Consensus clustering for Bayesian mixture models: Supplementary materials

Stephen Coleman, Paul DW Kirk and Chris Wallace

October 20, 2020

Abstract

Description of models used and analyses performed.

1 Definitions

Definition 1 (Consensus matrix) *Given S clusterings for a dataset of N items, $c_s = (c_{s1}, \ldots, c_{sN})$, 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*

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

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*.*

Definition 3 (Partition *or* Clustering) *For a dataset $X = (X_1, \ldots, X_N)$, a* partition *or* clustering *is a set of disjoint sets covering X, normally indicated by a N-vector of integers indicating which set each item is associated with. Note that these labels only have meaning relative to each other, they are symbolic. Each set within the clustering is referred to as a* cluster*.*

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 inde-

pendent items $X = (x_1, \ldots, x_N)$ as

$$x_n \sim \sum_{k=1}^{K} \pi_k f(x_n|\theta_k) \qquad \text{independently for } n = 1, \ldots, N \tag{2}$$

where $f(\cdot|\theta)$ is some family of densities parametrised by θ. A common choice is the Gaussian density function, with $\theta = (\mu, \sigma^2)$ (as in our simulation study). K, the number of subgroups in the population, $\{\theta_k\}_{k=1}^{K}$, the component parameters, and $\pi = (\pi_1, \ldots, \pi_K)$, the component weights are the objects to be inferred. In the context of *clustering*, such a model arises due to the belief that the population from which the random sample under analysis has been drawn consists of K unknown groups proportional to π. In this setting it is natural to include a latent *allocation variable*, $c = (c_1, \ldots, c_N)$, to indicate which group each item is drawn from, with each non-empty component of the mixture corresponds to a cluster. The model is

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

The joint model can then be written

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

We assume conditional independence between certain parameters such that the model reduces to

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

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

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

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

$$\pi \sim \text{Dirichlet}(\alpha). \tag{7}$$

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

$$\alpha = 1, \tag{8}$$

$$\xi = 0.0, \tag{9}$$

$$\lambda = 1.0, \tag{10}$$

$$a = 2.0, \tag{11}$$

$$b = 2.0. \tag{12}$$

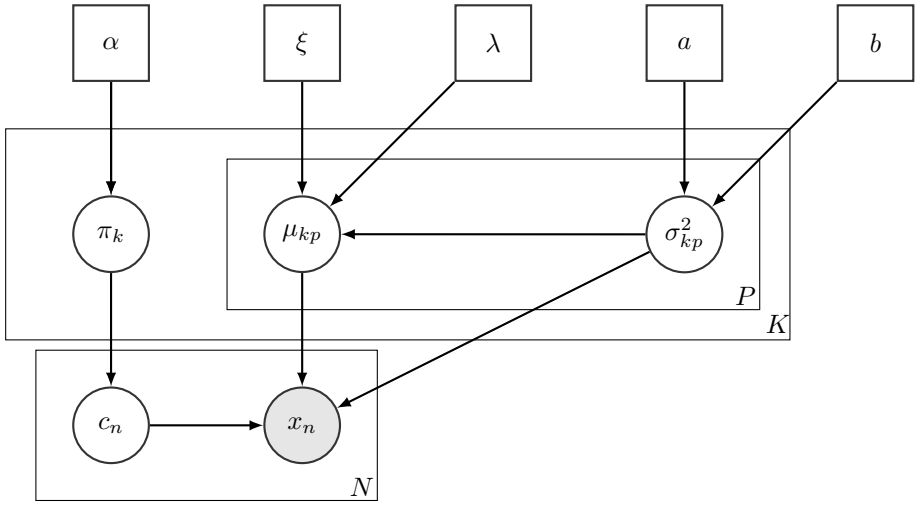

Figure 1: Directed acyclic graph for the mixture of Gaussians used.

2.2 Integrative clustering

We are interested in the use of Consensus clustering for integrative methods. We used Multiple Dataset Integration (**MDI**, Kirk et al., 2012) as an example of a Bayesian integrative clustering method. MDI models dataset specific clusterings, in contrast to, for example, Clusternomics (Gabasova et al., 2017) in which a *global clustering* is inferred.

The defining aspect of MDI is the prior on the allocation of the n^{th} item across the L datasets

$$p(c_{n1}, \ldots, c_{nL}) \propto \prod_{l=1}^{L} \pi_{c_{nl}l} \prod_{l=1}^{L-1} \prod_{m=l+1}^{L} (1 + \phi_{lm} \mathbb{I}(c_{nl} = c_{nm})) \text{ for } n = 1, \ldots, N. \tag{13}$$

ϕ_{lm} is the parameter defined by the similarity 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 $\phi_{lm} \to 0$ we have independent mixture models. In other words, MDI allows datasets with similar clustering of the items to inform the clustering in each other more strongly than the clustering for an unrelated dataset. The DAG for this model for three datasets is shown in figure 2.

3 Simulations

We defined a number of scenarios to test certain concepts of the method and to explore behaviour due to specific characteristics of real data. The parame-

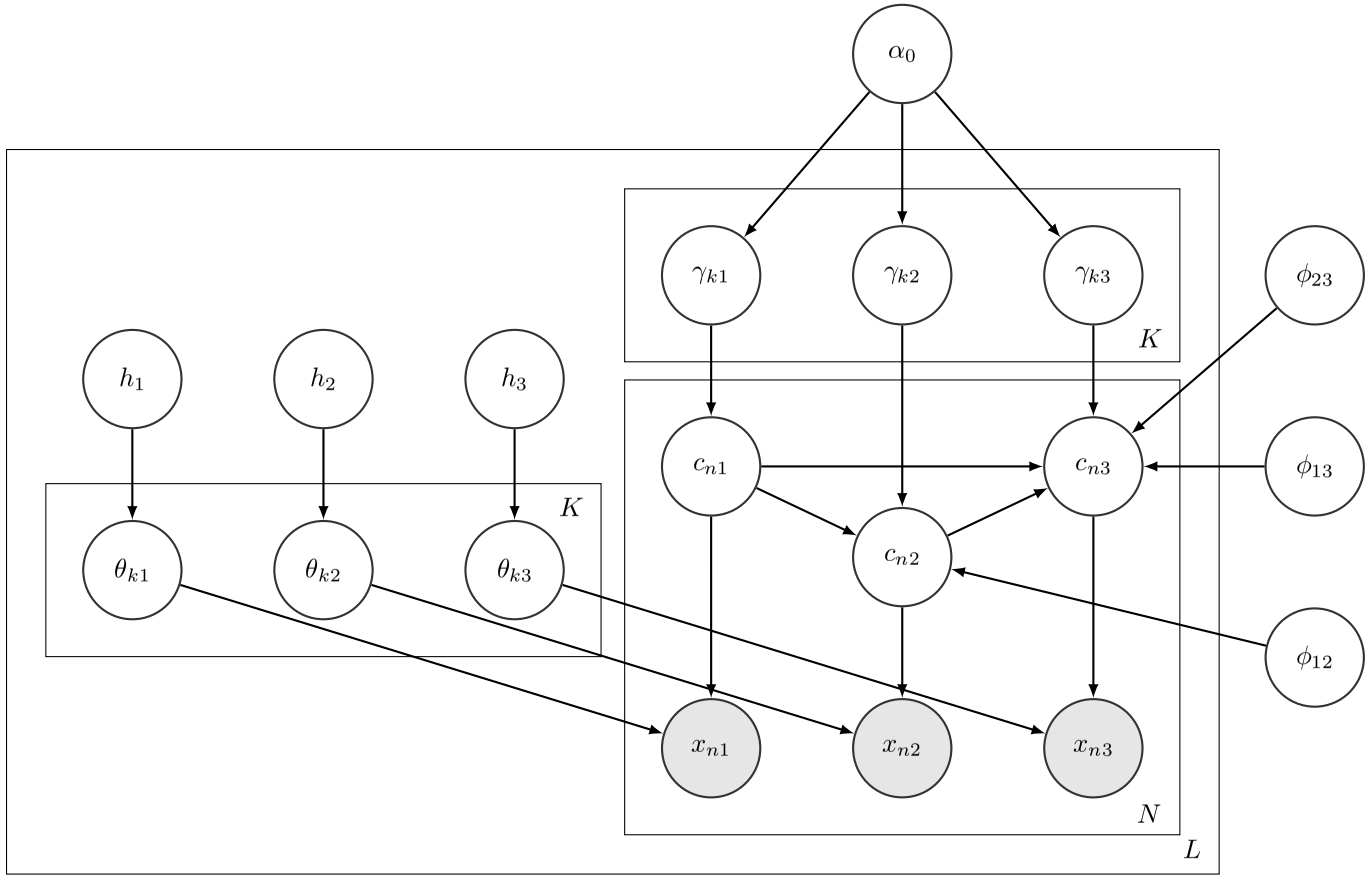

Figure 2: Directed acyclic graph for the Multiple Dataset Integration model for $L = 3$ datasets. h_l is the choice of hyperpriors for the l^{th} dataset.

