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: Ensembles have been highly successful in many applications. They frequently describe multiple modes in the likelihood surface better than any individual learner and also offer computational gains due to independence between learners. Bayesian mixture models are powerful tools that enable inference of the number of clusters present but suffer from problems associated with Markov-chain monte Carlo methods.

Results: We apply the clustering ensemble method, Consensus clustering, to Bayesian mixture models. We investigate the performance of this approach in simulations, comparing it to Bayesian inference of the same models and `Mclust`, a popular implementation of MLE inference of mixture models in R. Consensus clustering approximates Bayesian inference when the ensemble is sufficiently large and each learner is sufficiently deep, successfully capturing multiple modes and offering significant reductions in runtime when a parallel environment is available. We propose a heuristic to deciding upon the ensemble size and then apply Consensus clustering to Multiple Dataset Integration, an extension of Bayesian mixture models for integrative analyses, on three 'omics datasets for the cell-cycle of budding yeast and find biologically meaningful results, showing that Consensus clustering can also be applied to more complex extensions of mixture models.

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 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 and their interactions are more interpretable than their constituent members (Hejblum *et al.*, 2015) with possible applications for diagnosis and drug targets (Bai *et al.*, 2013; Emmert-Streib *et al.*, 2014).

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. These methods are often heuristic in nature, relying on rules of thumb to decide upon a final model choice. For example, different choices of K are compared under some metric such as silhouette or based upon the within-cluster sum of squared errors (ψ) as a function of K. For k-means clustering, its sensitivity to initialisation means multiple runs are often used in practice, with that which minimises ψ used (Arthur and Vassilvitskii, 2006). This problem arises as the algorithm has no guarantees on finding the global minimum of ψ.

In many analyses or decision making processes, quantifying confidence in the clustering can be of interest. Returning to the stratified medicine example of clustering patients, there might be individuals with almost equal probability of being allocated between a number of clusters which might influence decisions made. However in many cluster analyses only a point clustering is estimated and thus one is no wiser about which individuals are boundary members of clusters.

One solution to some of the problems prevalent in cluster analysis, i.e. sensitivity to initialisation, no measure of uncertainty, is through the use of *ensembles* of models. This approach has had great success in supervised learnings, most famously in the form of Random Forest (Breiman, 2001) and boosting (Friedman, 2002). In clustering, Consensus clustering (Monti *et al.*, 2003) is a popular method which has been implemented in R (Wilkerson *et al.*, 2010) and been applied to problems such as cancer subtyping (Lehmann *et al.*, 2011; Verhaak *et al.*, 2010) and identifying subclones in single cell analysis (Kiselev *et al.*, 2017). Consensus clustering uses W runs of some base model or learner (such as k-means clustering) and compiles the W proposed partitions 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 cocluster. This proportion represents some measure of confidence in the coclustering of any pair of items. Furthermore, ensembles can offer reductions in computational runtime because the learners in most ensemble methods are independent of each other and thus enable use of a parallel environment (Ghaemi *et al.*, 2009).

Monti *et al.* (2003) proposed some methods to choosing K using the Consensus matrix, but this remains a problem in the methods mentioned so far. An alternative clustering framework, model-based clustering or *mixture models*, embeds the cluster analysis within a formal, statistical framework (Fraley and Raftery, 2002). This means that models can be compared formally, and problems such as the choice of K is a model selection problem with all the associated tools.

Furthermore, *Bayesian mixture models* can treat K as a random variable that is inferred from the data and thus the final clustering is not conditional upon a user chosen value, but K is jointly modelled along with the clustering. 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 (Medvedovic and Sivaganesan, 2002), cell types in flow cytometry (Chan *et al.*, 2008; Hejblum *et al.*, 2019) or scRNAseq experiments (Prabhakaran *et al.*, 2016), and estimating protein localisation (Crook *et al.*, 2018).

While Bayesian methods are very powerful, their reliance upon Markov chain Monte Carlo (MCMC) methods presents problems in practice. Most prevalent is difficulty in exploring the posterior distribution (normally becoming trapped in a single mode) and slow runtimes (for a description of these and other problems, please see Robert *et al.*, 2018; Yao *et al.*, 2020; Chandra *et al.*, 2020). We believe that Bayesian methods are severely under-utilised in the ensemble framework and propose applying Consensus clustering to Bayesian mixture models to overcome some of the issues endemic in high dimensional Bayesian clustering. Monti *et al.* (2003) actually propose this as part of their original paper, but no investigation of this has been attempted to date. This ensemble approach sidesteps the problems of convergence associated MCMC methods and offers computational gains through shorter chains run in parallel.

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 relatively 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. We also show that the ensemble can describe multiple modes in scenarios where any individual chain becomes trapped, and thus the uncertainty present in the Consensus matrix can be more representative of the data.

Based upon our simulations we propose a heuristic for deciding upon the ensemble width (the number of learners used, S) and the ensemble depth (the number of iterations run within each chain, R).

