

Consensus clustering for Bayesian mixture models: Supplementary materials

Stephen Coleman, Paul DW Kirk and Chris Wallace

October 16, 2020

Abstract

Description of models used and analyses performed.

1 Definitions

Definition 1 (Consensus matrix) *Given S clusterings for a dataset of N items, $c_s = (c_{s1}, \dots, c_{sN})$, the Consensus matrix is a $N \times N$ matrix where the $(i, j)^{th}$ entry records the proportions of clusterings for which items i and j are allocated the same label. More formally, it is the matrix \mathbb{C} such that*

$$\mathbb{C}(i, j) = \frac{1}{S} \sum_{s=1}^S \mathbf{I}(c_{si} = c_{sj}) \quad (1)$$

where $\mathbb{I}(\cdot)$ is the indicator function taking a value of 1 if the argument is true and 0 otherwise.

Definition 2 (Posterior similarity matrix) *A Consensus matrix for which all the clusterings are generated from a converged Markov chain for some Bayesian clustering model. Sometimes abbreviated to PSM.*

2 The models

2.1 Individual dataset

In the simulations (see section 3) where individual datasets are modelled a *Bayesian mixture model* is used. We write the basic mixture model for independent items $X = (x_1, \dots, x_N)$ as

$$x_i \sim \sum_{k=1}^K \pi_k f(x_i | \theta_k) \quad \text{independently for } i = 1, \dots, N \quad (2)$$

where $f(\cdot | \theta)$ is some family of densities parametrised by θ . A common choice is the Gaussian density function, with $\theta = (\mu, \sigma^2)$. This is model used in the

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

$$\begin{aligned} p(c_n = k) &= \pi_k \quad \text{for } k = 1, \dots, K, \\ x_n | c_n &\sim f(x_n | \theta_k) \quad \text{independently for } n = 1, \dots, N. \end{aligned} \quad (3)$$

The joint model can then be written

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

An assumption that is frequently used is that the density of each feature is independent, with $\theta_k = (\theta_{k1}, \dots, \theta_{kP})$ for all $k = 1, \dots, K$. Furthermore, conditional independence is assumed between certain parameters such that the model reduces to

$$p(X, c, \theta, \pi, K) = p(\pi|K)p(\theta|K)p(K) \prod_{n=1}^N p(x_n|c_n, \theta_{c_n})p(c_n|\pi, K). \quad (4)$$

Additional flexibility is provided by the inclusion of hyperparameters on the priors for π and θ , denoted α and η respectively. In our context where $\theta = (\mu, \sigma^2)$, we use

$$\sigma^2 \sim \Gamma^{-1}(a, b) \quad (5)$$

$$\mu \sim \mathcal{N}(\xi, \frac{1}{\lambda}\sigma^2) \quad (6)$$

The directed acyclic graph (**DAG**) for this model is shown in figure 1. The value of the hyperparameters we use are

$$\alpha = 1, \quad (7)$$

$$\xi = 0.0, \quad (8)$$

$$\lambda = 1.0, \quad (9)$$

$$a = 2.0, \quad (10)$$

$$b = 2.0. \quad (11)$$

2.2 Integrative clustering

We are interested in the use of Consensus clustering for integrative methods. We use Multiple Dataset Integration (**MDI**, Kirk et al., 2012) as an example of a Bayesian integrative clustering method.

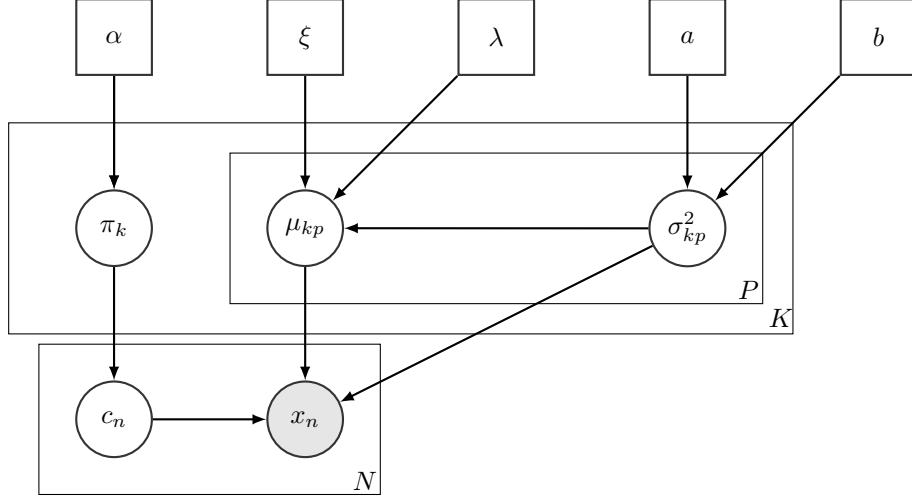


Figure 1: Directed acyclic graph for the mixture of Gaussians used.

In MDI information is shared between datasets. Dataset specific clusterings are modelled with more information shared between datasets with more similar clustering structure. This can be seen in the prior on the allocation of the n^{th} item across L datasets

$$p(c_{n1}, \dots, c_{nL}) \propto \prod_{l=1}^L \pi_{c_{nl}} \prod_{l=1}^{L-1} \prod_{m=l+1}^L (1 + \phi_{lm} \mathbb{I}(c_{nl} = c_{nm})) \text{ for } n = 1, \dots, N. \quad (12)$$

ϕ_{lm} is the parameter defined by the correlation of the clusterings for the l^{th} and m^{th} datasets and is also sampled in each iteration. As ϕ_{lm} increases more mass is placed on the common partition for these datasets. Conversely, in the limit $\phi_{lm} \rightarrow 0$ we have independent mixture models. The DAG for this model for three datasets is shown in figure 2.

3 Simulations

A number of scenarios are defined by the parameters that they use in generating individual simulations using algorithm 1. These parameters are shown in table 1. We included each scenario to test certain concepts of the method and to explore behaviour due to specific characteristics of real data.

- **2D:** a low dimensional scenario within which we expect **Mclust** to perform well and the long chains to converge and explore the full support of the posterior distribution.

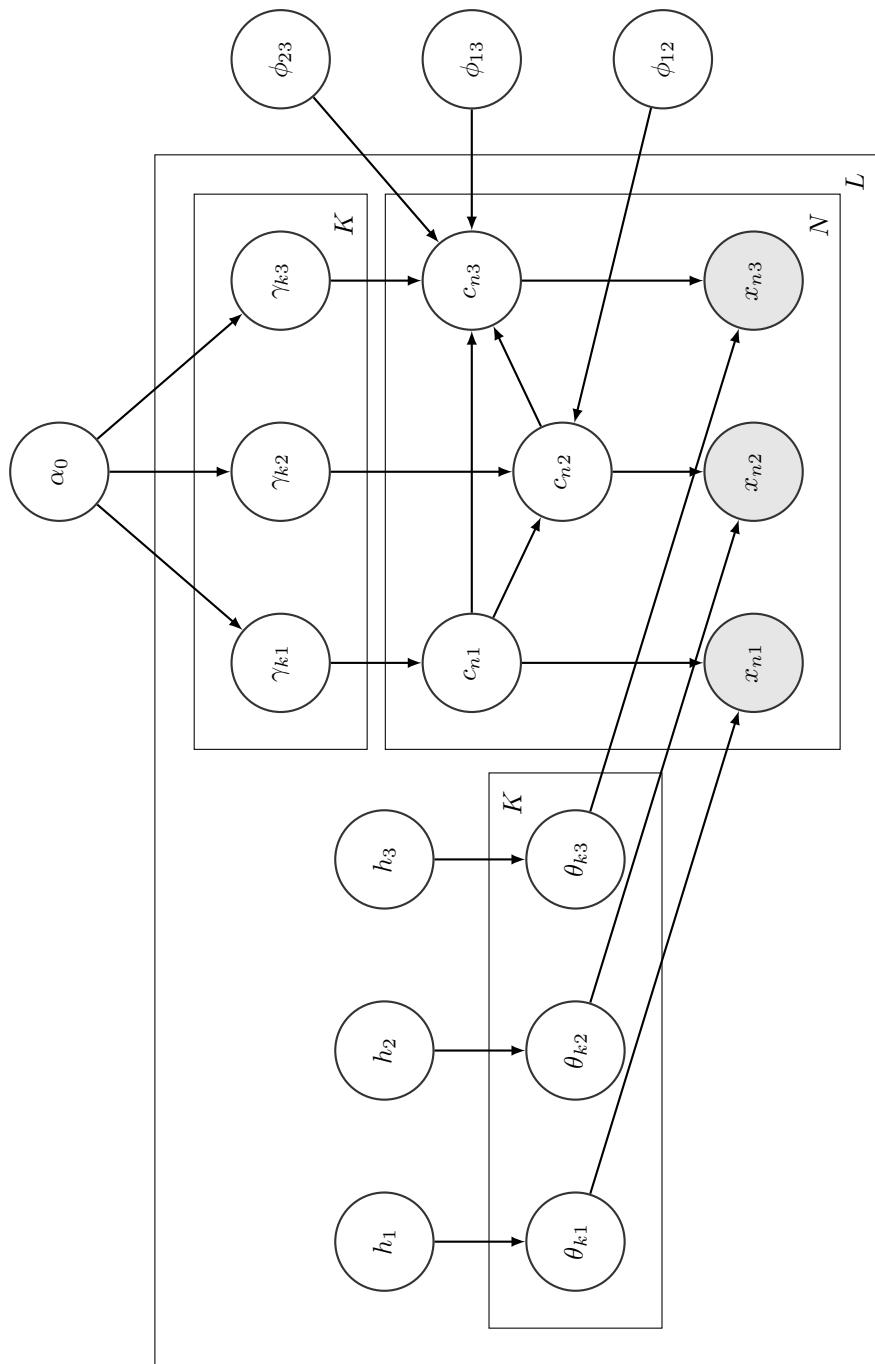


Figure 2: Directed acyclic graph for the Multiple Dataset Integration model for $L = 3$ datasets.