ters associated with each scenario in table 1 were used to generate individual simulations using algorithm 1.

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.

- *2D*: a low dimensional scenario within which we expected `Mclust` to perform well and the long chains to converge and explore the full support of the posterior distribution.
- *No structure*: we included this scenario to reassure fears that Consensus clustering has a predilection to finding clusters where none exist (Şenbabaoğlu et al., 2014a,b).
- *Base case*: highly informative datasets within which we expected methods to find the true generating labels quite easily. We included this scenario to benchmark the others that are variations of this setting.
- *Large standard deviation*: these two scenarios investigated the degree of distinction required between clusters for the methods to uncover their structure.
- *Irrelevant features*: we included these scenarios to investigate how robust the methods are to irrelevant features.
- *Varying proportions*: these scenarios investigated 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.

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 $\phi = (\phi_1, \ldots, \phi_P)$
The expected proportion of items in each cluster $\pi = (\pi_1, \ldots, \pi_K)$
A method for sampling x times from the array y, with weights π: $Sample(y, x, \pi)$
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 $c = (c_1, \ldots, c_N)$

```
begin
    /* initialise the empty data matrix                    */
    X ← 0_{N×P};
    /* create a matrix of K means                          */
    μ ← (Δ_μ, ..., KΔ_μ);
    /* generate the allocation vector                      */
    c ← Sample(1 : K, N, π);
    M ← 0_{N×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 φ_p = 1 then
            ν ← Permute(μ);
            for n = 1 to N do
                X(n, p) ← Gaussian(ν_{c_n}, σ^2)
            end
        end
        if φ_p = 0 then
            for n = 1 to N do
                X(n, p) ← Gaussian(0, σ^2)
            end
        end
    end
    /* Mean centre and scale the data                      */
    X ← 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`.

3.1 Bayesian analysis

For each simulation we ran 10 chains for 1 million iterations, keeping every thousandth sample. We discarded the first 10,000 iterations to account for burn-in bias, leaving 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 focused upon stationarity of the continuous variables as assesing convergence of the allocation labels is difficult due to label-switching. 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 PSM. This and the point estimate clustering found by applying the R function `maxpear` (Fritsch, 2012) to this PSM are used in model performance analysis in section 3.4. `maxpear` attempts to find the clustering that maximises the Adjusted Rand Index to the true clustering by using an approximation of the expected clustering under the posterior, $\mathbb{E}(c|X)$, believing that this converges to the true clustering. A sample average clustering is used to approximate the