We then perform an integrative analysis of 'omics data relating to the cell cycle of *Saccharomyces cerevisiae*. We apply Consensus clustering to an extension of Bayesian mixture models, Multiple Dataset Integration (MDI), a multiple dataset clustering method. We determine the ensemble size using our proposed stopping rule and find meaningful clusters of genes.

```
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. Monti *et al.* (2003) suggest rules for both choosing the best $K \in \mathcal{K}$ and for inferring a point estimate clustering from $\mathbf{M}^{(\hat{K})}$. $\mathbf{B}^{(s)}$ is the co-clustering matrix for the s^{th} model run.

2 Methods

2.1 Consensus clustering for Bayesian mixture models

We apply Consensus clustering to Bayesian mixture models. 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. The MCMC method driving each model offers diversity of partitions when combined with different initialisations. The method is described in algorithm 2.

Our application of Consensus clustering has two main parameters, the model depth, D, and width, W. In several figures we use a shorthand of $CC(d, w)$ to refer to Consensus clustering using the clustering from the d^{th} iteration from w different chains.

2.2 Bayesian inference

We use 10 long chains in where "Bayesian inference" is being considered in the simulations and the yeast analysis. In the simulations chains are kept in the analysis based upon a number of tests described below. To summarise the chains that are considered "converged" we pool the samples from across the chains (effectively using an ensemble of long chains). This choice is considered more in the supplementary materials. These pooled samples are expected to be an upper bound on the performance of any single chain in the simulations.

2.3 `Mclust`

`Mclust` (Scrucca *et al.*, 2016) is a function that compares mixture models based upon the maxmimum likelihood estimator of the parameters for a range of choices of K, the number of clusters used, and different covariance structures. The method initialises upon a hierarchical clustering of the

```
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.

data cut to K clusters. The "best" model is determined using the Bayesian information criterion, (**BIC**, Schwarz *et al.*, 1978).

2.4 Bayesian mixture model implementation

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

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

We use the implementation of Bayesian mixture models provided by Mason *et al.* (2016). Rather than directly using a Dirichlet process to infer the number of clusters (for example by using Reversible Jump MCMC as described by Richardson and Green, 1997), this implementation follows the logic of Van Havre *et al.* (2015) and uses an overfitted mixture model to approximate a Dirichlet process. We set the total number of occupied and empty components, K_{max}, to 50 in the simulations and 275 in the yeast analysis. A component is a density in the mixture model; if a component is occupied than it forms a cluster and contributes to partitioning the sample.

2.5 Predicting a clustering from Consensus matrices or PSMs

We use the `maxpear` function (Fritsch *et al.*, 2009) from the R package `mcclust` (Fritsch, 2012) to infer a point clustering from Consensus matrices. 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 Consensus matrix, except the partitions are all drawn from a single Markov chain), predicting a clustering that has maximum posterior expected adjusted Rand index (**ARI**, Hubert and Arabie, 1985) with the true clustering. In the context of the Consensus matrix, the function does not have this interpretation. However, the method produces a kind of "average clustering" based upon all sampled partitions, effectively averaging over each learner in the ensemble, which we feel is a sensible point estimate.

2.6 Data generating mechanism

We use a finite mixture model with independent features as the data generating model in the simulation study. Within this model there exist "irrelevant features" (Law *et al.*, 2003) that have global parameters rather than cluster specific parameters. The generating model is

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

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

2.7 Performance quantification

We use the ARI as our metric for the quality of the point clustering inferred by each method, comparing this estimate with the generating labels.

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

2.8 Bayesian model convergence

To assess within-chain convergence of our Bayesian inference we use the Geweke Z-score 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 Z-score statistic is expected to be normally distributed. Normality is tested for using a Shapiro-Wilks test (Shapiro and Wilk, 1965). If a chain fails this test (i.e. the associated p-value is less than 0.05), we assume that is has not achieved stationarity and is excluded from the remainder of the analysis.

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. A deeper consideration of the convergence of chains is included in the Supplementary Materials.

2.9 Stopping rule for ensemble growth

As our ensemble sidesteps the problem of convergence within each chain, we need an alternative stopping rule for growing the ensemble in chain depth, R, and number of chains, S. We propose a heuristic based upon the Consensus matrix to decide if a given value of R and S are sufficient. We suspect that increasing S and R might continuously improve the performance of the ensemble, but we observe in our simulations that these improvements will become smaller and smaller for greater values, approaching some asymptote for each of S and R.

Following this logic if the Consensus matrices for three ensembles define by the parameters (aR, S), (R, bS) and (R, S) are not visibly different for some reasonable values of a, b, S and R than increasing ensemble size or depth will see at most marginal improvement in performance.

However, comparing an array of Consensus matrices by eye is both difficult and timely. Inspired by the logic of using a scree plot in Principal Component Analysis (**PCA**) to decide the number of components to keep we use a plot of the mean absolute difference between sequential Consensus matrices as a tool in the decision making progress. PCA is analogous to Consensus clustering in that where for Consensus clustering some improvement might always be expected for increasing chain length or number, more variance will always be captured by increasing the number of components used. However, increasing this number beyond a certain point has diminishing returns. We recommend a plot of the mean absolute difference between the sequential Consensus matrices for a set of values of $R' = \{r_1, \dots, r_I\}$ and $S' = \{s_1, \dots, s_J\}$, comparing the Consenus matrix for the r_i^{th} iteration from s_j chains to that for the $r^t_{(i-1)}h$ iteration from s_j chains. If the values are no longer dropping sharply (i.e. the partial derivative of the mean absolute difference with respect to R and S is as small as it is likely to become) than the change in the analysis for greater values of R or S is probably marginal.