Algorithm: Simulation generation

Input: Distance between means Δ_μ
A common standard deviation σ^2
A number of clusters K
The number of items to generate in total N
The number of features to generate in total P
An indicator vector of feature relevance $\phi = (\phi_1, \dots, \phi_P)$
The expected proportion of items in each cluster $\pi = (\pi_1, \dots, \pi_K)$
A method for sampling x times from the array y , with weights π :
 $Sample(y, x, \pi)$
A method for permuting a vector x : $Permute(x)$
A method for generating a value from a univariate Gaussian
distribution with mean μ and standard deviation σ^2 : $Gaussian(\mu, \sigma^2)$

Output: A dataset, X

The generating cluster labels $c = (c_1, \dots, c_N)$

```

begin
    /* initialise the empty data matrix */ 
     $X \leftarrow 0_{N \times P};$ 
    /* create a matrix of  $K$  means */ 
     $\mu \leftarrow (\Delta_\mu, \dots, K\Delta_\mu);$ 
    /* generate the allocation vector */ 
     $c \leftarrow Sample(1 : K, N, \pi);$ 
     $M \leftarrow 0_{N \times N};$ 
    for  $p = 1$  to  $P$  do
        /* Test if the feature is relevant, if relevant
           generate data from a mixture of univariate
           Gaussians, otherwise draw all items from the same
           distribution */ 
        if  $\phi_p = 1$  then
             $\nu \leftarrow Permute(\mu);$ 
            for  $n = 1$  to  $N$  do
                |  $X(n, p) \leftarrow Gaussian(\nu_{c_n}, \sigma^2)$ 
            end
        end
        if  $\phi_p = 0$  then
            for  $n = 1$  to  $N$  do
                |  $X(n, p) \leftarrow Gaussian(0, \sigma^2)$ 
            end
        end
    end
    /* Mean centre and scale the data */ 
     $X \leftarrow Normalise(X)$ 
end

```

Algorithm 1: Data generation for a mixture of Gaussian with independent features. This algorithm is implemented in the `generateSimulationDataset` function from the `mdiHelpR` package available at www.github.com/stcolema mdiHelpR.

Scenario	N	P_s	P_n	K	$\Delta\mu$	σ^2	π
2D	100	2	0	5	3.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
No structure	100	0	2	1	0.0	1	1
Base Case	200	20	0	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Large standard deviation	200	20	0	5	1.0	9	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Large standard deviation	200	20	0	5	1.0	25	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Irrelevant features	200	20	10	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Irrelevant features	200	20	20	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Irrelevant features	200	20	100	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Varying proportions	200	20	0	5	1.0	1	$(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16})$
Varying proportions	200	20	0	5	0.4	1	$(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16})$
Small N , large P	50	500	0	5	1.0	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$
Small N , large P	50	500	0	5	0.2	1	$(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$

Table 1: Parameters defining the simulation scenarios as used in generating data and labels.

- *No structure*: included to reassure fears that Consensus clustering has a predilection to finding clusters where none exist (Şenbabaoglu et al., 2014a,b).
- *Base case*: highly informative datasets that all methods are expect to find the true generating labels quite easily and is included to benchmark other scenarios that are variations of this setting.
- *Large standard deviation*: these two scenarios investigate the degree of distinction required between clusters for the methods to uncover their structure.
- *Irrelevant features*: these scenarios investigate how robust the methods are to irrelevant features.
- *Varying proportions*: these scenarios investigate how well each method uncovers clusters when the clusters have significantly different membership counts.
- *Small N , large P* : an investigation of behaviour when the number of features is far greater than the number of items.

3.1 Bayesian analysis

For each simulation we run 10 chains of 1 million iterations. These chains are then thinned to every thousandth sample. We used discarded the first 10,000 iterations to remove burn-in bias. This leaves 990 samples per chain. To check if the chains were converged we used

- the Geweke convergence diagnostic (Geweke et al., 1991) to investigate within-chain stationarity, and

- the potential scale reduction factor (\hat{R} , Gelman et al., 1992) and the Vats-Knudson extension (*stable* \hat{R} , Vats and Knudson, 2018) to check across-chain convergence.

The Geweke convergence diagnostic is a standard Z-score; it compares the sample mean of two sets of samples (in this case buckets of samples from the first half of the samples to the sample mean of the entire second half of samples). It is calculated under the assumption that the two parts of the chain are asymptotically independent and if this assumption holds (i.e. the chain is sampling the same distribution in both samples) than the scores are expected to be standard normally distributed. If a chain's Geweke convergence diagnostic passed a Shapiro-Wilks test for normality (Shapiro and Wilk, 1965) (based upon a threshold of 0.05), we considered it to have achieved stationarity and included it in the model performance analysis.

\hat{R} is expected to approach 1.0 if the set of chains are converged. Low \hat{R} is not sufficient in itself to claim chain convergence, but values above 1.1 are clear evidence for a lack of convergence (Gelman et al., 2013). Vats and Knudson (2018) show that this threshold is significantly too high (1.01 being a better choice) and propose extensions to \hat{R} that enable a more formal rule for a threshold. We use their method as implemented in the R package `stableGR` (Knudson and Vats, 2020) as the final check of convergence. An example of the \hat{R} series across the 100 simulations for a scenario where chains are well-behaved is shown in figure 3.

We focus upon stationarity of the continuous variables. This is as assessing convergence of the allocation labels is difficult due to the label-switching problem. In our simulations, the only recorded continuous variable is the concentration parameter of the Dirichlet distribution for the component weights.

We pooled the samples from the stationary chains and used these to form a Posterior similarity matrix. There are three possibilities to consider this decision under.

- The chains are converged and agree upon the distribution sampled (see figure 4 for an example).
- The chains are not in agreement upon the partition sampled, becoming trapped in different modes. However, a mode does dominate being the mode present in a majority of chains (see figure 5 for an example of this behaviour).
- The chains are not in agreement and no one mode dominates among chains.

In the first case pooling has no effect upon the predicted clustering compared to using any one chain. In the second case it feels natural that one would use the mode that dominates. Pooling the samples effectively does this for the predictive performance of the method as the mode with the greatest number of samples across the chains dominates, however the uncertainty for this mode is increased. In the third case the analysis is non-trivial and further thought,

Base case: Convergence across chains

0% of simulations failed to converge

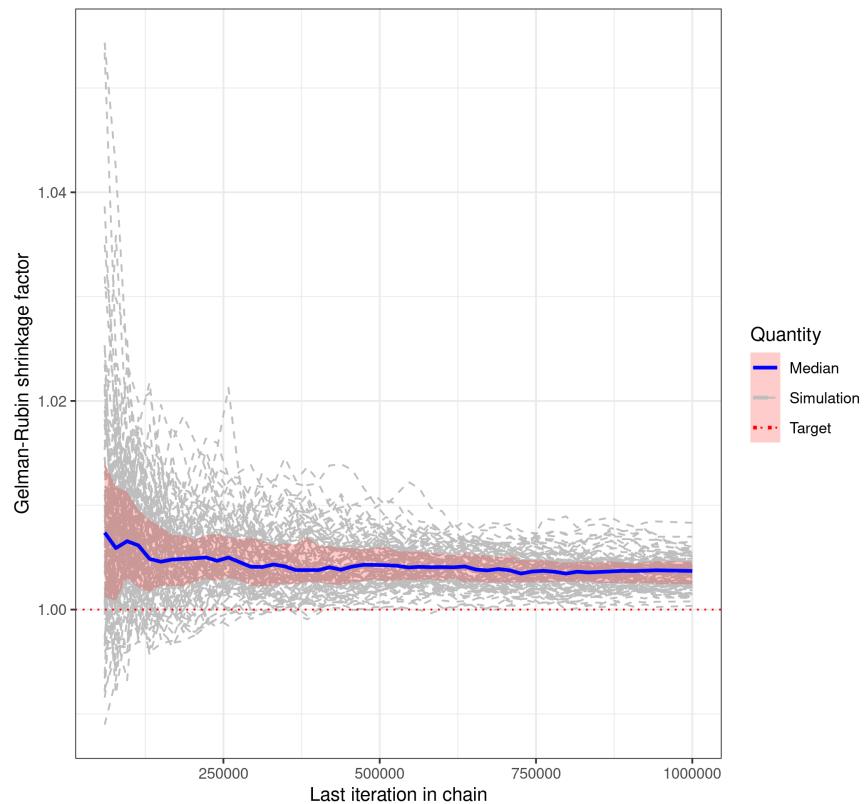


Figure 3: The \hat{R} values for each simulation (in dotted grey), the median value and the interquartile range. One can see that \hat{R} approaches 1.0. The “0% of simulations failed to converge” is a statement based upon the percentage of simulations which passed the test of stable \hat{R} .

