls B, Duncker B, Kim D, McConkey B. Systems level modeling of the cell cycle using budding yeast. Cancer informatics. 2007;3:117693510700300020.

63. Jiménez J, Bru S, Ribeiro M, Clotet J. Live fast, die soon: cell cycle progression and lifespan in yeast cells. Microbial Cell. 2015;2(3):62.

64. Granovskaia MV, Jensen LJ, Ritchie ME, Toedling J, Ning Y, Bork P, et al. High-resolution transcription atlas of the mitotic cell cycle in budding yeast. Genome biology. 2010;11(3):1–11.

65. Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, Danford TW, et al. Transcriptional regulatory code of a eukaryotic genome. Nature. 2004;431(7004):99–104.

66. Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M. BioGRID: a general repository for interaction datasets. Nucleic acids research. 2006;34(suppl_1):D535–D539.

67. Simon I, Barnett J, Hannett N, Harbison CT, Rinaldi NJ, Volkert TL, et al. Serial regulation of transcriptional regulators in the yeast cell cycle. Cell. 2001;106(6):697–708.

68. Iyer VR, Horak CE, Scafe CS, Botstein D, Snyder M, Brown PO. Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. Nature. 2001;409(6819):533–538.

69. Carlson M, Falcon S, Pages H, Li N. Org. sc. sgd. db: Genome wide annotation for yeast. R package version. 2014;2(1).

70. Bando M, Katou Y, Komata M, Tanaka H, Itoh T, Sutani T, et al. Csm3, Tof1, and Mrc1 form a heterotrimeric mediator complex that associates with DNA replication forks. Journal of Biological Chemistry. 2009;284(49):34355–34365.

71. Lao JP, Ulrich KM, Johnson JR, Newton BW, Vashisht AA, Wohlschlegel JA, et al. The yeast DNA damage checkpoint kinase Rad53 targets the exoribonuclease, Xrn1. G3: Genes, Genomes, Genetics. 2018;8(12):3931–3944.

72. Tóth A, Ciosk R, Uhlmann F, Galova M, Schleiffer A, Nasmyth K. Yeast cohesin complex requires a conserved protein, Eco1p (Ctf7), to establish cohesion between sister chromatids during DNA replication. Genes & development. 1999;13(3):320–333.

73. Mehta GD, Kumar R, Srivastava S, Ghosh SK. Cohesin: functions beyond sister chromatid cohesion. FEBS letters. 2013;587(15):2299–2312.

74. Fischle W, Wang Y, Allis CD. Histone and chromatin cross-talk. Current opinion in cell biology. 2003;15(2):172–183.

75. Bannister AJ, Kouzarides T. Regulation of chromatin by histone modifications. Cell research. 2011;21(3):381–395.

76. de Bruin RA, Kalashnikova TI, Chahwan C, McDonald WH, Wohlschlegel J, Yates III J, et al. Constraining G1-specific transcription to late G1 phase: the MBF-associated corepressor Nrm1 acts via negative feedback. Molecular cell. 2006;23(4):483–496.

77. Aligianni S, Lackner DH, Klier S, Rustici G, Wilhelm BT, Marguerat S, et al. The fission yeast homeodomain protein Yox1p binds to MBF and confines MBF-dependent cell-cycle transcription to G1-S via negative feedback. PLoS Genet. 2009;5(8):e1000626.

78. Ciosk R, Zachariae W, Michaelis C, Shevchenko A, Mann M, Nasmyth K. An ESP1/PDS1 complex regulates loss of sister chromatid cohesion at the metaphase to anaphase transition in yeast. Cell. 1998;93(6):1067–1076.

79. Cooper KF, Mallory MJ, Guacci V, Lowe K, Strich R. Pds1p is required for meiotic recombination and prophase I progression in Saccharomyces cerevisiae. Genetics. 2009;181(1):65–79.

Additional Files

Additional file 1 — Supplementary materials

Additional relevant theory, background and results. This includes some more formal definitions, details of Bayesian mixture models and MDI, the general consensus clustering algorithm, additional simulations and the generating algorithm used, steps in assessing Bayesian model convergence in both the simulated datasets and yeast analysis, a table of the transcription factors that define the clustering in the ChIP-chip dataset, a table of the gene descriptions for some of the clusters that emerge across the time course and ChIP-chip datasets and Gene Ontology term over-representation analysis of the clusterings from the yeast datasets. (PDF, 10MB)