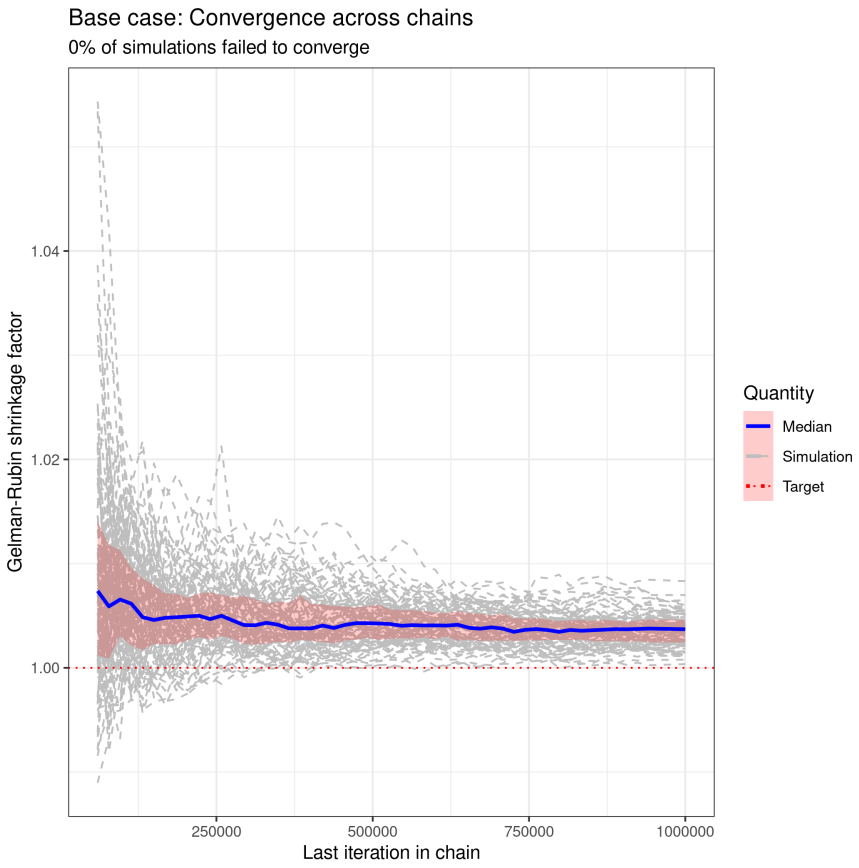

Figure 3: The $\hat{R}$ values for each simulation (in dotted grey), the median value and the interquartile range across simulations. One can see that $\hat{R}$ approaches 1.0, being below 1.01 for every simulation by the end of the chains. The "0% of simulations failed to converge" is a statement based upon the percentage of simulations which passed the test of stable $\hat{R}$.

expected clustering. This is estimated from the PSM by maximising

$$\frac{\sum_{i<j} \mathbb{I}(c_i^* = c_j^*) p_{ij} - \sum_{i<j} \mathbb{I}(c_i^* = c_j^*) \sum_{i<j} p_{ij} / \binom{N}{2}}{\frac{1}{2}\left[\sum_{i<j} \mathbb{I}(c_i^* = c_j^*) + \sum_{i<j} p_{ij}\right] - \sum_{i<j} \mathbb{I}(c_i^* = c_j^*) \sum_{i<j} p_{ij} / \binom{N}{2}} \tag{14}$$

where p_{ij} is the $(i,j)^{th}$ entry of the PSM (Fritsch et al., 2009). When the chain has converged this maximises the posterior expected ARI to the true clustering.

There are three possibilities to consider the decision to pool the samples across chains 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 (see figure 6 for an example of this behaviour).

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, chains and samples would be required. In our simulations this case only arises in the most pathological form in the second *Large N, small P* scenario, where each chain remains trapped in the initial partition. The clustering inferred from any chain is not meaningful being a random clustering; thus the clustering predicted by pooling the PSMs is no more or less relevant as it too is random.

3.2 Consensus clustering analysis

We investigated a range of ensembles, using all combinations of chain depth, $R = (1, 10, 100, 1000, 10000)$, and the number of chains, $S = (1, 10, 30, 50, 100)$. This gave 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. An example of the Consensus matrices for each ensemble in a given simulation is shown in figure 7. We used the `maxpear` function from the R package `mcclust` to create a point clustering estimate from the Consensus matrix. In this context where we do not assume that the Consensus matrix of the samples is the Posterior similarity matrix we do not expect that the predicted clustering maximises the posterior expected ARI. Instead `maxpear` is used as calculating a sample average clustering which we believe is representative of the ensemble.

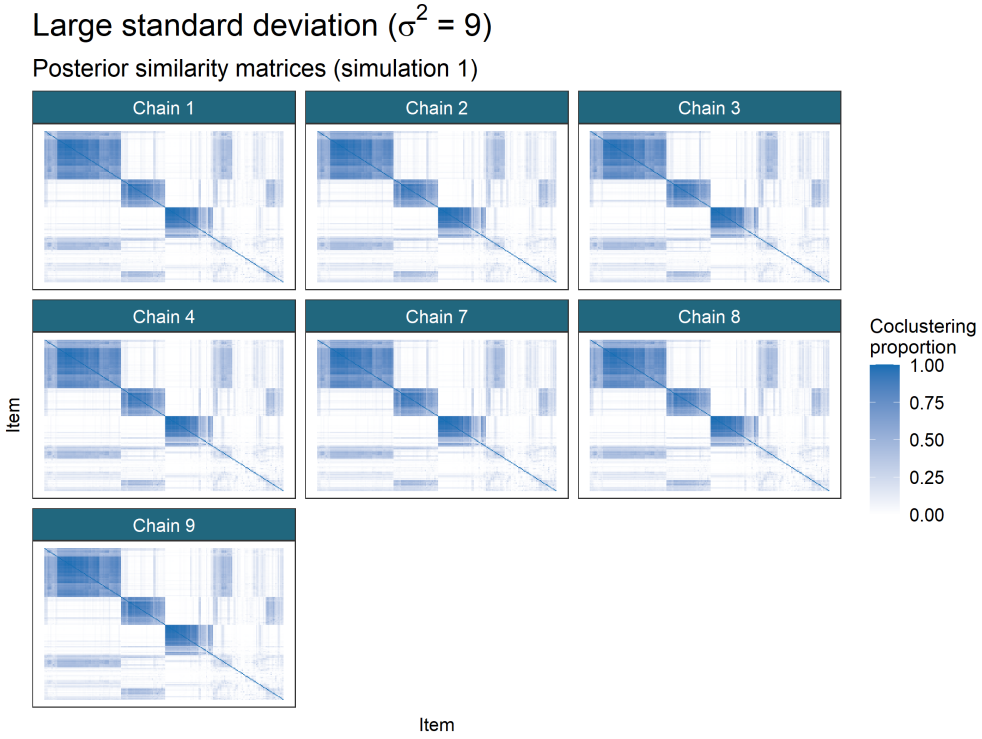