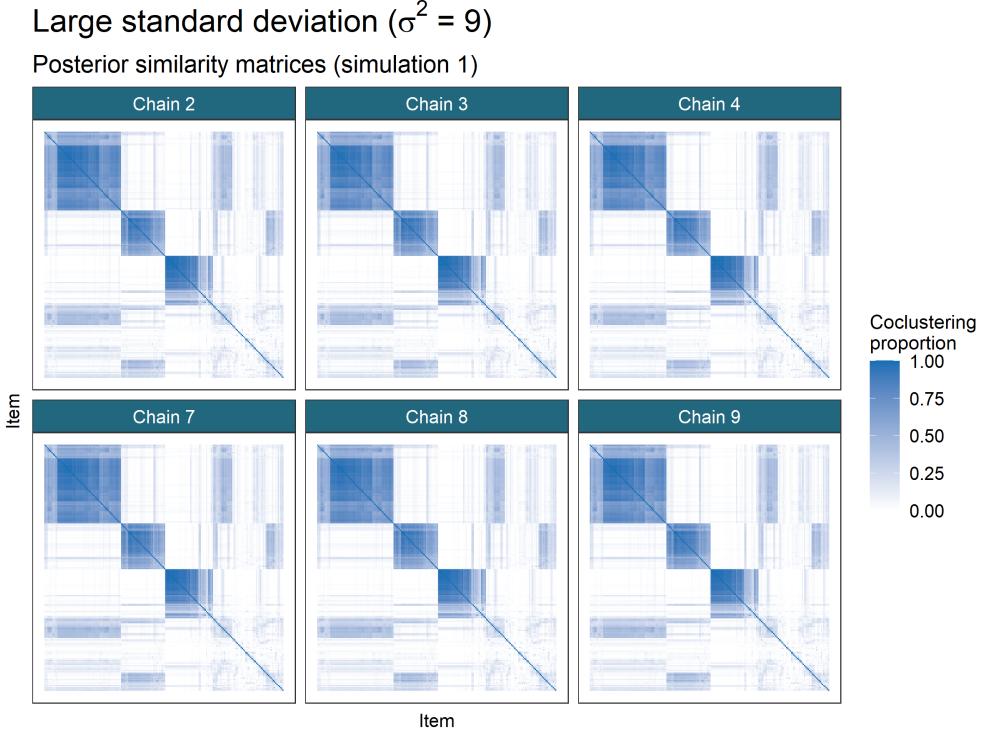


Figure 4: Posterior similarity matrices for the simulation generated using a random seed set to 1 for the first large standard deviation scenario from table 1. This is an example of all stationary chains agreeing in a simulation (and thus pooling of samples is no different to using any choice of chain for the performance analysis). Ordering of rows and columns is defined by hierarchical clustering of the first matrix in the series, in this case that from Chain 2.

chains and samples would be required. Thankfully the generating process used means that only a small number of modes ever emerge in the PSMs and this case does not arise (based upon the simulations we investigated).

3.2 Consensus clustering analysis

A range of ensembles are investigated. All combinations of chain depth, $R = (1, 10, 100, 1000, 10000)$, and the number of chains, $S = (1, 10, 30, 50, 100)$ were used, a total of 25 different ensembles. A consensus matrix was constructed from the samples generated by each ensemble by finding the proportion of samples within which any pair of items are coclustered.

Small N large P ($\Delta\mu = 1.0$)

Posterior similarity matrices (simulation 1)

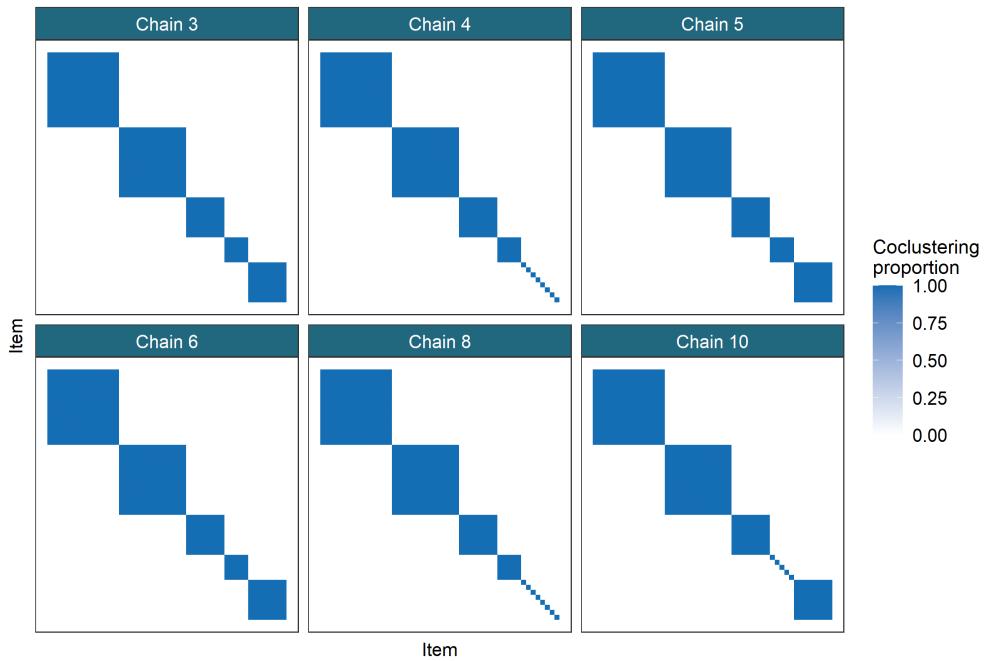


Figure 5: Posterior similarity matrices for the simulation generated using a random seed set to 1 for the first small N , large P scenario from table 1. This is an example of different chains becoming trapped in different modes, but one mode (which does represent the generating structure well) is dominant, being fully present in 3 of the 6 chains, with the two other modes present having significant overlap. Ordering of rows and columns is defined by hierarchical clustering of the first matrix in the series, in this case that from Chain 3.

3.3 Mclust

Mclust was called using the default settings and a range of inputs for the choice of K . This was $K = (2, \dots, \min(\frac{N}{2}, 50))$. The choice of $K = \min(\frac{N}{2}, 50)$ was made to mirror the default value of $K_{max} = 50$ used for the overfitted mixture models, with the bound of $\frac{N}{2}$ to avoid fitting 50 clusters in the *Small N, large P* scenario where $N = 50 = K_{max}$. The model choice is performed using the Bayesian Information Criterion (Schwarz et al., 1978, as implemented in Mclust).

3.4 Model performance

The different models (Bayesian (pooled), Mclust and the 25 Consensus clustering ensembles) were compared under their ability to predict the generating clustering and their uncertainty about this quantity.

4 Yeast

The "Yeast data" consists of three *S. cerevisiae* datasets with gene products associated with a common set of 551 genes. The datasets are:

- microarray profiles of RNA expression from Granovskiaia et al. (2010). This a cell cycle dataset that comprises measurements taken at 41 time points (the **Timecourse** dataset).
- Chromatin immunoprecipitation followed by microarray hybridization (**ChIP-chip**) data from Harbison et al. (2004). This dataset has 117 features.
- Protein-protein interaction (**PPI**) data from BioGrid (Stark et al., 2006). This dataset has 603 features.

The datasets were reduced to 551 items by considering only the genes identified by Granovskiaia et al. (2010) as having periodic expression profiles with no missing data in the PPI and ChIP-chip data, following the same steps as the original MDI paper (Kirk et al., 2012). The datasets were modelled using a base measure of a Gaussian process in the Timecourse dataset and Multinomial distributions in the ChIP-chip and PPI datasets.

4.1 Bayesian analysis

We ran 10 chains of MDI for 36 hours saving every thousandth sample. This resulted in chains of varying length. We reduced the chains to 676 samples as this was the number of samples achieved by the slowest chain. Similar to section 3.1 these chains were then investigated for

- within-chain stationarity using the Geweke convergence diagnostic (Geweke et al., 1991), and

Predictive performance

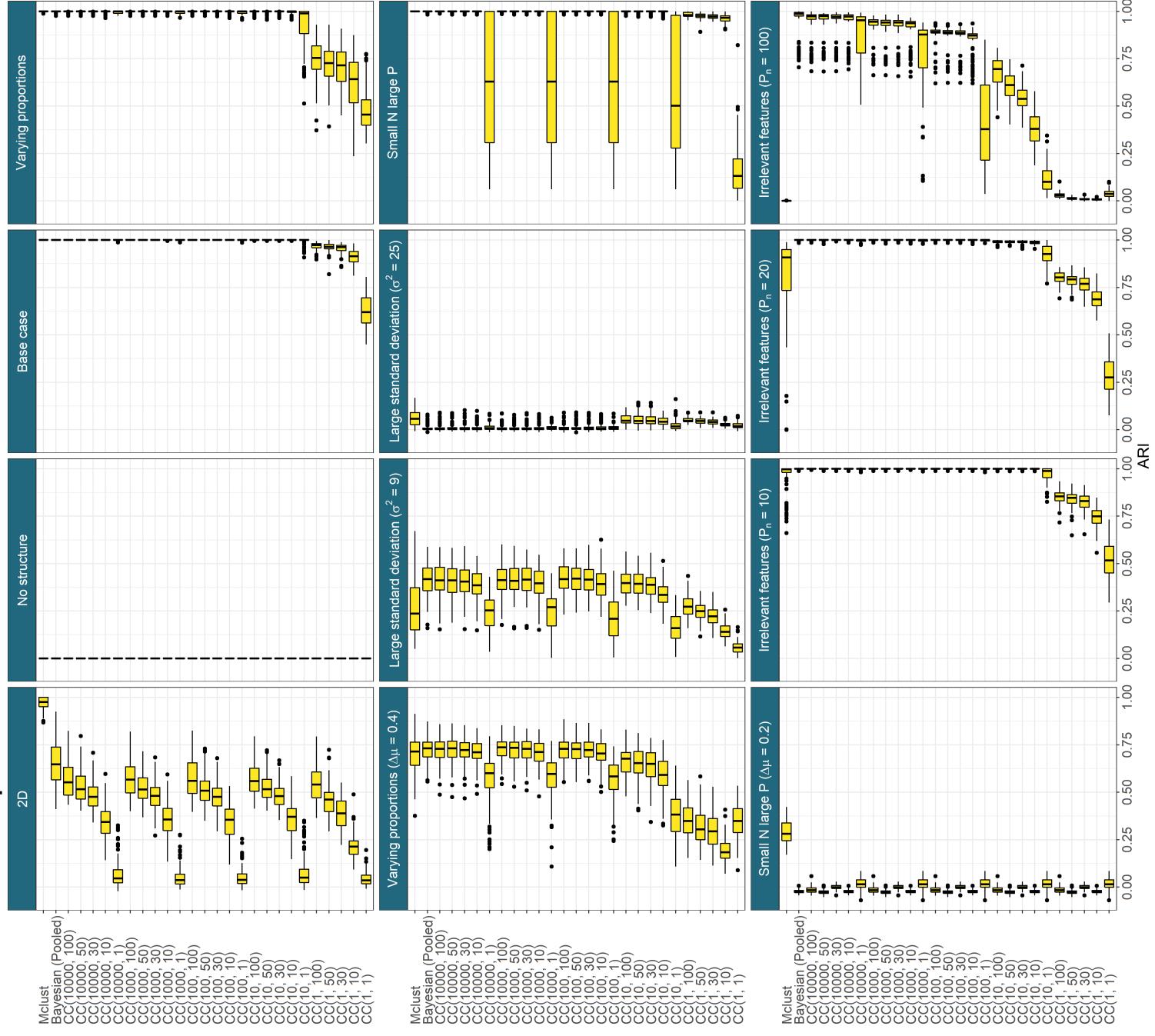


Figure 6: Predictive performance across all simulations. $CC(R, S)$ denotes consensus clustering using the R^{th} sample from S different chains.