If this heuristic is used, we believe that the Consensus matrix and the resulting inference should be stable, providing consistent estimate of the clustering. In contrast, if there is still strong variation in the Consensus matrix for varying chain length or number, than we believe that the inferred clustering is influenced significantly by the random initialisation. This means that the inferred partition that it is unlikely to be stable for similar datasets or reproducible for a random choice of seeds. This stability is often a desirous property in a clustering method (Von Luxburg and Ben-David, 2005; Meinshausen and Bühlmann, 2010).

3 Examples

3.1 Simulations

The methods used are

- `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 $R \in \{1, 10, 100, 1000, 10000\}$.

These are compared within a range of 12 scenarios, of which 3 are shown here (please see the supplementary materials for additional results). These are described in table 2 and test different settings represent of specific characteristics of real datasets.

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
2D	100	2	0	5	3.0	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	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Small N, large P	50	500	0	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$

Table 2. Parameters defining the simulation scenarios as used in generating data and labels.

The examples included are

- a low-dimensional dataset,

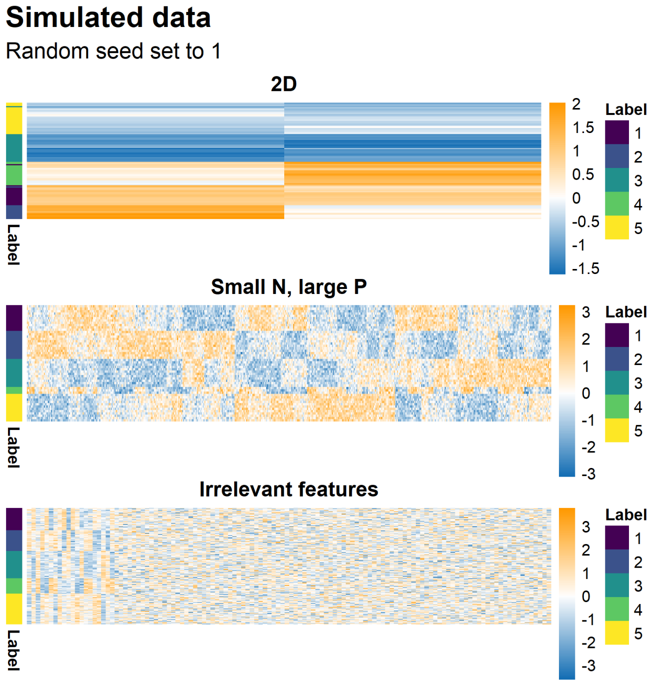

Fig. 1. Example of generated datasets. The low-dimensional dataset (which is ordered by hierarchical clustering here) should enable proper mixing of chains in the MCMC. The small N, large P case has clear structure (observable by eye). This is intended to highlight the problems of poor mixing due to 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 lefthand side of this heatmap). This setting is intended to test how sensitive each approach is to noise.

- 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, with no pathological behaviour emerging. 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. More details of both the process of generating the data and the analysis are available in the Supplementary Material.

In theory we would expect the Bayesian chains to explore a common posterior distribution, but the practice sees chains become trapped in distinct modes in different scenarios. We believe that the mode within which the greatest number of chains become trapped would be that which is used to perform the inference in a real application (and longer chains did not solve the problem). As part of a pipeline where such analyses have to happen $12 \times 100 = 1,200$ times, we pool the Bayesian samples into a single PSM to aovid model selection problems. If the chains all explore the same space pooling the samples has little effect on the inference. Based upon visual inspection of the PSMs indicates that the disagreement between modes tends to be one of

- several clusters are merged or
- a cluster is represented by two or more components,

each of the modes tends to have large overlap with all others. This means that the clustering inferred from the PSM created from the samples pooled across all stationary chains will represent the most popular mode and the ARI will not be unduly inflated.

3.2 Yeast data

We perform an integrative analysis of cell cycle data from *Saccharomyces cerevisiae*. The cell cycle is the process by which a growing cell divides into two daughter cells. This involves virtually all cellular processes - metabolism, protein synthesis, secretion, DNA replication, organelle biogenesis, cytoskeletal dynamics and chromosome segregation - and diverse regulatory events (Granovskaia *et al.*, 2010). The cell cycle is crucial to biological growth, repair, reproduction, and development; it is fundamental to sustaining life (Tyson *et al.*, 2013; Chen *et al.*, 2004; Alberts *et al.*, 2018). The regulatory proteins of the system first appeared over a billion years ago and are so highly conserved among eukaryotes that many of them function perfectly when transferred from a human cell to a yeast cell (Alberts *et al.*, 2018). This conservation means that a relatively simple eukaryote such as *Saccharomyces cerevisiae* can provide insight into a variety of cell cycle perturbations including those that occur in human cancer (Ingalls *et al.*, 2007; Chen *et al.*, 2004) and ageing (Jiménez *et al.*, 2015). Budding yeast is particularly attractive for genetic analysis as it can proliferate as haploid cells, its genetic makeup can be easily altered by standard tools of molecular genetics, and large numbers of cells may be synchronised in a particular stage of the cell cycle (Tyson *et al.*, 2013; Juanes, 2017).