Figure 4: Posterior similarity matrices for the simulation generated using a random seed set to 1 for the first large standard deviation scenario from table 1. This is an example of all stationary chains agreeing in a simulation (and thus pooling of samples is no different to using any choice of chain 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 1.

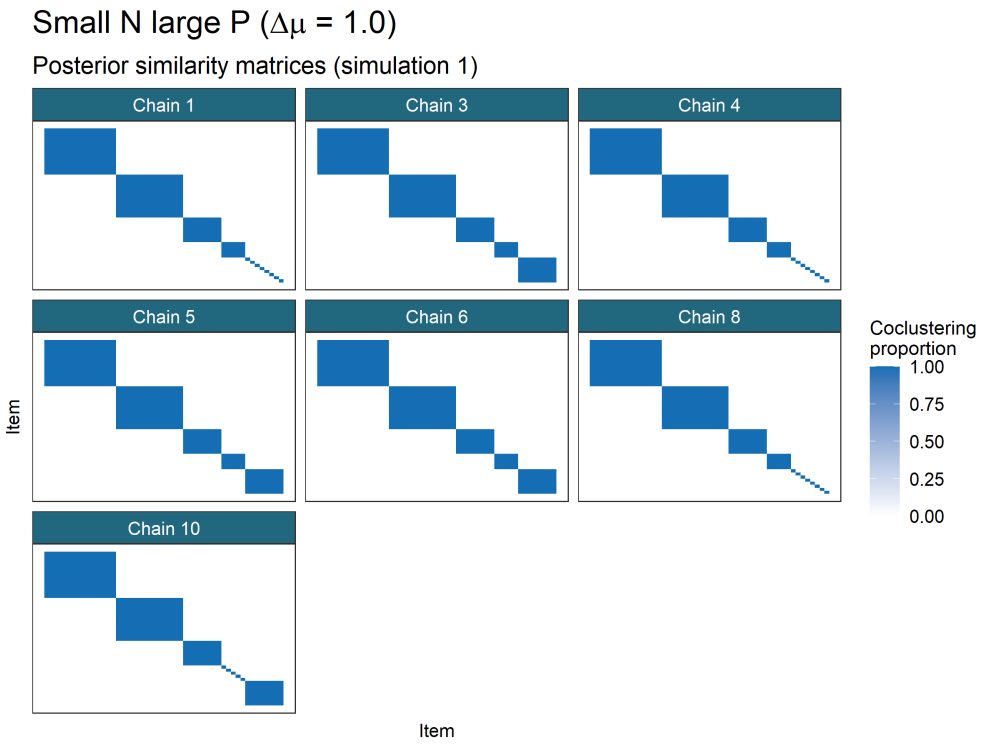

Figure 5: Posterior similarity matrices for the simulation generated using a random seed set to 1 for the first small N, large P scenario from table 1. This is an example of different chains becoming trapped in different modes, but one mode (which does represent the generating structure well) is dominant, 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 1.

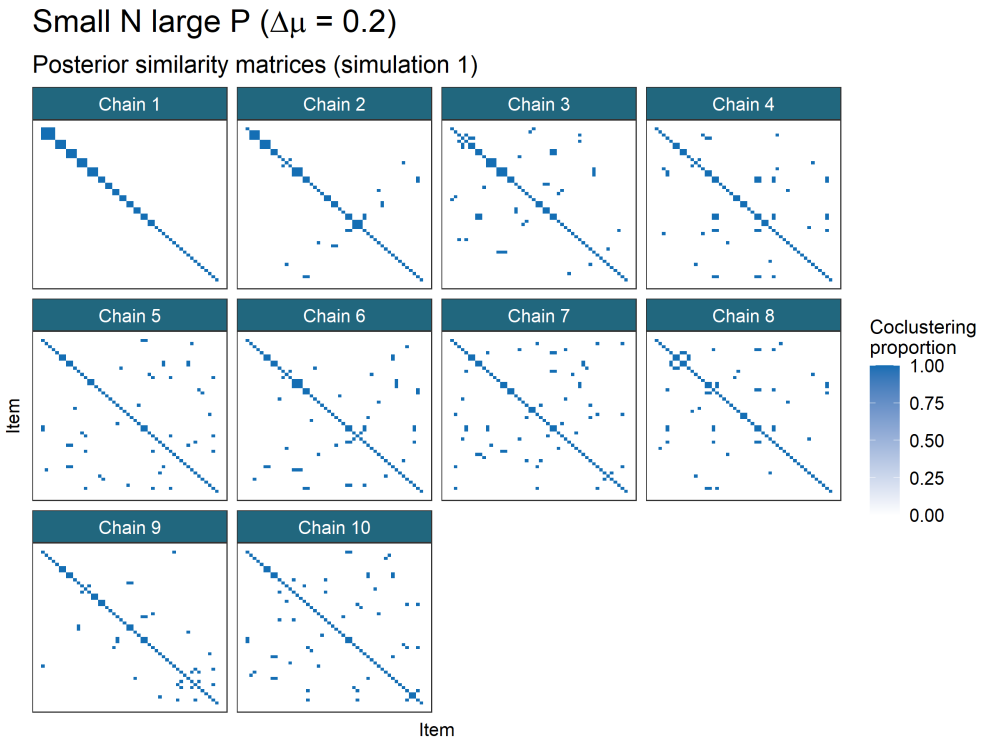

Figure 6: Posterior similarity matrices for the simulation generated using a random seed set to 1 for the second small N, large P scenario from table 1. This is an example of different chains becoming trapped in different modes with no mode being dominant. In this scenario each chain remains trapped in initialisation. Ordering of rows and columns is defined by hierarchical clustering of the first matrix in the series, in this case that from Chain 1.

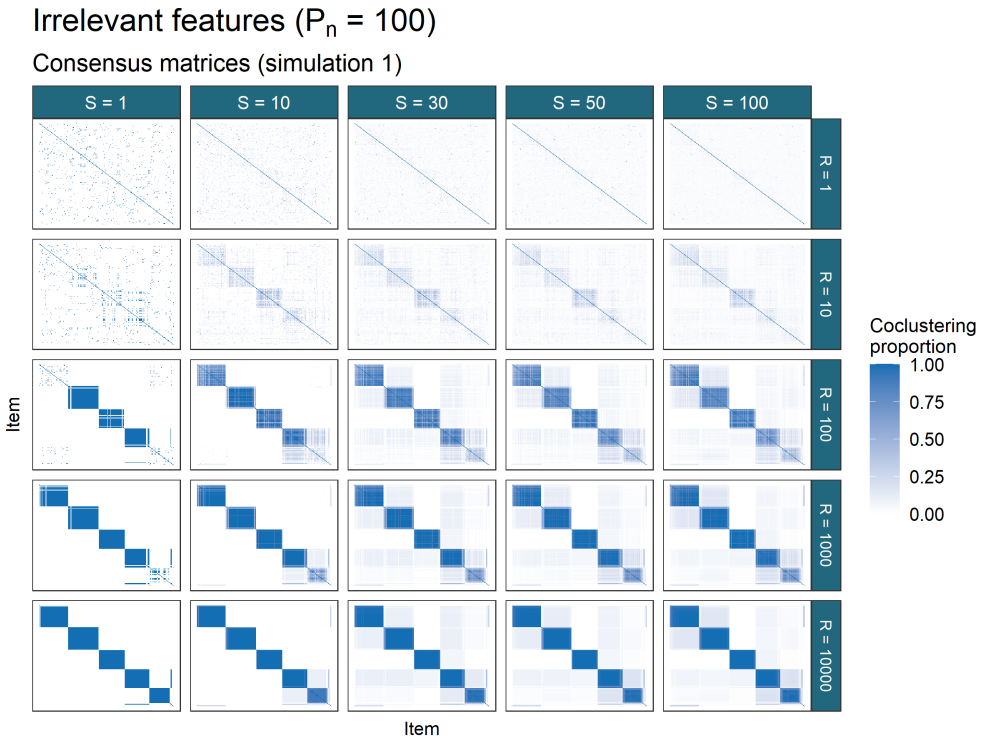

Figure 7: Consensus matrices for the simulation generated using a random seed set to 1 for the third irrelevant features scenario from table 1. R is the individual chain length and S is the number of chains used. In this example there are several modes present (as seen in the entries with values between 0 and 1) but one mode is clearly dominant (the 5 dark squares along the diagonal which correspond closely to the generating labels).

3.3 `Mclust`

We called `Mclust` using the default settings and a range of inputs for the choice of K. We used $K = (2, \ldots, \min(\frac{N}{2}, 50))$ to mirror the choice of $K_{max} = 50$ used for the overfitted mixture models (the default in the software we used), with the bound of $\frac{N}{2}$ to avoid fitting 50 clusters in the *Small N, large P* scenario where $N = 50 = K_{max}$. In the *No structure* scenarios we extended to range to $K = (1 \ldots, 50)$ to include the correct structure as an option. The model choice was 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.

In figure 11 the ARI between the generating labels and the point estimate clustering from each method is shown. For two partitions c_1, c_2,

- $ARI(c_1, c_2) = 1.0$: a perfect match between the two partitions,
- $ARI(c_1, c_2) = 0.0$: c_1 is no more similar to c_2 than is expected for a random partition of the data.

In several scenarios `Mclust` performs the best under this metric (e.g. in the scenarios *2D*, *Small N, large P* ($\Delta\mu = 0.2$)). However when the number of irrelevant features is large `Mclust` performs less well (see *Irrelevant features* ($P_n = 20$) and ($P_n = 100$)) than the other methods. The initialisation used by `Mclust` is based upon a hierarchical clustering of the data. We suspect that this contributes to the better performance in the *Small N, large P* ($\Delta\mu = 0.2$) case and the poor performance in the presence of large numbers of irrelevant features.