Uncertainty quantification

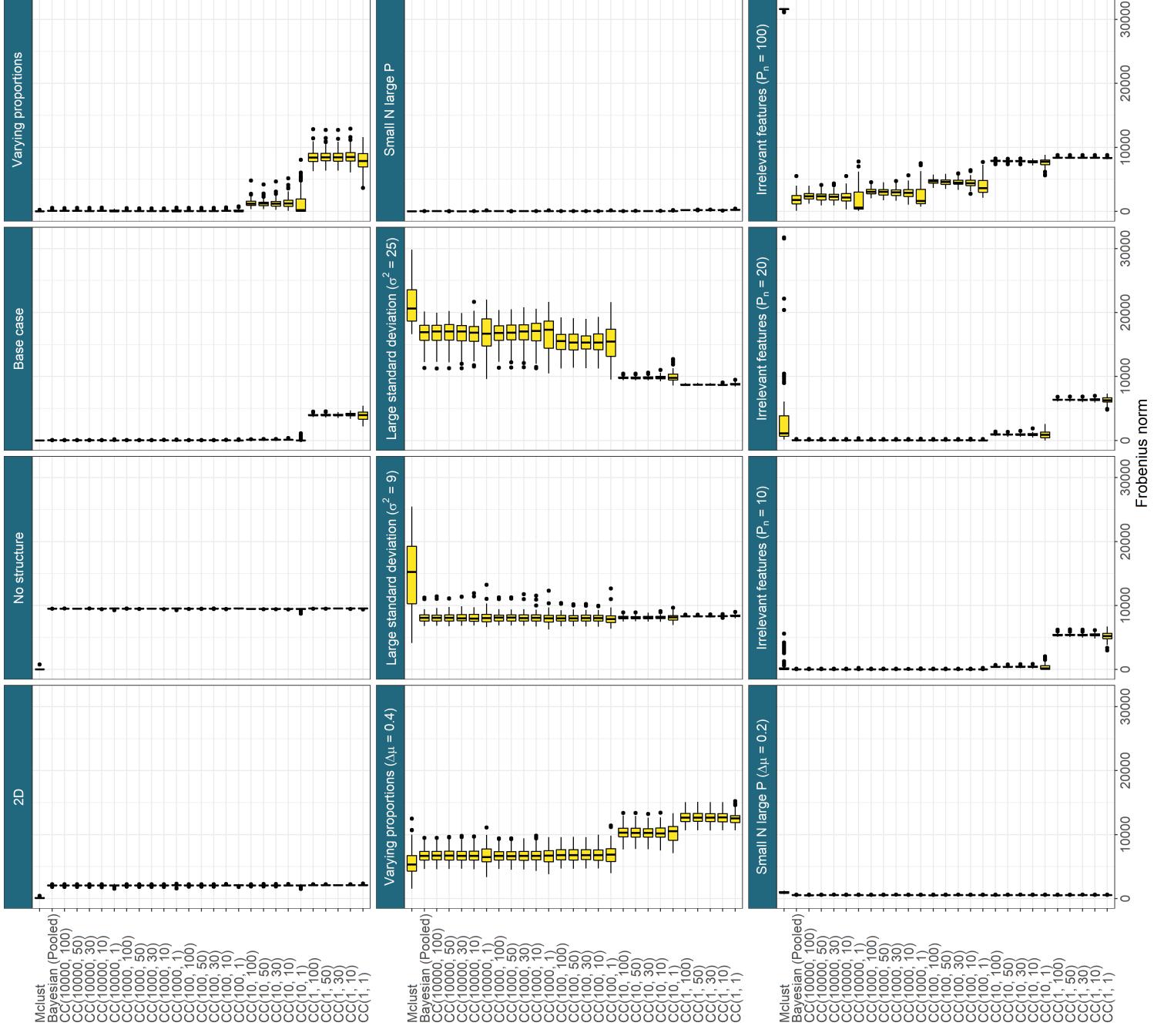


Figure 7: Frobenius norm across simulations. $CC(R, S)$ denotes consensus clustering using the R^{th} sample from S different chains. Lower values are better.

- across-chain convergence using \hat{R} (Gelman et al., 1992) and the Vats-Knudson extension (*stable* \hat{R} , Vats and Knudson, 2018).

Again we focus upon stationarity of the continuous variables. In the implementation of MDI we used Mason et al. (2016), the recorded continuous variables are the concentration parameters of the Dirichlet distribution for the dataset-specific component weights and the ϕ_{ij} parameter associated with the correlation between the i^{th} and j^{th} datasets.

We plot the Geweke-statistic for each chain in figure 8. No chain is perfectly behaved; as we cannot reduce to the set of stationary chains we thus exclude the most poorly behaved chains. Our lack of belief in the convergence of these chains is fortified by the behaviour of \hat{R} (which can be seen in figure 9) and the different distributions sampled for the ϕ_{lm} parameters shown in figure 10.

We visualise the PSMs for each dataset in figure 11.

4.2 Consensus clustering analysis

We investigate an ensemble of depth $R = 1001$ and width $S = 10000$. The consensus matrices for this ensemble was compared to those for the combinations of $R = (1, 101, 501, 1001, 5001, 10001)$, $S = (1, 100, 500, 1, 000)$ in the three datasets. We decide the ensemble is sufficiently deep and wide to stop growing if, for a given depth r and width s , there is no visible difference between the consensus matrices from the ensembles using $R = (ar, r)$, $S = (s, bs)$. In our analysis we used $a = b = 0.5$ (the smaller the choice of a, b the more extreme the stopping criterion). This heuristic is inspired by the belief that a clustering method should produce stable results across similar datasets (Von Luxburg and Ben-David, 2005; Meinshausen and Bühlmann, 2010). We believe that if the method is still producing a partition that is visibly changing for additional chains and depth, than the random initialisation is influencing the result sufficiently that it is unlikely to be stable for similar datasets or reproducible for a random choice of seeds (i.e. not stable within the current dataset). An example of this logic can be seen in figures 13 and 14 (and to a lesser degree in figure 12). Here the decision to stop growing the ensemble is made as there is no apparent gain in increasing chain depth from $R = 5001$ to $R = 10001$, but it can be seen that a chain depth of $R = 1001$ is insufficient as there is a marked difference in the consensus matrices for the PPI dataset particularly between $R = 1001$ and $R = 5001$. The number of chains appears required appears to have stabilised quickly, as there is no obvious change in increasing S from 100.

If we compare the distribution of sampled values for the ϕ parameters for the Bayesian chains that we keep based upon their convergence diagnostics, the final ensemble used ($R = 10001$, $S = 1000$) and the pooled samples from the 5 long chains, then we see that the ensemble consisting of the long chains (which might be believed to sampling different parts of the posterior distribution) is closer in its appearance to the distributions sampled by the Consensus clustering than to any single chain.

Within chain convergence

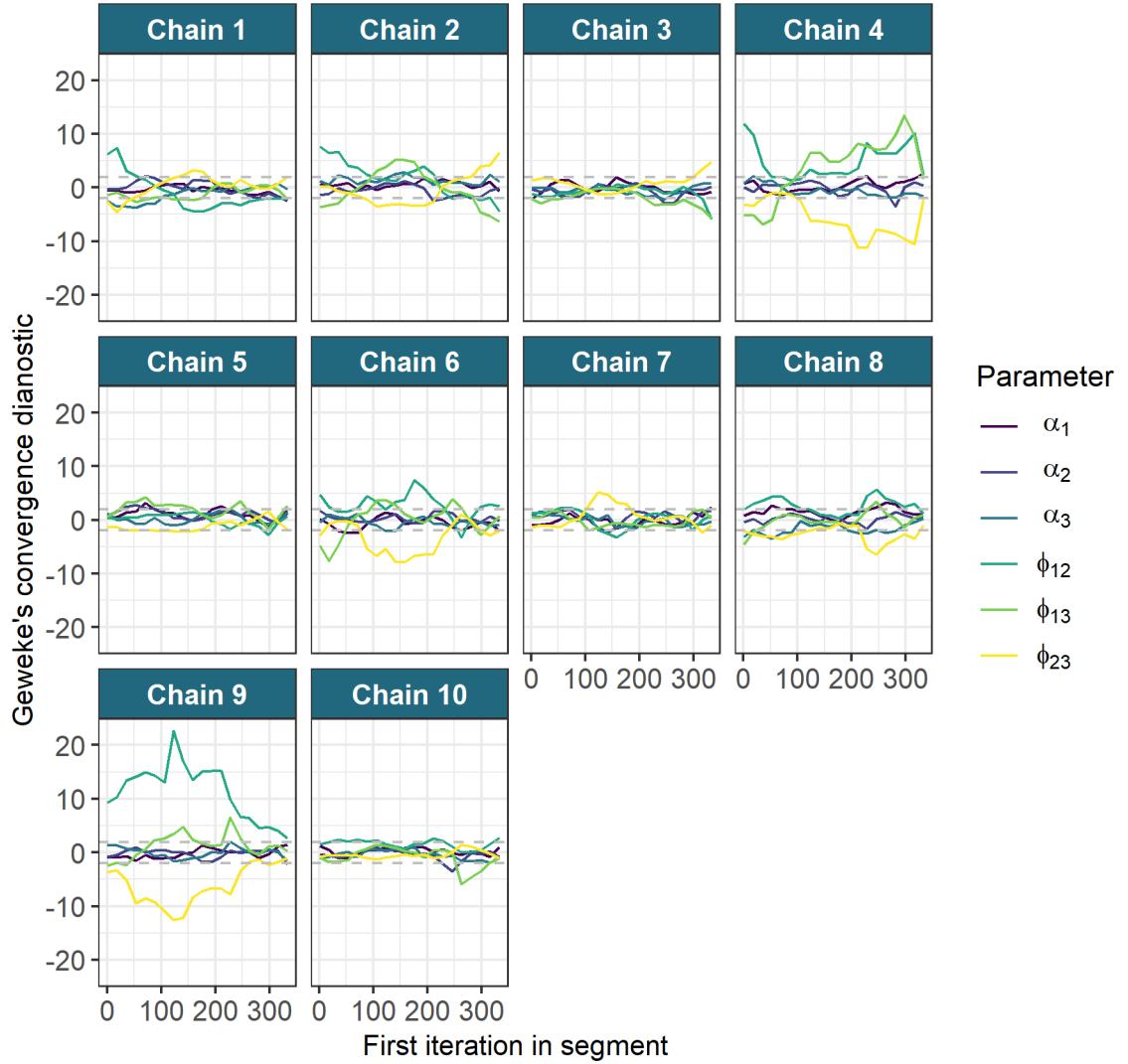


Figure 8: Chain 9 can be seen to have the most extreme behaviour in the distribution of the Geweke diagnostic for the parameters. We remove this chain from the analysis. Of the remaining chains we believe that 1, 2, 4 and 6 express the distributions furthest removed from the desired behaviour and are dropped from the analysis.