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 data 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 Granovskaia *et al.* (2010). This dataset comprises measurements of cell-cycle-regulated expression at 5-minute intervals for 41 time points (up to three cell division cycles) and is referred to as the **Timecourse** dataset. The cells are synchronised at the START checkpoint in late G1-phase using alpha factor arrest (Granovskaia *et al.*, 2010). We include only the genes identified by Granovskaia *et al.* (2010) as having periodic expression profiles. This includes some non-coding RNAs (**ncRNAs**) of which the majority are anti-sense RNAs.
- Chromatin immunoprecipitation followed by microarray hybridization (**ChIP-chip**) data from Harbison *et al.* (2004). 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 (Stark *et al.*, 2006). This database consists of of physical and genetic interactions between gene and gene products. The interactions included are a collection of results observed in high throughput experiments and some computationally inferred interactions. 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.

We believe that the integrative aspect of the analysis means that the clusters are more interpretable than in a standalone cluster analysis. Cluster analysis of a single dataset entails interpreting the clusters defined by similarity within a single experiment which often involves strong assumptions about the biological processes behind the result (e.g. correlation of transcripts implies co-regulation).

We applied Consensus clustering to MDI for our integrative analysis. MDI jointly models the clustering in each dataset, inferring individual clusterings for each dataset that are informed by similarity in the other clusterings. MDI learns the similarity between the datasets being analysed and 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 not strong modelling assumptions. Thus we can include the PPI data and expect it to contribute to our final clustering despite the expectation that there will be less shared information between the Timecourse dataset with its large set of ncRNAs might and this PPI dataset.

The datasets were reduced to 551 items by considering only the genes with no missing data in the PPI and ChIP-chip data. The choices to reduce the datasets to these 551 genes are the same steps as in Kirk *et al.* (2012). The datasets were modelled using a mixture of Gaussian processes in the Timecourse dataset and Multinomial distributions in the ChIP-chip and PPI datasets. To ensure that our mixture model is initially overfitted we set $K_{max} = 275 \approx \frac{N}{2}$.

4 Results

4.1 Simulations

A summary of the results for a selection of the simulation scenarios are shown in table 3 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 subpopulation structure. Instead the method is blinded by the irrelevant features and identifies a clustering of $K = 1$. 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 practical implementations of consensus clustering and any individual long chain. 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.

Table 3. Median ARI for 100 datasets within three simulation scenarios between the generating labels and the predicted clustering for a subset of methods. CC(r, s) indicates consensus clustering using the r^{th} sample from s chains.

Method	Irrelevant features 100	Simple 2D	Small N, large P
Mclust	0.000	0.976	1
Bayesian (Pooled)	0.986	0.648	1
CC(10000, 100)	0.974	0.553	1
CC(10000, 50)	0.974	0.516	1
CC(10000, 10)	0.971	0.343	1
CC(100, 100)	0.892	0.561	1
CC(100, 50)	0.889	0.508	1
CC(100, 10)	0.872	0.355	1
CC(10, 100)	0.695	0.559	1
CC(10, 50)	0.611	0.516	1
CC(10, 10)	0.380	0.370	1

Figure 3 shows an example of different long chains becoming trapped in different modes and failing to explore a common partition space. In the simulations shown here the overlap between the modes and signal in the data is clear enough that one can pick the true clustering structure and select the chain that best represents this, but in a practical analysis additional steps (for example, running more chains) would have to be followed to decide upon which chain best represented the posterior distribution before proceeding in the anlaysis.

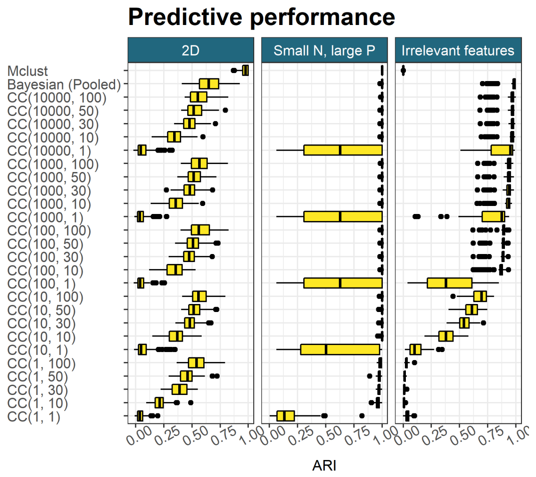

Fig. 2. Model performance in a subset of simulation scenarios. We notice that for a constant chain lenght, increasing the number of chains used follows a pattern of diminishing returns with strong initial gains and then diminishing returns. A similar pattern emerges in increasing chain length for a constant number of chains. We also notice that the Consensus clustering appears bound by some asymptote, defined by the pooled long chains.

Figure 4 reveals the gains in computation runtime achieved by consensus clustering when a parallel environment is available. As Consensus clustering means that chains can be on the scale of 100 times shorter, thus offering huge time savings when run upon a high performance computing cluster, and even on a laptop of 8 cores running a Consensus of 1,000 chains of length 1,000 will be about twice as fast as running 10 chains of length 100,000 due to parallelisation.