The pooled Bayesian samples act as an upper bound on the Consensus clustering ensembles in these simulations.

For the ensembles there are two parameters changing between each model, the iteration used to provide the clustering in the ensemble, R, and the number of chains (and hence samples) used, S. In many of the scenarios we find that the benefit of increasing R stabilises by approximately $R = 10$. We believe that in a low-dimensional dataset (such as *2D*), or a highly informative dataset (such as *Base case* or any of the higher dimensional scenarios with no irrelevant features where $\frac{\Delta\mu}{\sigma^2} \geq 1$) the chains quickly find a "sensible" partition of the data and thus increasing the depth within the chain does not increase the probability that any partition sampled will be closer to the generating partition. For example in figure 11 in the *Small N, large P* case, the distribution of the ARI across the ensembles for which $R \geq 10$ and $S = 1$ is nearly identical; this suggests that the chain is sampling a very similar partition again and again for 990,000 iterations (and possibly beyond based upon the PSMs shown in figure 5) and it is through

adding more chains rather than using particularly long chains that we improve the ability to uncover the generating structure.

We also notice that even if the behaviour has not stabilised for R that the ensemble can uncover meaningful structure. The ARI for the ensembles of short chains can be quite high (as is the case in many of the scenarios). The behaviour of the Consensus matrices also shows that low R is not a disqualifier from meaningful inference even if longer chains would be ideal. Consider the Consensus matrices in figure 7, it can be seen that the behaviour has not stabilised before $R = 10000$ (and possibly there is still some benefit in increasing R), but the structure being uncovered when there is a sufficient number of chains and R is small does correspond to the structure uncovered in the largest and deepest ensemble. We believe that the order in which components merge and items are co-clustered varies depending on initialisation, and thus if the chain is not sufficiently deep that all of the final mergings have occured that a sufficiently large ensemble can still perform meaningful inference of the subpopulation structure. Even though each learner probably has too many clusters for small R the consensus among them will have less. This is why the entries of the Consensus matrix for $R = 100$ and $S = 100$ in figure 7 are more pale than in deeper ensembles; very few items correctly (possibly none) cocluster