Gelman-R Rubin diagnostic plot

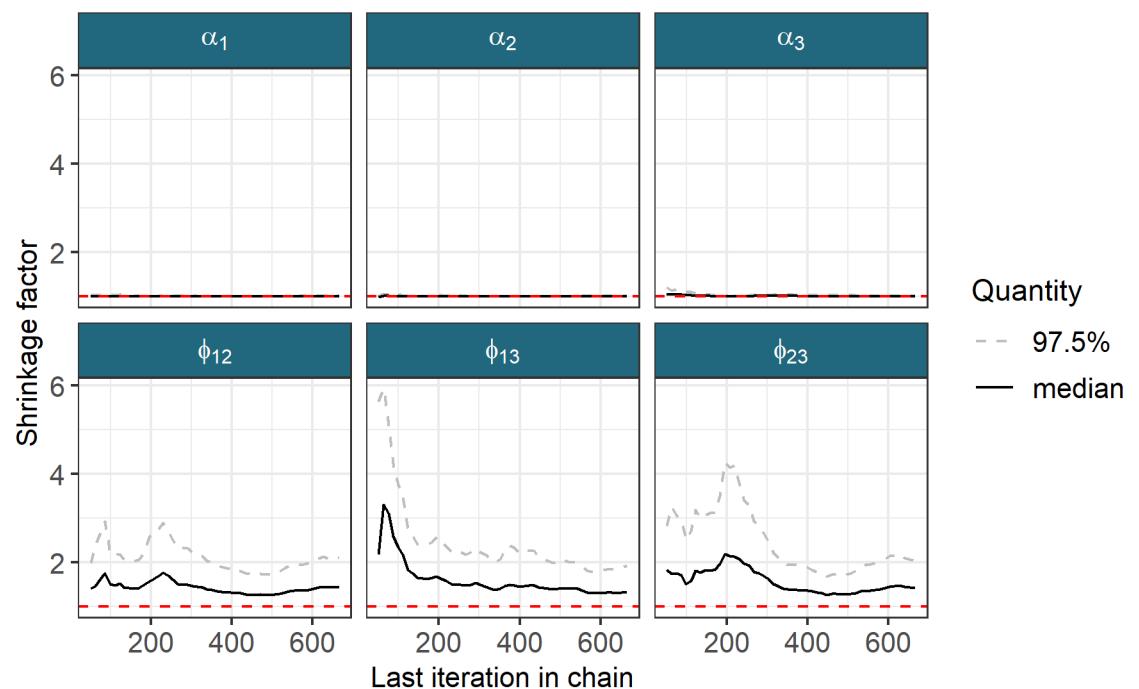


Figure 9: The chains still appear to be unconverged with \hat{R} remaining above 1.25 for the ϕ_{12}, ϕ_{13} and ϕ_{23} parameters. Stable \hat{R} is also too high with values of 1.049, 1.052 and 1.057.

Parameter density

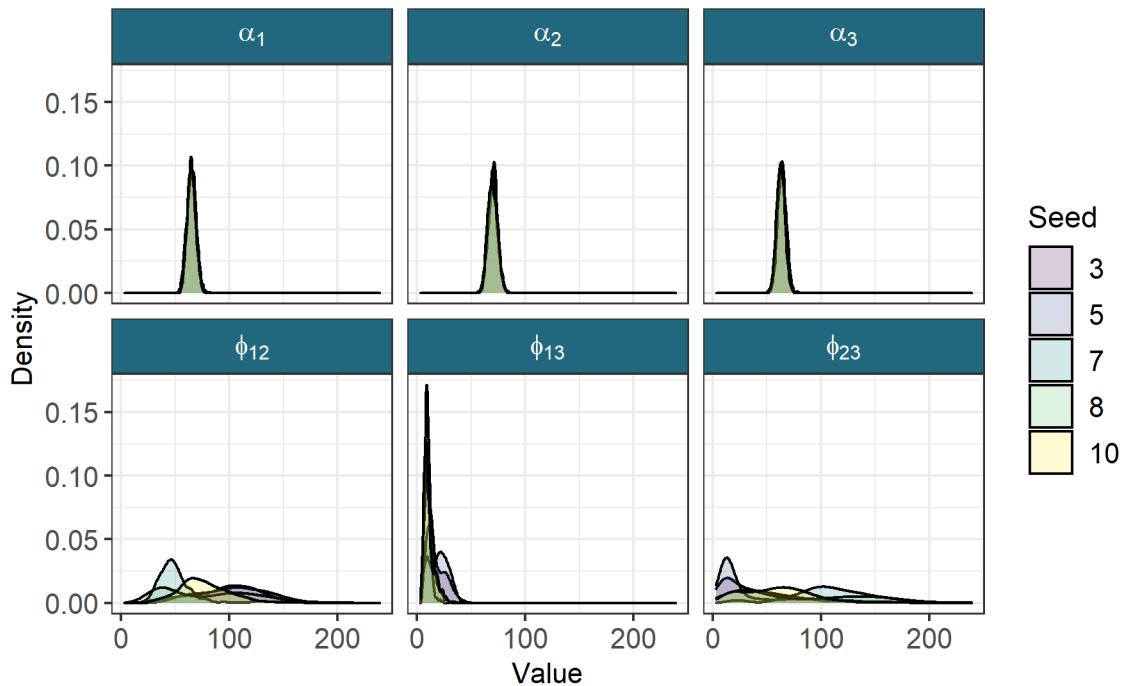


Figure 10: The densities of the continuous variables across the 5 chains kept for analysis. The mean sampled values are $\alpha_1 = 64.84$, $\alpha_2 = 69.85$, $\alpha_3 = 63.22$, $\phi_{12} = 81.76$, $\phi_{13} = 13.87$, and $\phi_{23} = 65.03$. It can be seen that different modes are being sampled for the ϕ parameters in each chain.

Yeast dataset

Posterior similarity matrices

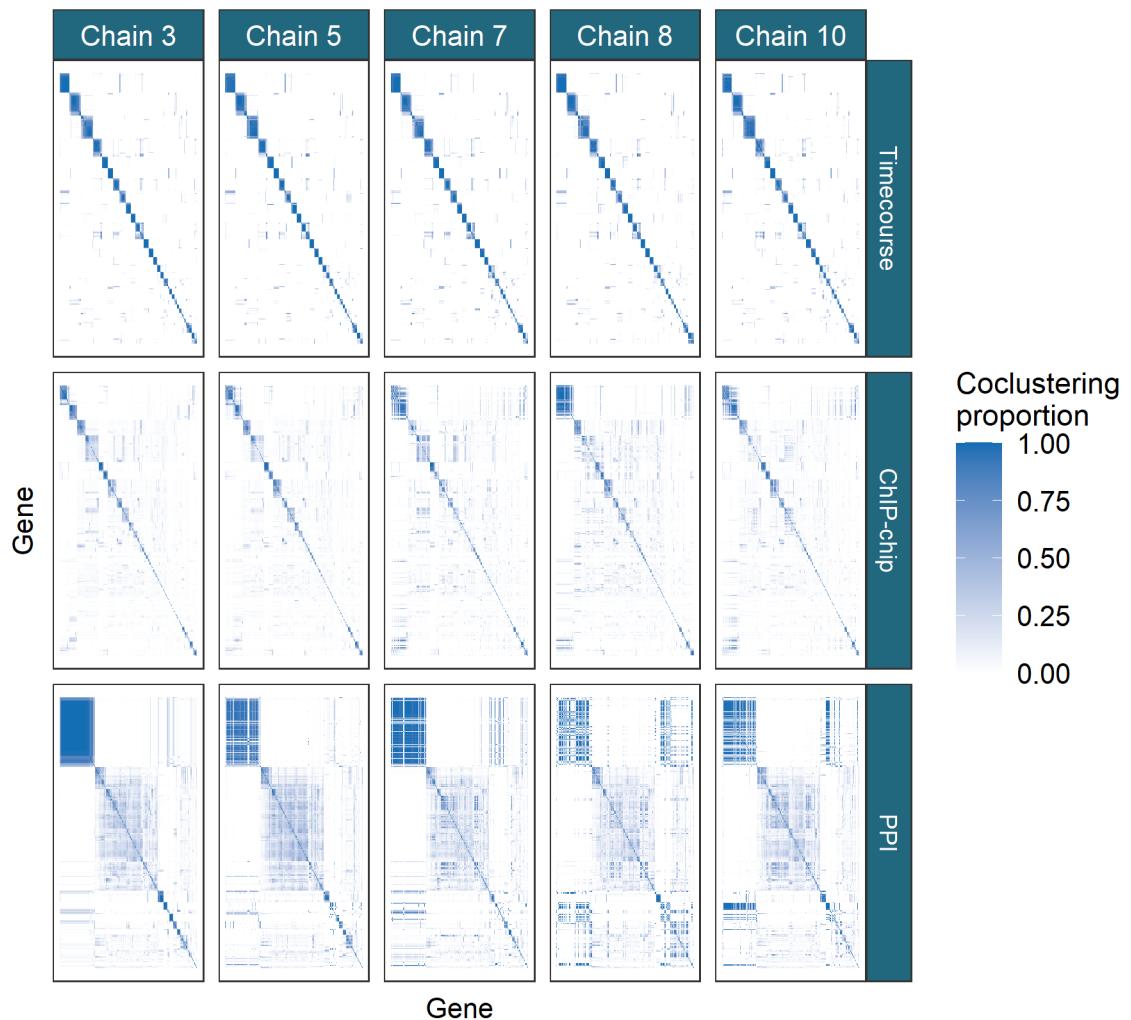


Figure 11: PSMs for each chain within each dataset. The PSMs are ordered by hierarchical clustering of the rows of the PSM for chain 3 in each dataset. There is no marked difference between the matrices for the Timecourse data with disagreement becoming more prominent in the ChIP-chip data and more so again in the PPI dataset.