4.2 Yeast

4.2.1 Ensemble choice

We use an ensemble of depth $R = 10001$ and width $S = 1000$ with a base learner of MDI. This ensemble depth and width were decided using the stopping rule from section 2.9. We include a plot of the mean squared difference between the Consensus matrix for $R = 10001$ and $S = 1000$ to a range of smaller and shallower ensembles (figure 5).

We decide to stop increasing at $R = 10001$ as there is little change between Consensus matrices for increasing chain depth from $R = 5001$ to $R = 10001$ across the three datasets.

4.2.2 Integrated clusters

We define a gene to be integrated across some set of datasets if the gene has the same label in each of these datasets for at least half of the recorded clustering samples. Integrated genes are those most affected by the integrative aspect of the analysis and therefore we focus upon these. We focus upon the genes that integrated across the Timecourse and ChIp-chip datasets as the PPI dataset appears to contribute less to the analysis. 261 genes integrate acorss this pair of datasets (nearly half of the genes present), of which 56 integrate across all three datasets. We show the integrated clusters that emerge in figure 6. In this plot we exclude the 15 clusters where more than half of the member genes have no interactions in the ChIP-chip data and any clusters of one. We find that a small number of transcription factors dominate, with different combinations emerging across the 10 clusters (these are listed in the supplementary materials). Many of these 10 correspond to transcription factors that are well known to regulate cell cycle expression (Simon *et al.*, 2001).

As an example, we briefly analyse clusters 9 and 16 in greater depth. We used the `org.Sc.sgd.db` (Carlson *et al.*, 2014) package to find gene descriptions (which are included in the supplemenatry materials). Cluster 9 has strong association with MBP1 and some interactions with SWI6 in figure 6. The Mbp1-Swi6p complex, MBF, is associated with DNA replication (Iyer *et al.*, 2001). The timepoint of 0 minutes in the Timecourse data is at the START checkpoint, or the G1/S-phase. The members of

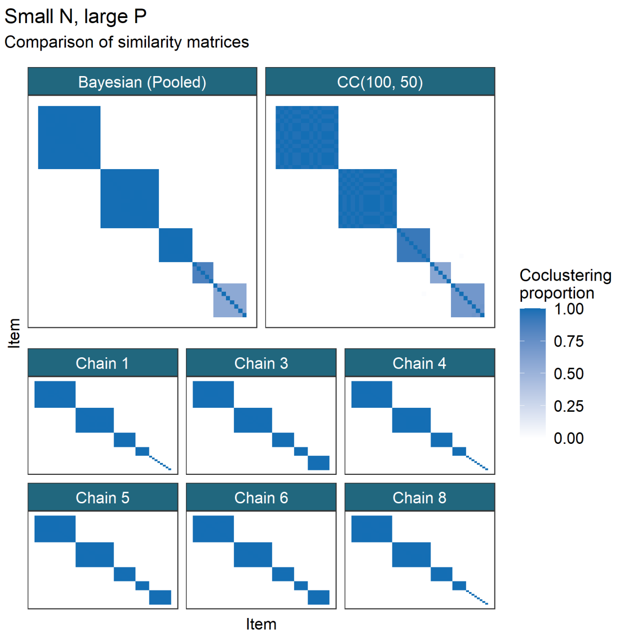

Fig. 3. In the first row the PSM of the pooled Bayesian samples is compared to the CM for CC(100, 50), with a common ordering of rows and columns in both heatmaps. There is very little difference between these two objects, with similar clusters emerging (the different solid blocks). Both matrices also contain some uncertainty about the final clustering, with some values being between 0 and 1. In the next two rows the PSMs constructed from 6 of the long chains that passed the stationarity test are displayed. Each PSMs is binary, with all entries being 0 or 1. This means only a single clustering is sampled across the chain, implying very little uncertainty in the partition. However, three different modes emerge across the different chains showing that the chains are failing to explore the full support of the posterior distribution of the clustering.

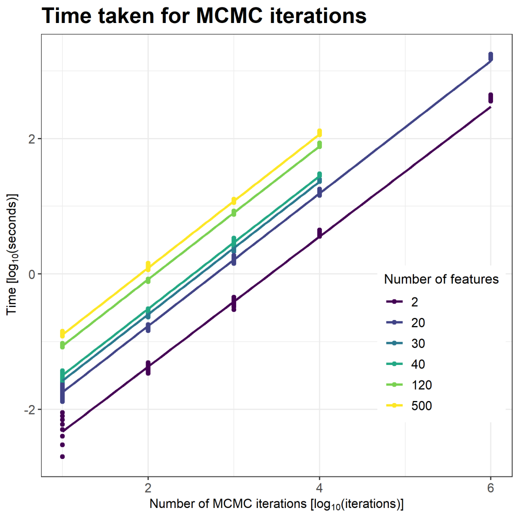

Fig. 4. The time taken for different numbers of iterations of MCMC moves in $\log(s)$. The relationship between chain length, R, and the time taken is linear (the slope is approximately 1 on the $\log$ scale), with a change of intercept for different dimensions.