in every partition, it is only in observing the consensus that the global structure of interest emerges. Thus if there is some limit to the length of chains available for an analysis (e.g. computational or temporal constraints) than the inference obtained from the shorter chains can still be meaningful, with the caveat that the point clustering might have more clusters than the same analysis with longer chains would provide. Additional post-hoc merging of some clusters might be necessary in this case.

In contrast, when the dataset is sparse or contains many irrelevant features, we believe that deeper chains are required to reach this steady-state sampling where no single sample is expected to be better than any other (see the *Irrelevant features ($P_n = 100$)* facet of figure 11).

In some scenarios no method is successful in uncovering the generating labels. In the *Large standard deviation ($\sigma^2 = 25$)* and *Small N, large P ($\Delta\mu = 0.2$)* this is due to the lack of signal - the clusters overlap so significantly that it is not possible for any of these methods to uncover much of the generating structure. In the *No structure* case it is different. In this case all items are generated from a common distributions. For the Bayesian chains and the ensembles, a clustering of singletons is predicted; each item is allocated a unique label (see figures 8 and 9). While failing to perform well under the ARI, this is a sensible result. Rather than indicating (as we did with the shared label) that no item is particularly distinct from the others and thus all share a common label, this clustering of singletons states that no item is more similar to any other and thus no two items should cluster together. We consider this evidence that an ensemble of Bayesian mixture models is not as susceptible to predicting labels than an ensemble based upon K-means clustering as in Şenbabaoğlu et al. (2014a,b).

Increasing S is also required when the dimensionality of the dataset is large. In this case it is due to individual chains exploring only a single mode (as can

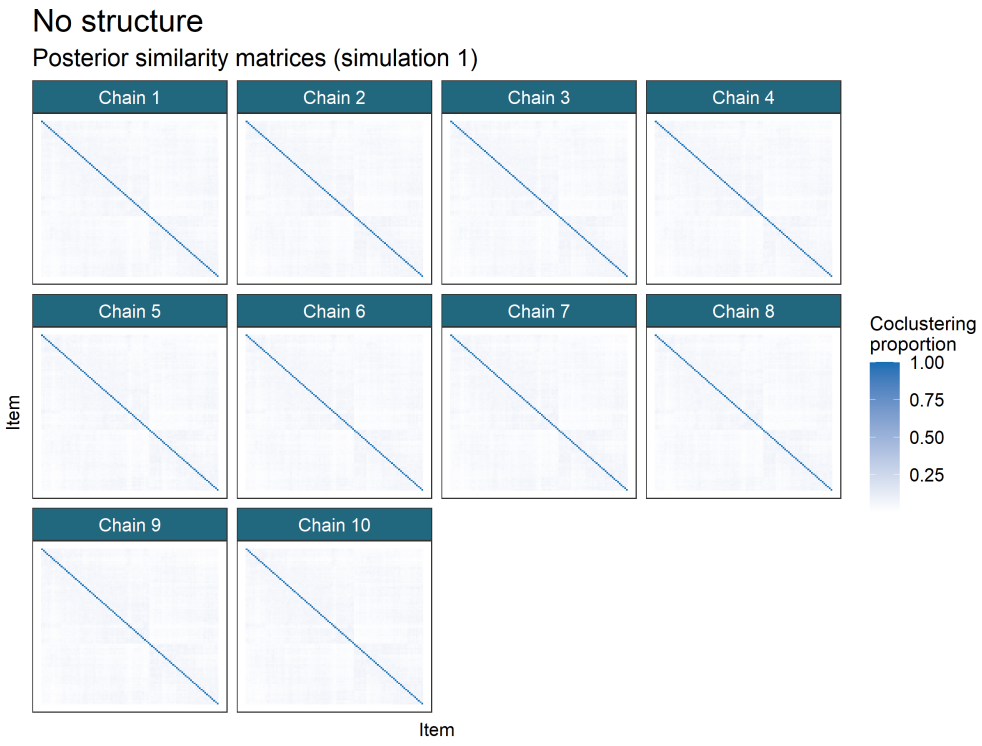

Figure 8: Posterior similarity matrices for simulation 1 of the *No structure* scenario. Each item is allocated to a singleton.

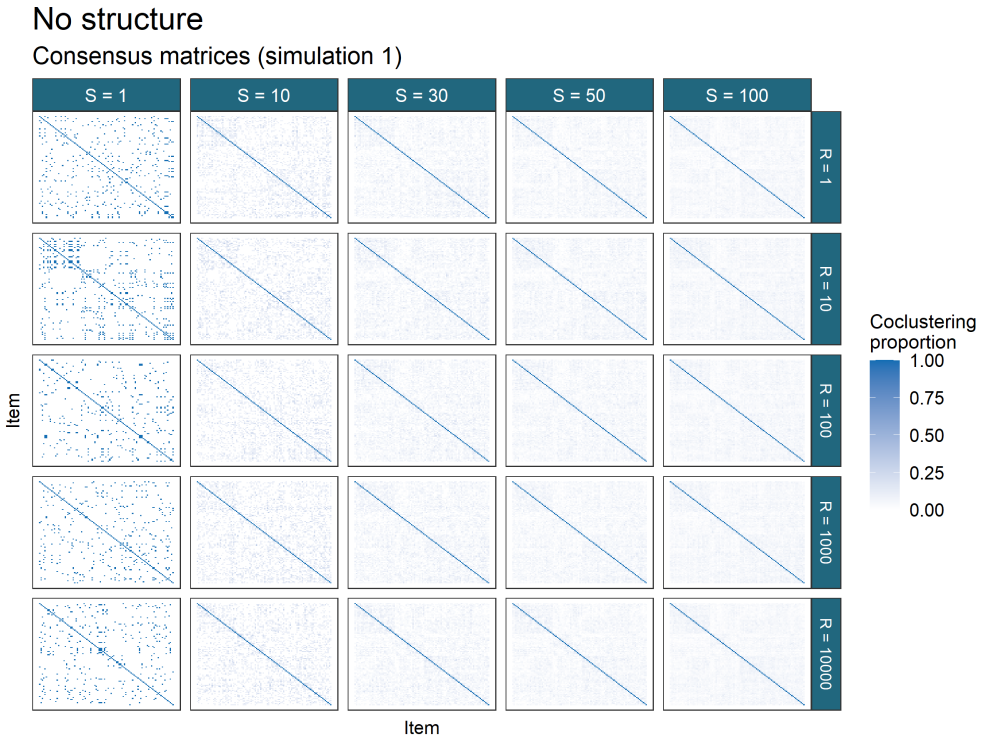

Figure 9: Consensus matrices for simulation 1 of the *No structure* scenario. Each item is allocated to a singleton in many of the Consensus matrices.

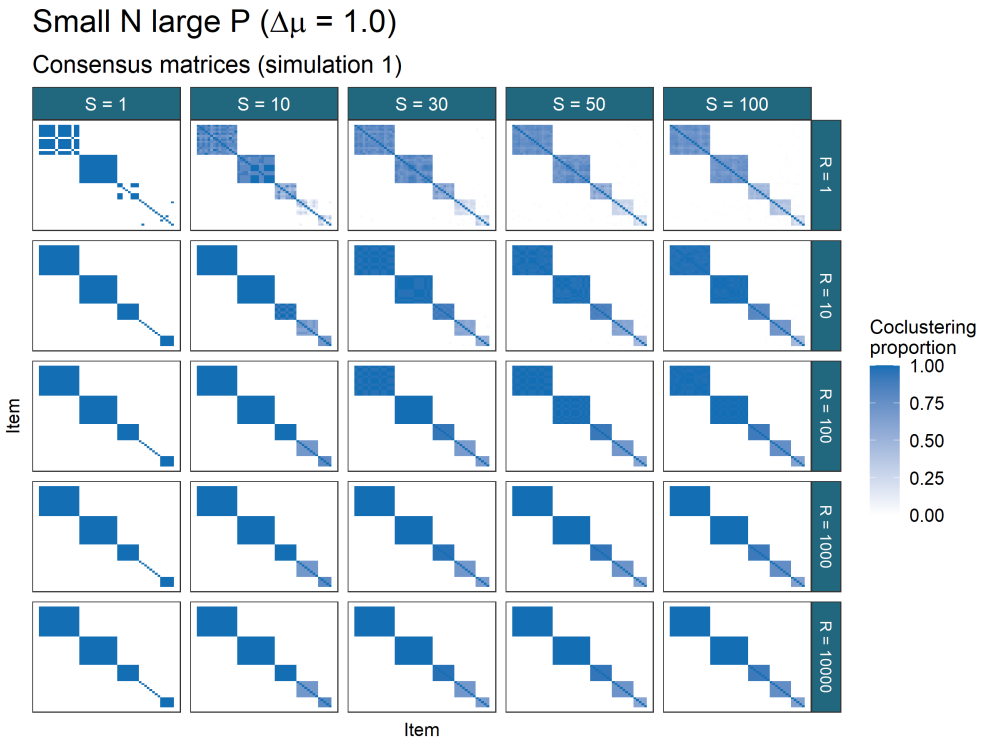

Figure 10: Consensus matrices for simulation 1 of the first *Large N, small P* scenario. One can see that by iteration ten the sample being drawn is from the mode (for $S = 1$, $R = 10$), and that an ensemble of chains does find structure that recalls the generating labels (see figure 11, the ARI for $CC(10, s)$ is 1.0 for $s > 1$, meaning that the true labels perfectly align with those predicted by the Consensus matrix).