Timecourse

Consensus matrices

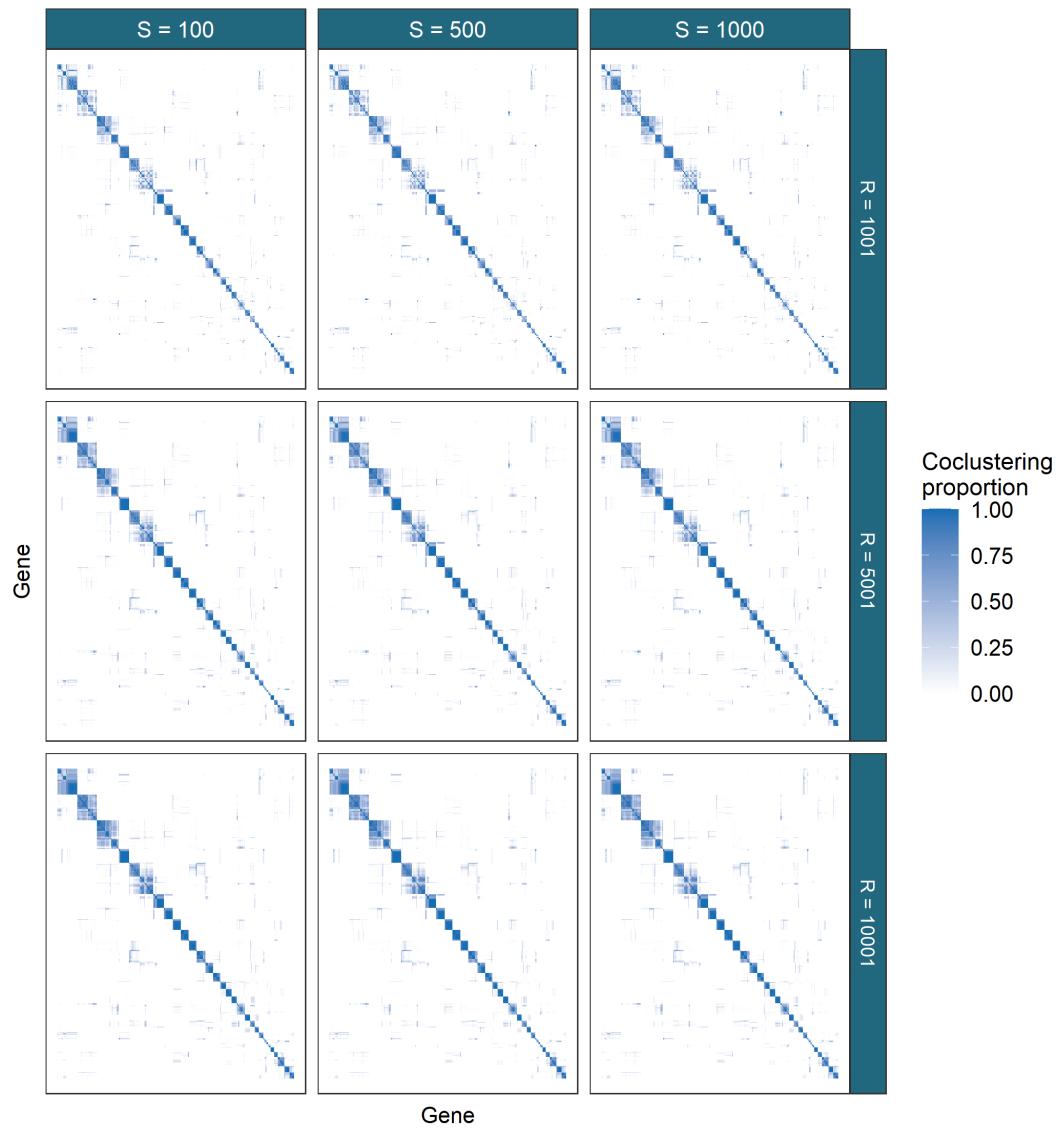


Figure 12: Consensus matrices for different ensembles of MDI for the Timecourse data. This dataset has stable clustering across the different choices of number of chains, S , and chain depth, R , with some components merging as the chain depth increases.

ChIP-chip

Consensus matrices

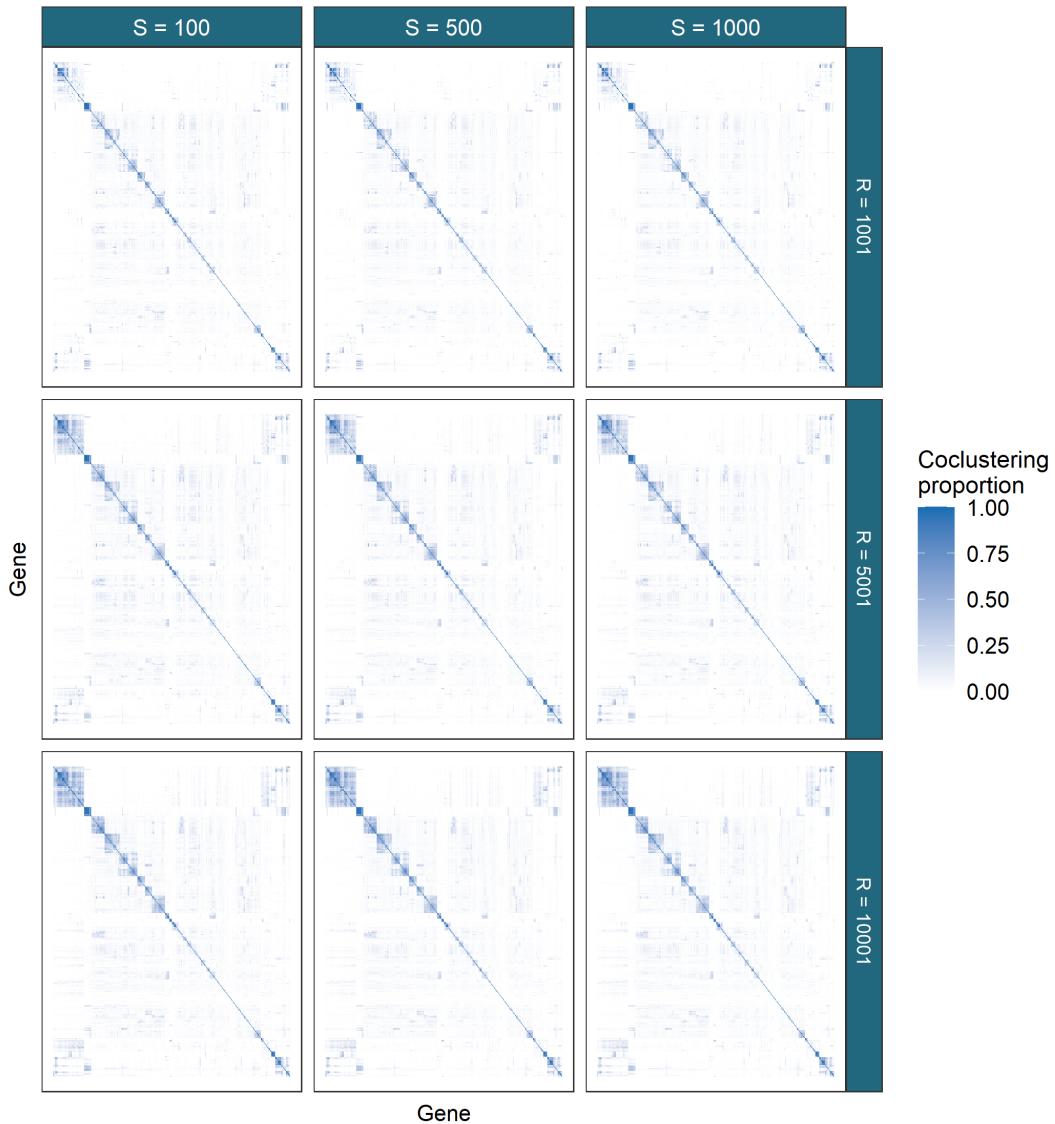


Figure 13: The ChIP-chip dataset is more sparse than the Timecourse data. In keeping with the results from the simulations for mixture models, deeper chains are required for better performance. It is only between $R = 5,001$ and $R = 10,001$ that no change in the clustering can be observed and the result is believed to be stable. In this dataset the number of chains used, S , appears relatively unimportant, with similar results for $S = 100, 500, 1000$.