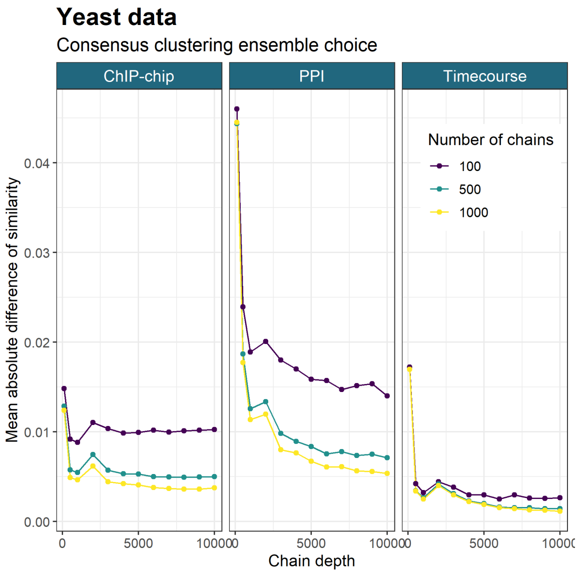

Fig. 5. The mean absolute difference between the sequential Consensus matrices. For a set of chain lengths, $R' = \{r_1, \ldots, r_I\}$ and number of chains, $S' = \{s_1, \ldots, s_J\}$, we take the mean of the absolute difference between the Consensus matrix for (r_i, s_j) and (r_{i-1}, s_j) (here $R' = \{101, 501, 1001, 2001, \ldots, 10001\}$ and $S' = \{100, 500, 1000\}$). It appears that the improvement from increasing the ensemble depth and width is diminishing, with no strong improvement after $R = 6,000$, and increasing the number of learners from 500 to 1,000 has small effect.

cluster 9 begin highly expressed at this point before quickly dropping in expression (in the first of the 3 cell cycles). This suggests that a large number of transcripts are produced immediately in advance of S-phase, and thus are required for the first stages of DNA synthesis. The description of the genes found using the data external to our clustering also supports this hypothesis. Many of the genes in cluster 9 we find to be associated with DNA replication, repair and/or recombination. We see several genes that control the replication checkpoint, TOF1, MRC1 and RAD53 emerge (Bando *et al.*, 2009; Lao *et al.*, 2018). Some members of the cohesin complex are also in this cluster. Cohesin is associated with sister chromatid cohesion, which is established during the DNA synthesis phase of the cell cycle (Tóth *et al.*, 1999), but also contributes to transcription regulation, DNA repair, chromosome condensation, homolog pairing(Mehta *et al.*, 2013).

Cluster 16 consists of genes whose products form the histones H1, H2A, H2B, H3 and H4 and then three others members, GAS3, NRM1 and PDS1. Histones are the chief protein components of chromatin (Fischle *et al.*, 2003) and are important contributors to gene regulation (Bannister and Kouzarides, 2011). They are known to peak in expression in S-phase (Granovskaia *et al.*, 2010), which matches the first peak of this cluster early in the time series. Of the other members, NRM1 is transcriptional co-repressor of MBF-regulated gene expression acting at the transition from G1 to S phase (de Bruin *et al.*, 2006; Aligianni *et al.*, 2009). Pds1p binds to and inhibits the Esp1 class of sister separating proteins, preventing sister chromatids separation before M phase (Ciosk *et al.*, 1998; Tóth *et al.*, 1999). The remaining member, GAS3, is poorly understood. 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 (Cooper *et al.*, 2009) and is instantiated in S-phase (Tóth *et al.*, 1999). These results along with the very similar expression profile to the histone genes suggests GAS3 is involved more directly in DNA replication or chromatid cohesion as part of the S-phase genes.

5 Discussion

We have applied Consensus clustering to Bayesian mixture models. This sidesteps the problem of mixing for MCMC based clustering methods and also improves the speed at which an analysis can be performed, using a parallel environment. The proposed method has demonstrated good performance on simulation studies, finding meaningful structure and approximating Bayesina inference when the Markov chain is exploring the full support of the posterior. However, we have shown that if a finite Markov chain fails to describe the full posterior and is itself only approximating Bayesian inference, our method has better ability to represent several modes in the data than individual chains and thus offers a more consistent and reproducible analysis. Furthermore, Consensus clustering is significantly faster in a parallel environment than inference using individual chains, while retaining the ability to consistently infer K, the number of occupied components present.

Based upon the simulations we proposed a method of assesing ensemble stability and deciding upon ensemble size. We then used this to decide upon our ensemble size in an integrative analysis of yeast cell cycle data, using a base learner of MDI, an extension of Bayesian mixture models to the multiple dataset case. We uncovered a large number of genes with shared signal across several datasets. We then showed that the clusters found have biological meaning, validated using external data. We also found signal for possible novel biology, showing that Consensus clustering can be used in novel situations to discover meaning. We expect that Consensus clustering could be applied with similar success to other extensions of Bayesian mixture models such as Clusternomics (Gabasova *et al.*, 2017).

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.