be seen in figure 5 where each chain appears to sample only a single partition). In this example where each sample is a partition that appears to be a mode in the posterior distribution of the allocation vector from very early in the chain (based upon the stable performance for $R \geq 10$), increasing S allows each chain to "vote" on which mode is the global mode, as we believe that the mode that attracts the most chains is the global mode (although in real datasets the number of chains required might be greater than in our simulations). An example of this behaviour may be seen in figure 10.

In figure 11, limiting behaviour for increases of S and R can be seen for the ensemble. This inspires our belief that the ensemble depth and width should be grown until this limiting behaviours emerge. In practice where no ground truth is available, we recommend visually inspecting the Consensus matrices in a grid like figures 7, 9 and 10 and checking if there is any noticeable difference

between the Consensus matrices for some sets of chain depth, $R' = \{r_1, \ldots, r_a\}$, and chain length $S' = \{s_1, \ldots, s_b\}$ (where $r_i < r_j \iff i < j$ and $s_i < s_j \iff i < j$ for all i, j). If there is no difference between the Consensus matrix for (r_i, s_j) and that for (r_a, s_b) for any $i = 1, \ldots, a-1$, $j = 1, \ldots, b-1$ than the limiting behaviour is assumed to have emerged. For example, in figure 10 this limiting behaviour appears to have emerged by $R = 10$ and $S = 10$, but this can only be seen by comparing the ensembles of $R = 10, S = 30$ and $R = 100, S = 10$ with $R = 100, S = 30$. Note that we suggest the requirement that $\frac{r_a}{r_{a-1}} \leq 0.5$, $\frac{s_b}{s_{b-1}} \leq 0.5$ and $r_a, s_b \geq 10$.

Beyond the empricial behaviour of the ensembles in this simulation study, this heuristic is also 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.

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 Granovskaia 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 Granovskaia 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 mixture of Gaussian processes in the Timecourse dataset and Multinomial distributions in the ChIP-chip and PPI datasets. The data is shown in figure 13.

Following the original MDI paper we set $K_{max} = 275$.

4.1 Bayesian analysis

We used the implementation of MDI from Mason et al. (2016). 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

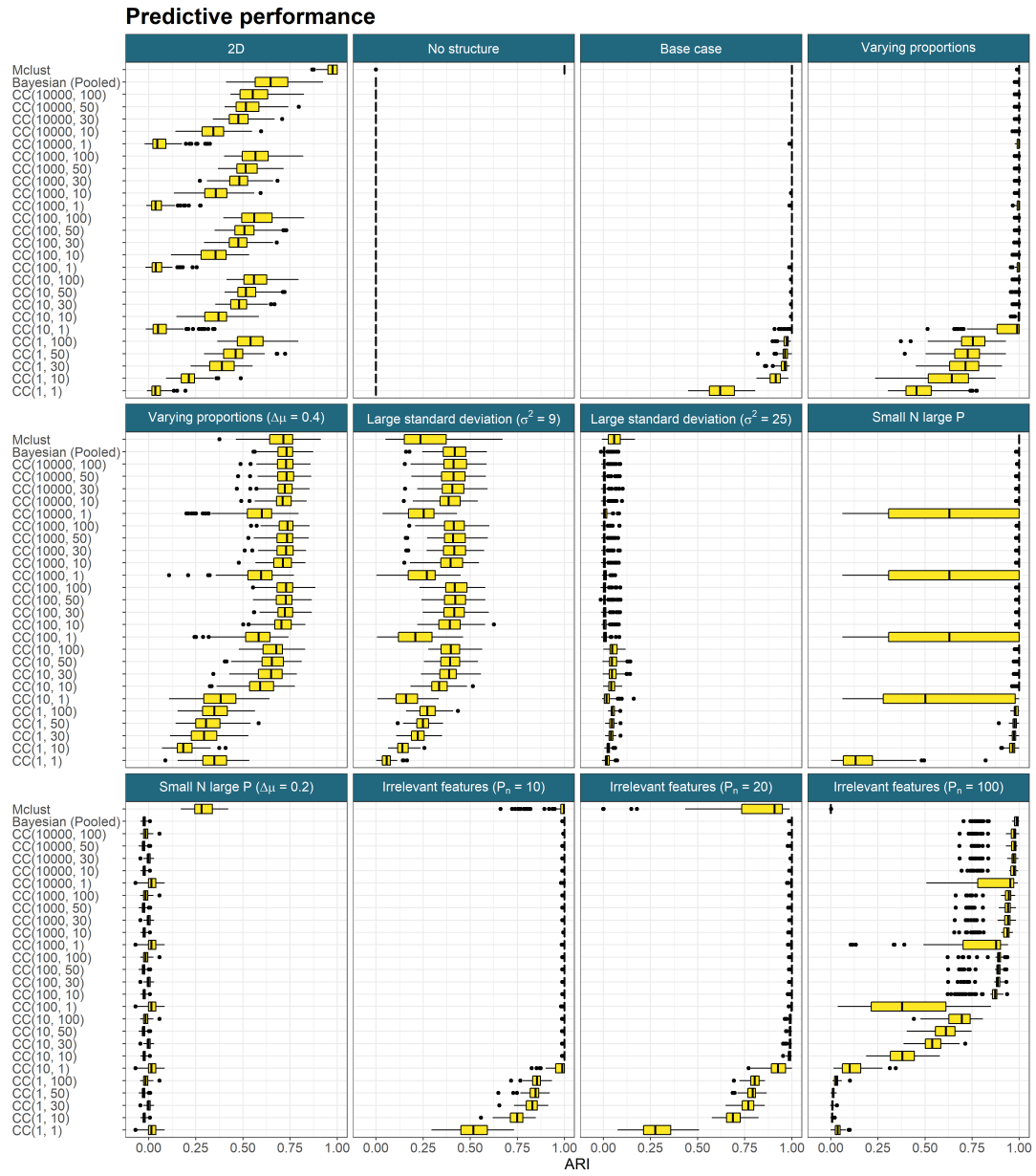

Figure 11: Predictive performance across all simulations. $CC(R, S)$ denotes Consensus clustering using the R^{th} sample from S different chains. In the cases where the generating structure is not exactly found, increasing R and S sees some improvement in the ARI between the truth and the predicted clusterings before some limiting behaviour emerges and and further increase appears to have no change in the performance.