PPI

Consensus matrices

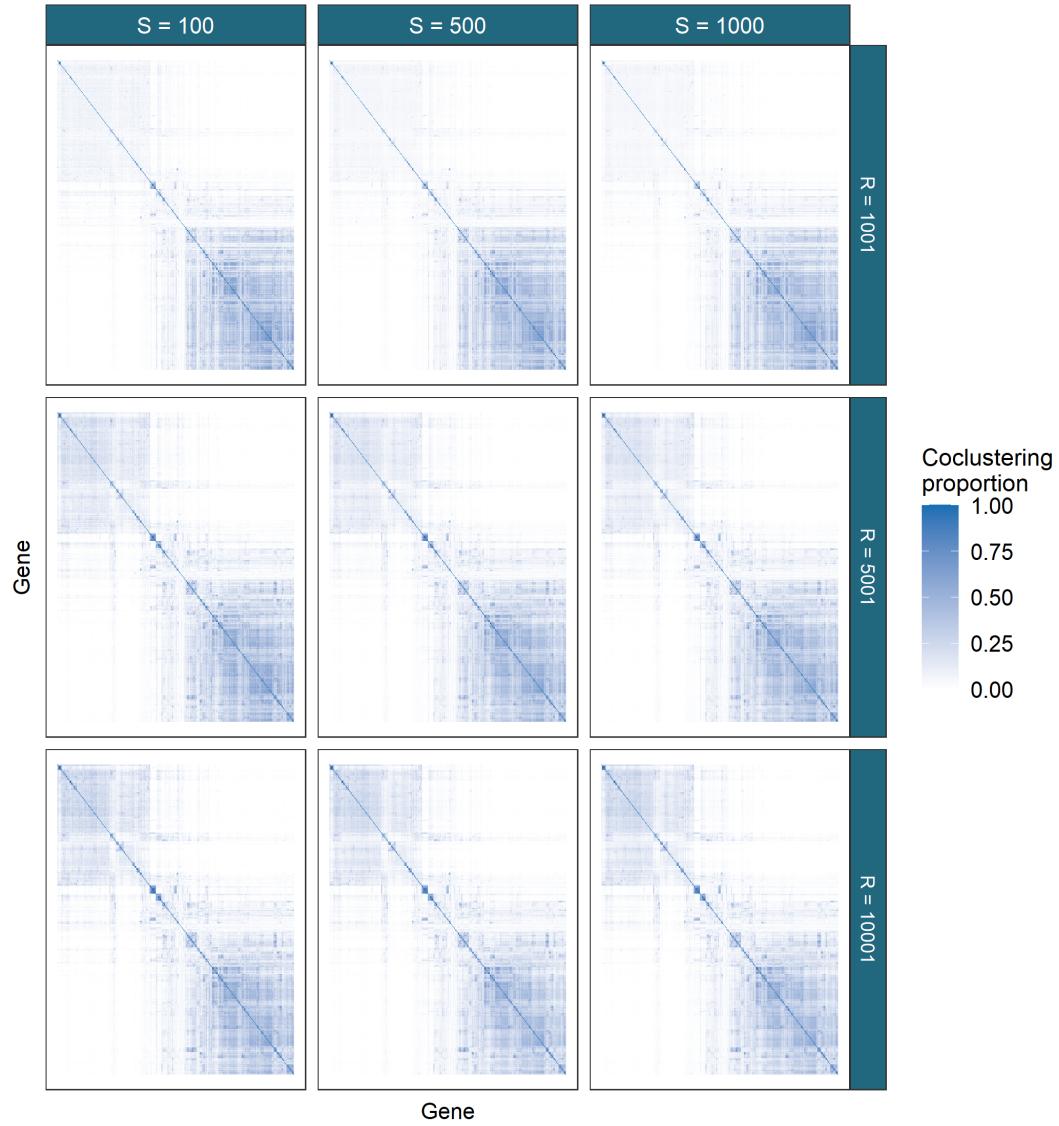


Figure 14: The PPI dataset has awkward characteristics for modelling. A wide, sparse dataset it is again chain depth that is the most important parameter for the ensemble. Similar to the results in figure 13, the matrices only stabilise from $R = 5001$ to $R = 10001$.

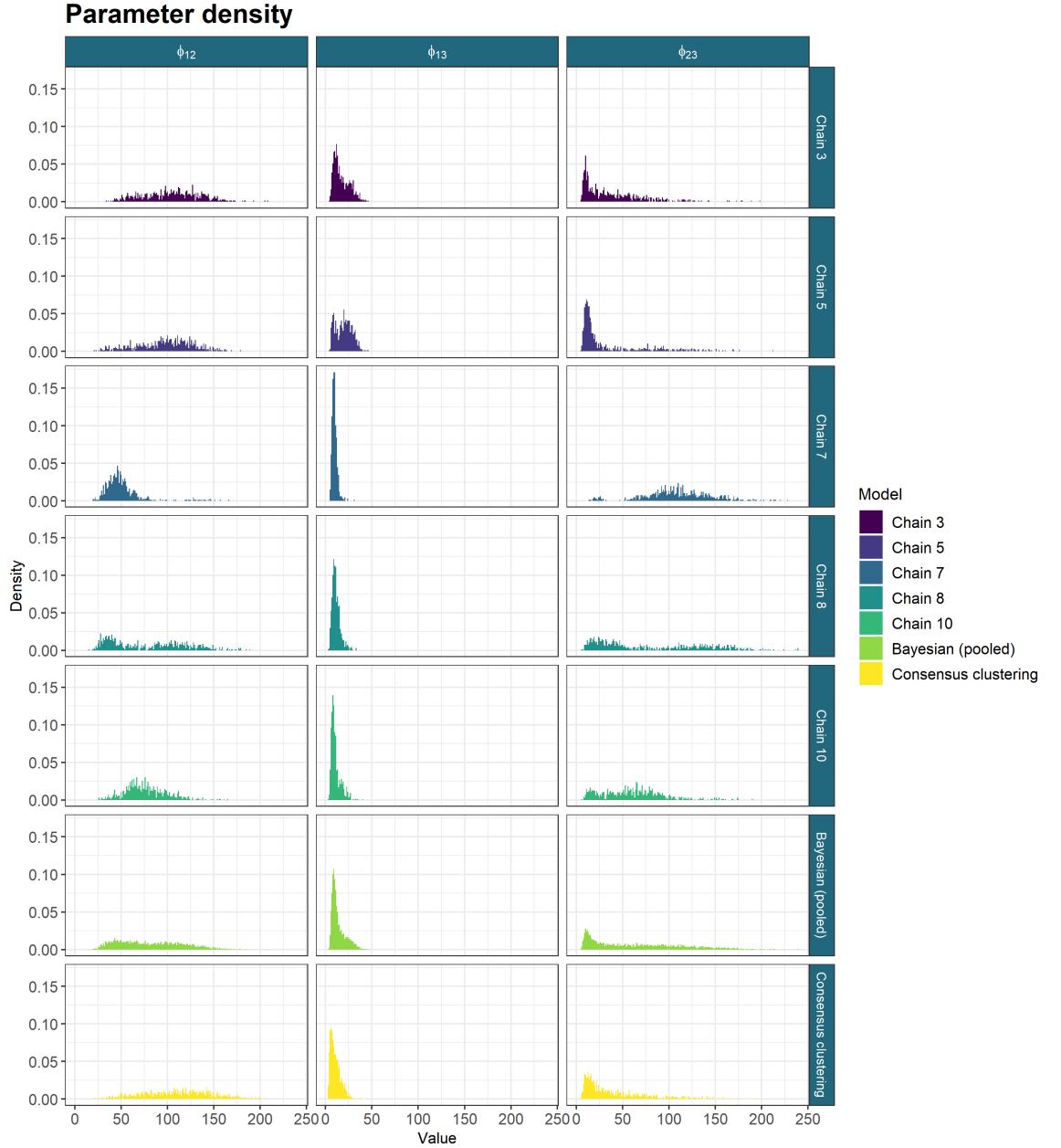


Figure 15: The sampled values for the ϕ parameters from the long chains, their pooled samples and the consensus using 1000 chains of depth 10,001. The long chains display a variety of behaviours. Across chains there is no clear consensus on the nature of the posterior distribution. The samples from any 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.

4.3 GO term over-representation

To validate the predicted clusters we test if they contain a higher concentration of specific Gene Ontology (GO) terms than would be expected by chance. Clusters were predicted from the PSMs of the chains kept from section 4.1 visualised in figure 11 and the consensus matrix of the largest ensemble run (i.e. $CC(10001, 1000)$) using the `maxpear` function from the R package `mcclust` Fritsch (2012) using default settings except for `k.max` which was set to 275 (the rounding down of $N/2$). To perform the GO term over-representation analysis we used the Bioconductor packages `clusterProfiler` (Yu et al., 2012), `biomaRt` (Durinck et al., 2009) and the annotation package `org.Sc.sgd.db` (Carlson et al., 2014).

We conditioned the test on the background set of the 551 yeast genes in the data. The gene labelled YIL167W was not found in the annotation database and was dropped from the analysis leaving a background universe of 550 genes. A hypergeometric test was used to check if the number of genes associated with specific GO terms within a cluster was greater than expected by random given the 550 possible genes. The false discovery rate of this test was controlled using the Benjamini-Hochberg correction (Benjamini and Hochberg, 1995) and significance threshold of 0.05 was used. The over-represented GO terms were then plotted to compare methods. The three different ontologies of “Molecular function” (**MF**), “Biological process” (**BP**) and “Cellular component” (**CC**) were all investigated. The over-represented terms are displayed in figures 16, 17 and 18 respectively. It can be seen that the Consensus clustering finds very similar results to the long chains, finding any term that is over-represented in each chain and a number of terms unique to itself. Very few terms found by a single chain are over-represented in the clustering from the ensemble.

References

- Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.
- M Carlson, S Falcon, H Pages, and N Li. Org. sc. sgd. db: Genome wide annotation for yeast. *R package version*, 2(1), 2014.
- Steffen Durinck, Paul T Spellman, Ewan Birney, and Wolfgang Huber. Mapping identifiers for the integration of genomic datasets with the r/bioconductor package biomart. *Nature protocols*, 4(8):1184, 2009.
- Yasin Şenbabaoğlu, George Michailidis, and Jun Z Li. A reassessment of consensus clustering for class discovery. *bioRxiv*, page 002642, 2014a.
- Yasin Şenbabaoğlu, George Michailidis, and Jun Z Li. Critical limitations of consensus clustering in class discovery. *Scientific reports*, 4(1):1–13, 2014b.