Funding

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

References

Alberts, B. *et al.* (2018). The cell cycle and programmed cell death. In W. Dubitzky, O. Wolkenhauer, K.-H. Cho, and H. Yokota, editors, *Molecular biology of the cell*, chapter 17. Garland Science, Taylor and Francis Group, New York, NY, 6 edition.

Aligianni, S. *et al.* (2009). The fission yeast homeodomain protein yox1p binds to mbf and confines mbf-dependent cell-cycle transcription to g1-s via negative feedback. *PLoS Genet*, **5**(8), e1000626.

Arthur, D. and Vassilvitskii, S. (2006). k-means++: The advantages of careful seeding. Technical report, Stanford.

Bai, J. P. *et al.* (2013). Strategic applications of gene expression: from drug discovery/development to bedside. *The AAPS journal*, **15**(2), 427–437.

Bando, M. *et al.* (2009). Csm3, tof1, and mrc1 form a heterotrimeric mediator complex that associates with dna replication forks. *Journal of Biological Chemistry*, **284**(49), 34355–34365.

Bannister, A. J. and Kouzarides, T. (2011). Regulation of chromatin by histone modifications. *Cell research*, **21**(3), 381–395.

Breiman, L. (2001). Random forests. *Machine learning*, **45**(1), 5–32.

Carlson, M. *et al.* (2014). Org. sc. sgd. db: Genome wide annotation for yeast. *R package version*, **2**(1).

Chan, C. *et al.* (2008). Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry Part A: The Journal of the International Society for Analytical Cytology*, **73**(8), 693–701.

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

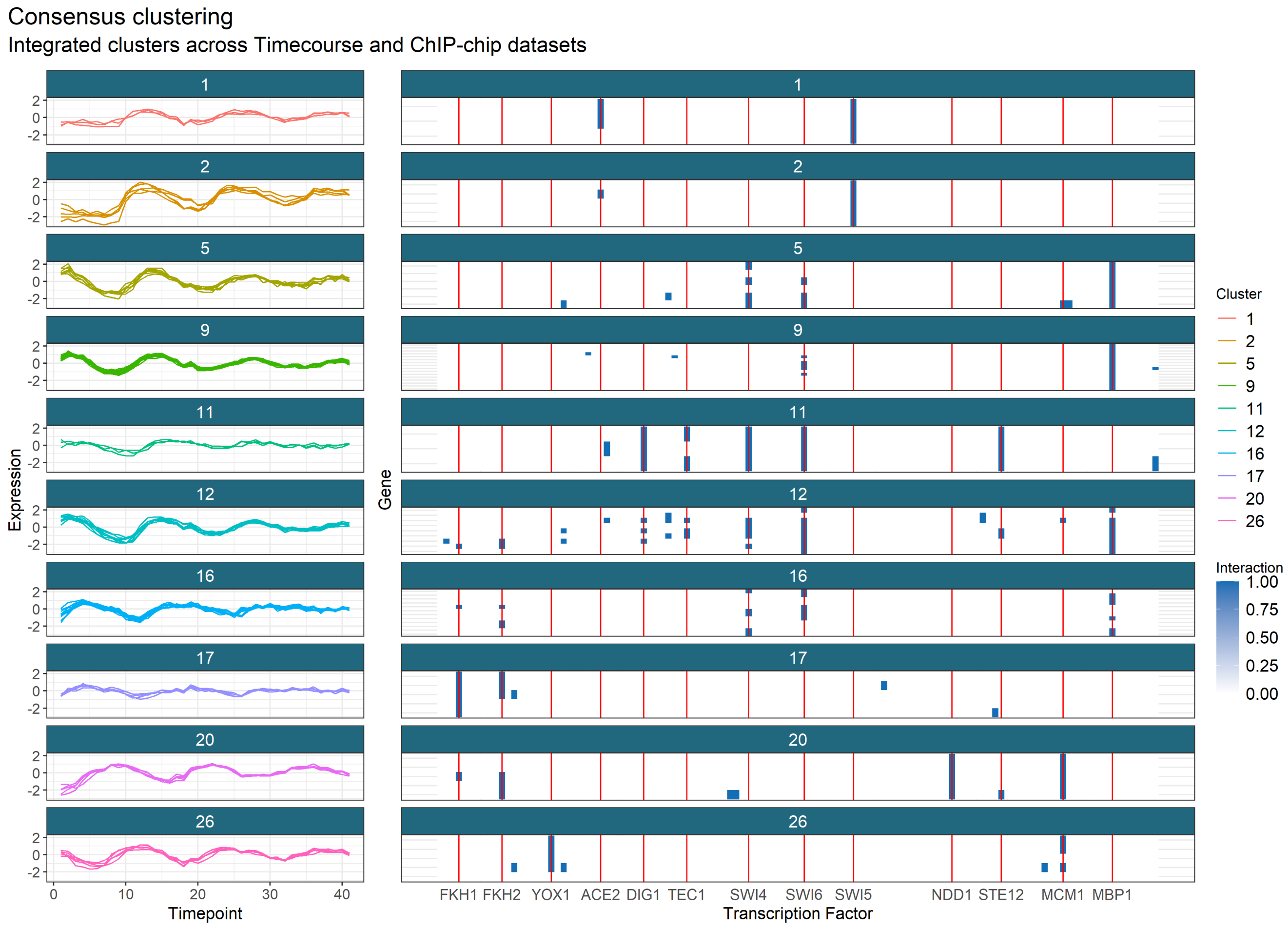

Fig. 6. The integrated clusters across the Timecourse and ChIP-chip datasets. We exclude the clusters with no interactions in the ChIP-chip dataset and include a red line for the Transcription factors that dominate the clustering structure in the ChIP-chip dataset. The clusters in the time series data are quite well banded and distinct (having different periods, amplitudes or both) and in the ChIP-chip data a small number of Transcription factors dominate the clustering structure. The clusters have tight, unique signatures in the Timecourse dataset and tend to be defined by a small number of well studied transcription factors in the ChIP-chip dataset.

Chen, K. C. *et al.* (2004). Integrative analysis of cell cycle control in budding yeast. *Molecular biology of the cell*, **15**(8), 3841–3862.

Ciosk, R. *et al.* (1998). An esp1/pds1 complex regulates loss of sister chromatid cohesion at the metaphase to anaphase transition in yeast. *Cell*, **93**(6), 1067–1076.

Cooper, K. F. *et al.* (2009). Pds1p is required for meiotic recombination and prophase i progression in saccharomyces cerevisiae. *Genetics*, **181**(1), 65–79.

Crook, O. M. *et al.* (2018). A bayesian mixture modelling approach for spatial proteomics. *PLoS computational biology*, **14**(11), e1006516.

de Bruin, R. A. *et al.* (2006). Constraining g1-specific transcription to late g1 phase: the mbf-associated corepressor nrm1 acts via negative feedback. *Molecular cell*, **23**(4), 483–496.

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.

Fischle, W. *et al.* (2003). Histone and chromatin cross-talk. *Current opinion in cell biology*, **15**(2), 172–183.

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

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

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, **38**(4), 367–378.

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.

Gabasova, E. *et al.* (2017). Clusternomics: Integrative context-dependent clustering for heterogeneous datasets. *PLoS computational biology*, **13**(10), e1005781.

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.

Granovskaia, M. V. *et al.* (2010). High-resolution transcription atlas of the mitotic cell cycle in budding yeast. *Genome biology*, **11**(3), 1–11.

Harbison, C. T. *et al.* (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**(7004), 99–104.

Hejblum, B. P. *et al.* (2015). Time-course gene set analysis for longitudinal gene expression data. *PLoS computational biology*, **11**(6), e1004310.

Hejblum, B. P. *et al.* (2019). Sequential dirichlet process mixtures of multivariate skew t-distributions for model-based clustering of flow cytometry data. *The Annals of Applied Statistics*, **13**(1), 638–660.

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

Ingalls, B. *et al.* (2007). Systems level modeling of the cell cycle using budding yeast. *Cancer informatics*, **3**, 117693510700300020.

Iyer, V. R. *et al.* (2001). Genomic binding sites of the yeast cell-cycle transcription factors sbf and mbf. *Nature*, **409**(6819), 533–538.

Jiménez, J. *et al.* (2015). Live fast, die soon: cell cycle progression and lifespan in yeast cells. *Microbial Cell*, **2**(3), 62.

Juanes, M. A. (2017). Methods of synchronization of yeast cells for the analysis of cell cycle progression. In *The Mitotic Exit Network*, pages 19–34. Springer.

Kirk, P. *et al.* (2012). Bayesian correlated clustering to integrate multiple datasets. *Bioinformatics*, **28**(24), 3290–3297.

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

Lao, J. P. *et al.* (2018). The yeast dna damage checkpoint kinase rad53 targets the exoribonuclease, xrn1. *G3: Genes, Genomes, Genetics*, **8**(12), 3931–3944.

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

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.

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

Mehta, G. D. *et al.* (2013). Cohesin: functions beyond sister chromatid cohesion. *FEBS letters*, **587**(15), 2299–2312.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(4), 417–473.

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.

Prabhakaran, S. *et al.* (2016). Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. In *International Conference on Machine Learning*, pages 1070–1079.

Richardson, S. and Green, P. J. (1997). On bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society: series B*, **59**(4), 731–792.

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

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

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.

Simon, I. *et al.* (2001). Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, **106**(6), 697–708.

Stark, C. *et al.* (2006). Biogrid: a general repository for interaction datasets. *Nucleic acids research*, **34**(suppl_1), D535–D539.

Tóth, A. *et al.* (1999). Yeast cohesin complex requires a conserved protein, eco1p (ctf7), to establish cohesion between sister chromatids during dna replication. *Genes & development*, **13**(3), 320–333.

Tyson, J. J. *et al.* (2013). Cell cycle, budding yeast. In W. Dubitzky, O. Wolkenhauer, K.-H. Cho, and H. Yokota, editors, *Encyclopedia of Systems Biology*, pages 337–341. Springer New York, New York, NY.

Van Havre, Z. *et al.* (2015). Overfitting bayesian mixture models with an unknown number of components. *PloS one*, **10**(7), e0131739.

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.

Von Luxburg, U. and Ben-David, S. (2005). Towards a statistical theory of clustering. In *Pascal workshop on statistics and optimization of clustering*, pages 20–26. Citeseer.

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*.