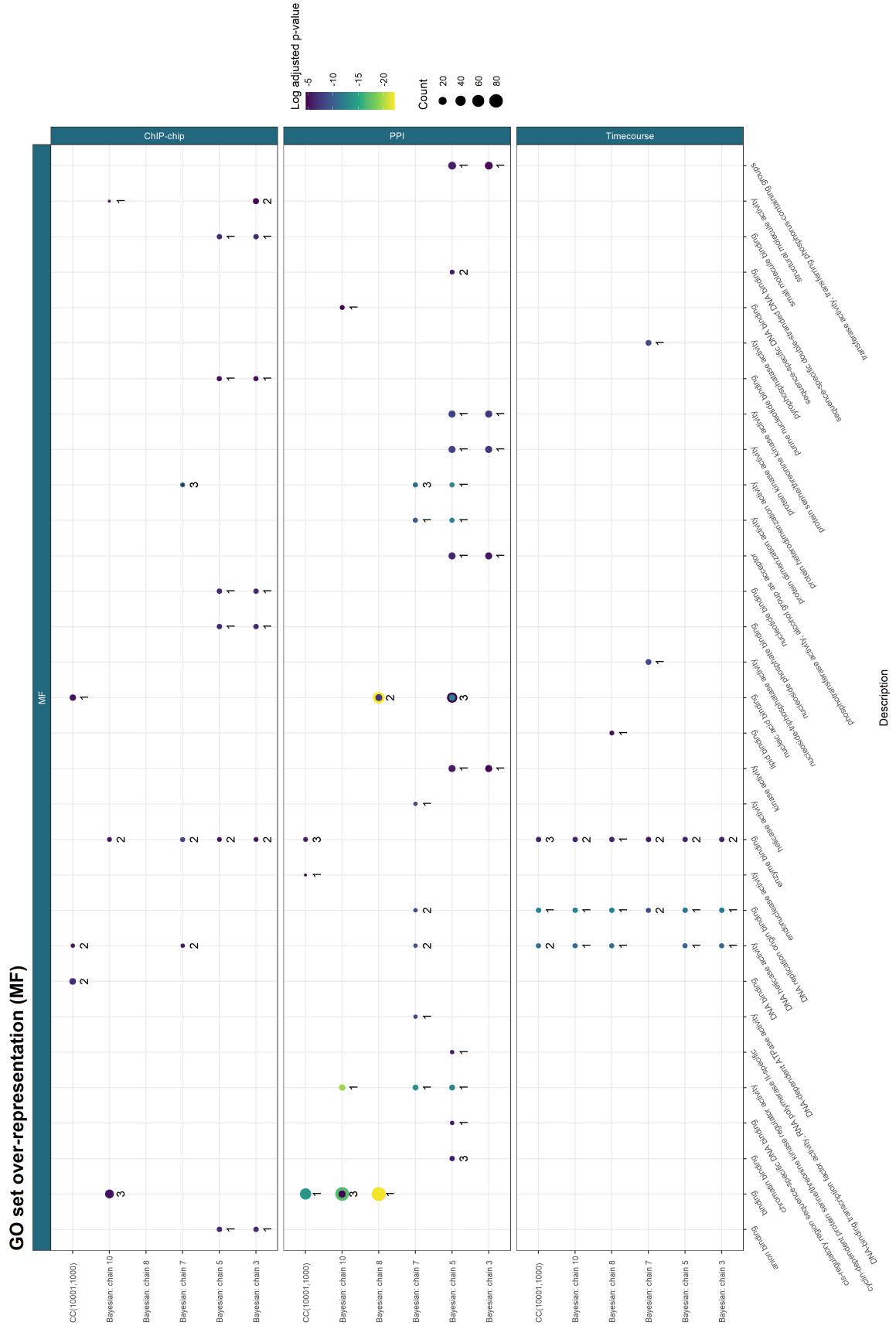


Figure 16:



Figure 17: .

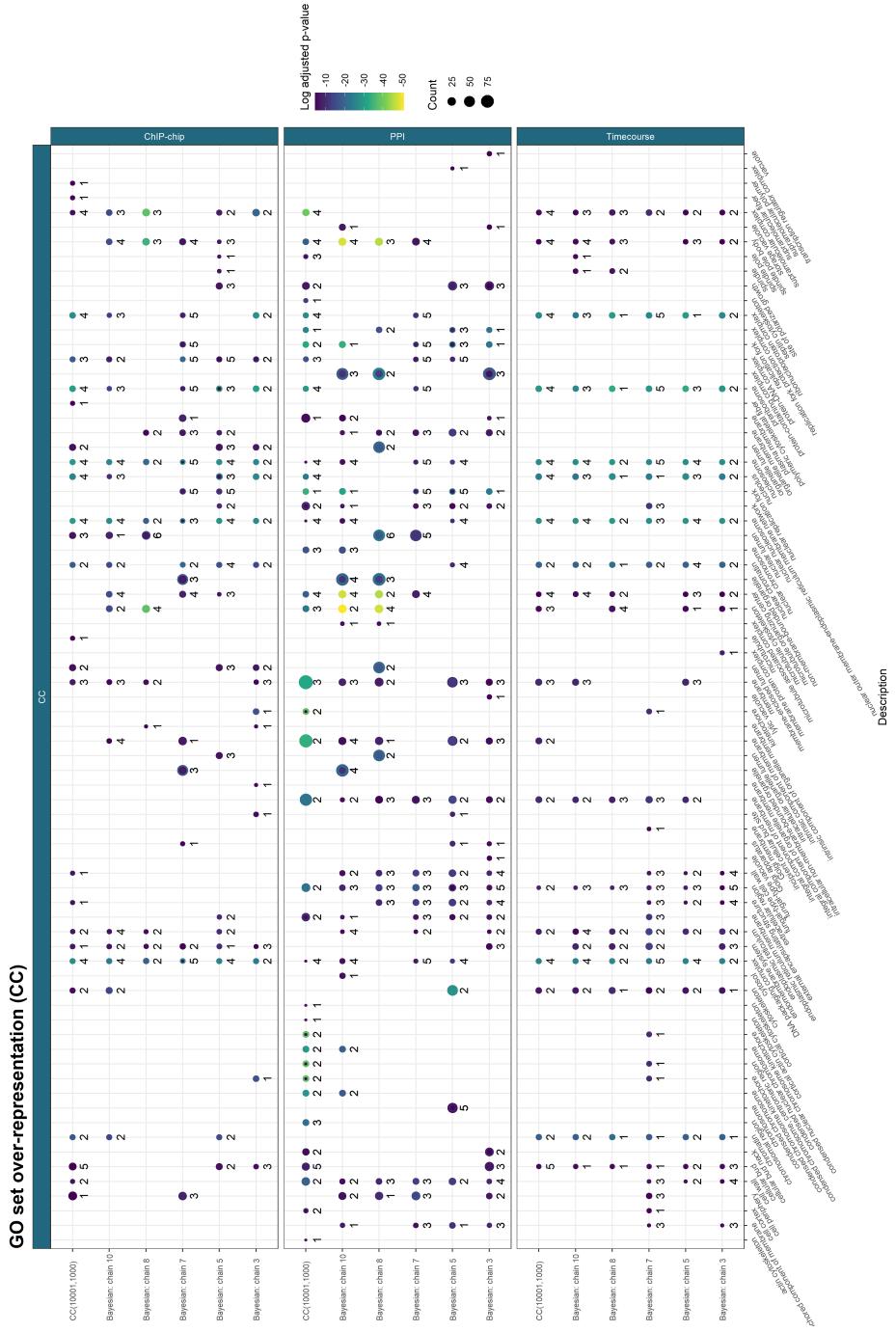


Figure 18:

- Arno Fritsch. *mcclust: Process an MCMC Sample of Clusterings*, 2012. URL <https://CRAN.R-project.org/package=mcclust>. R package version 1.0.
- Andrew Gelman, Donald B Rubin, et al. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.
- Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.
- John Geweke et al. *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*, volume 196. Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN, 1991.
- Marina V Granovskaia, Lars J Jensen, Matthew E Ritchie, Joern Toedling, Ye Ning, Peer Bork, Wolfgang Huber, and Lars M Steinmetz. High-resolution transcription atlas of the mitotic cell cycle in budding yeast. *Genome biology*, 11(3):1–11, 2010.
- Christopher T Harbison, D Benjamin Gordon, Tong Ihn Lee, Nicola J Rinaldi, Kenzie D Macisaac, Timothy W Danford, Nancy M Hannett, Jean-Bosco Tagne, David B Reynolds, Jane Yoo, et al. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, 2004.
- Paul Kirk, Jim E Griffin, Richard S Savage, Zoubin Ghahramani, and David L Wild. Bayesian correlated clustering to integrate multiple datasets. *Bioinformatics*, 28(24):3290–3297, 2012.
- Christina Knudson and Dootika Vats. *stableGR: A Stable Gelman-Rubin Diagnostic for Markov Chain Monte Carlo*, 2020. URL <https://CRAN.R-project.org/package=stableGR>. R package version 1.0.
- Samuel A Mason, Faiz Sayyid, Paul DW Kirk, Colin Starr, and David L Wild. Mdi-gpu: accelerating integrative modelling for genomic-scale data using gpu computing. *Statistical Applications in Genetics and Molecular Biology*, 15(1):83–86, 2016.
- Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.
- Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
- Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.
- Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. Biogrid: a general repository for interaction datasets. *Nucleic acids research*, 34(suppl_1):D535–D539, 2006.

Dootika Vats and Christina Knudson. Revisiting the gelman-rubin diagnostic.
arXiv preprint arXiv:1812.09384, 2018.

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

Guangchuang Yu, Li-Gen Wang, Yanyan Han, and Qing-Yu He. cluster-profiler: an r package for comparing biological themes among gene clusters. *OMICS: A Journal of Integrative Biology*, 16(5):284–287, 2012. doi: 10.1089/omi.2011.0118.