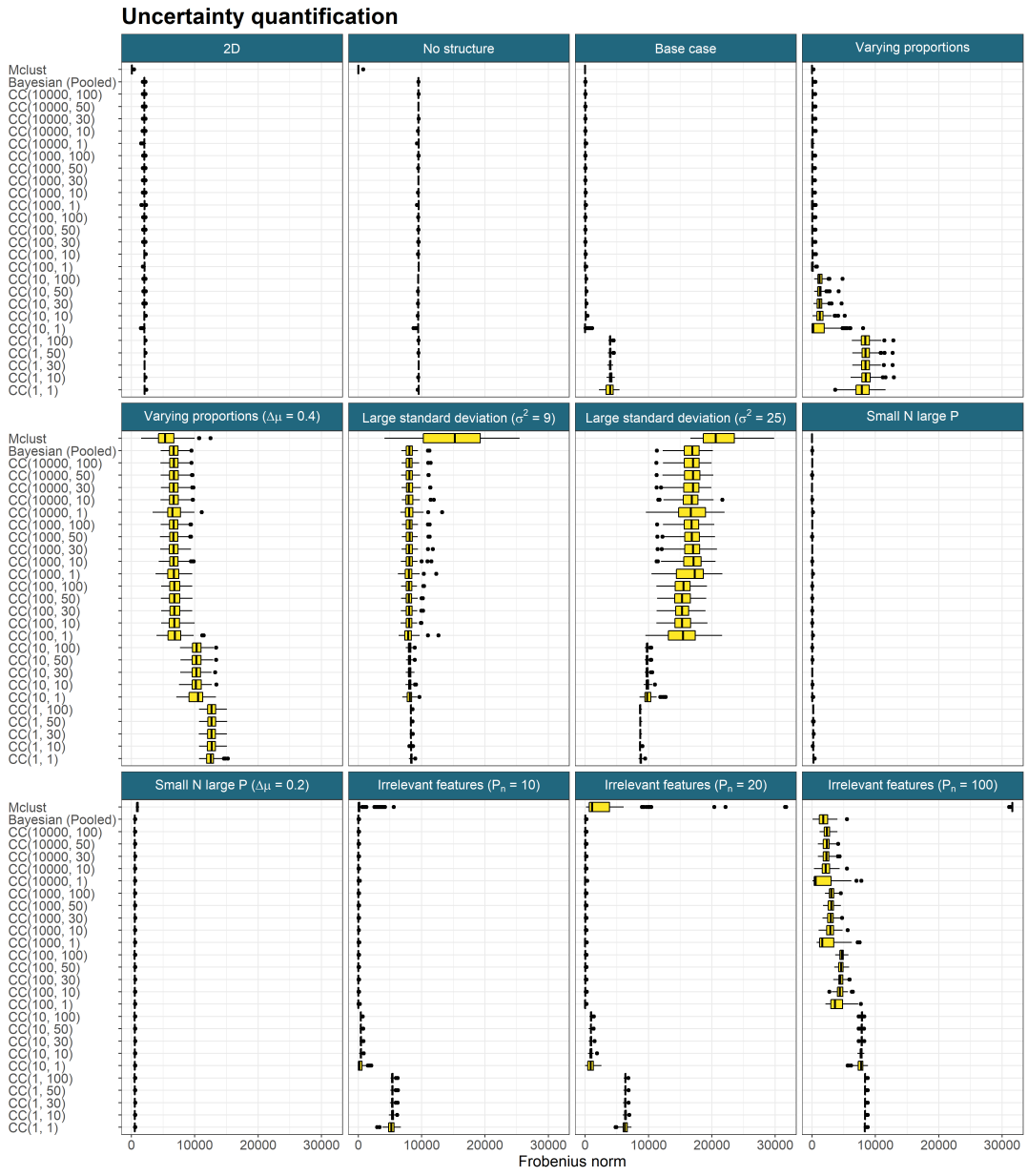

Figure 12: Frobenius norm across simulations. $CC(R, S)$ denotes Consensus clustering using the R^{th} sample from S different chains. Lower values are better. In the *Large standard deivation* ($\sigma^2 = 25$) scenario, the very low valued entries from the ensembles of very short chains are rewarded. These ensembles are not closer to the true structure than the longer ensembles, but they are rewarded for the lack of certainty.

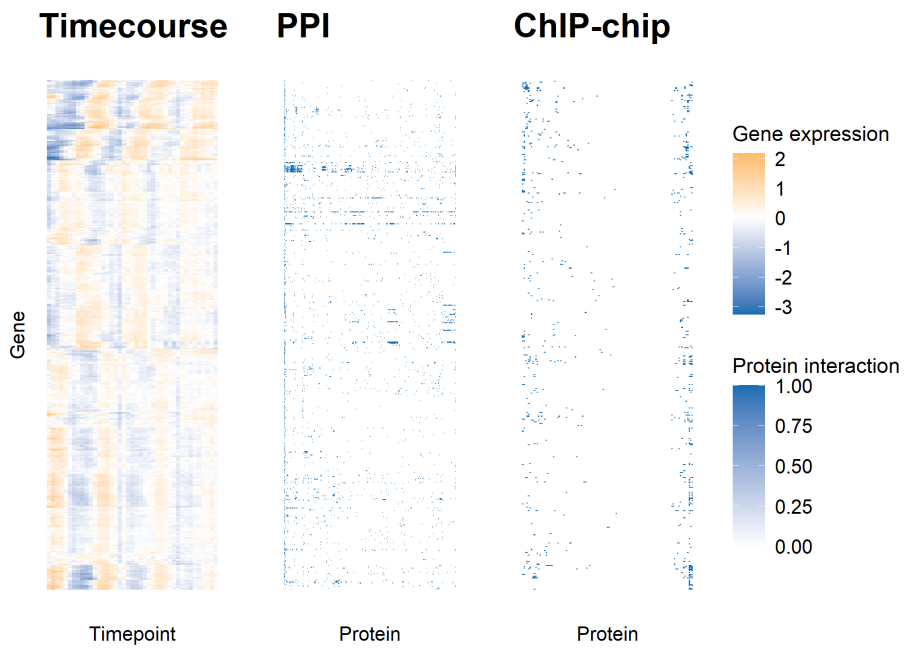

Figure 13: Heatmap of the yeast datasets. Each plot has a common row order corresponding to the gene products being clustered. This order was decided by a hierarchical clustering of the rows of the Timecourse expression matrix. The Timecourse data is associated with the "Gene expression" legend and the ChIP-chip and PPI data with "Protein interaction" legend.

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
- 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, 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 14. 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 15) and the different distributions sampled for the ϕ_{lm} parameters shown in figure 16.

We visualise the the PSMs for each dataset in figure 17.

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. Following the logic inspired by the behaviour seen in section 3.4, 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). An example of this logic can be seen in figures 19 and 20 (and to a lesser degree in figure 18). 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. We also find that the clusters of time series in the Timecourse dataset shown in figure 22 to be consistent.

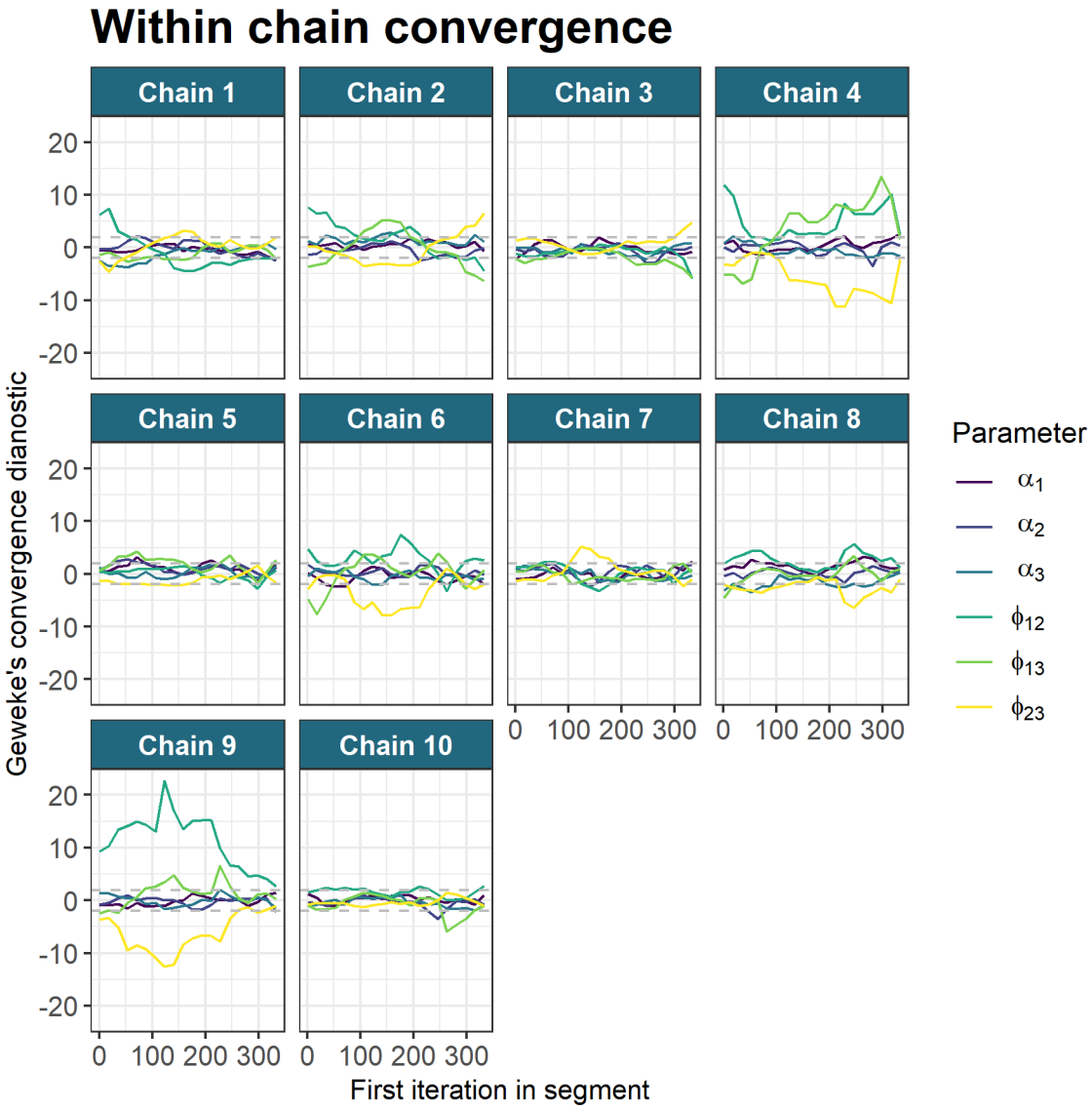

Figure 14: Chain 9 can be seen to have the most extreme behaviour in the distribution of the Geweke diagnostic for the parameters. We remove this chain from the analysis. Of the remaining chains we believe that 1, 2, 4 and 6 express the distributions furthest removed from the desired behaviour and are dropped from the analysis.

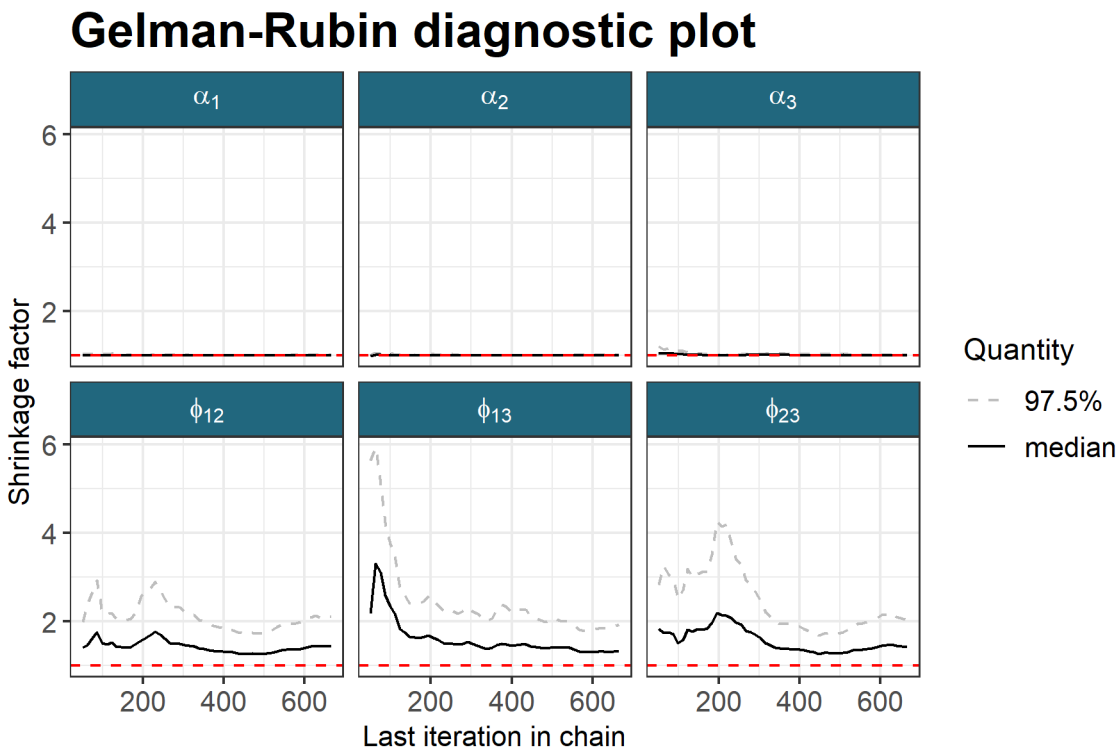

Figure 15: The chains still appear to be unconverged with $\hat{R}$ remaining above 1.25 for the ϕ_{12}, ϕ_{13} and ϕ_{23} parameters. Stable $\hat{R}$ is also too high with values of 1.049, 1.052 and 1.057 for ϕ_{12}, ϕ_{13} and ϕ_{23} respectively.

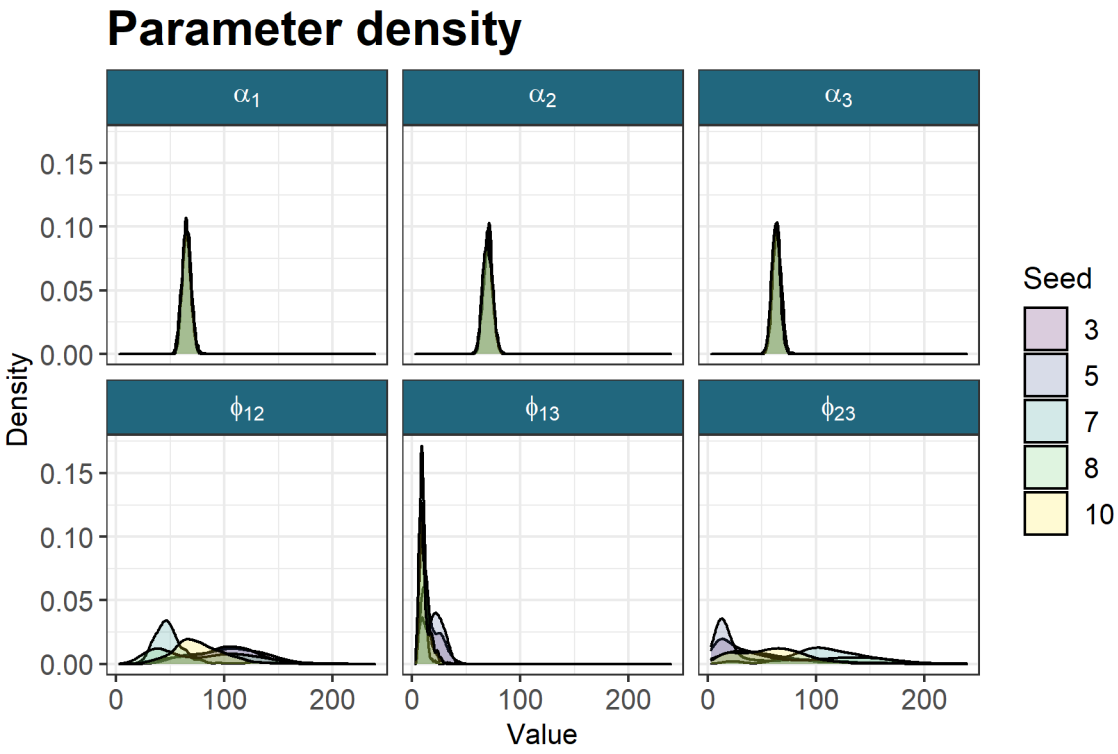

Figure 16: The densities of the continuous variables across the 5 chains kept for analysis. The mean sampled values are $\alpha_1 = 64.84$, $\alpha_2 = 69.85$, $\alpha_2 = 63.22$, $\phi_{12} = 81.76$, $\phi_{13} = 13.87$, and $\phi_{23} = 65.03$. It can be seen that different modes are being sampled for the ϕ parameters in each chain.

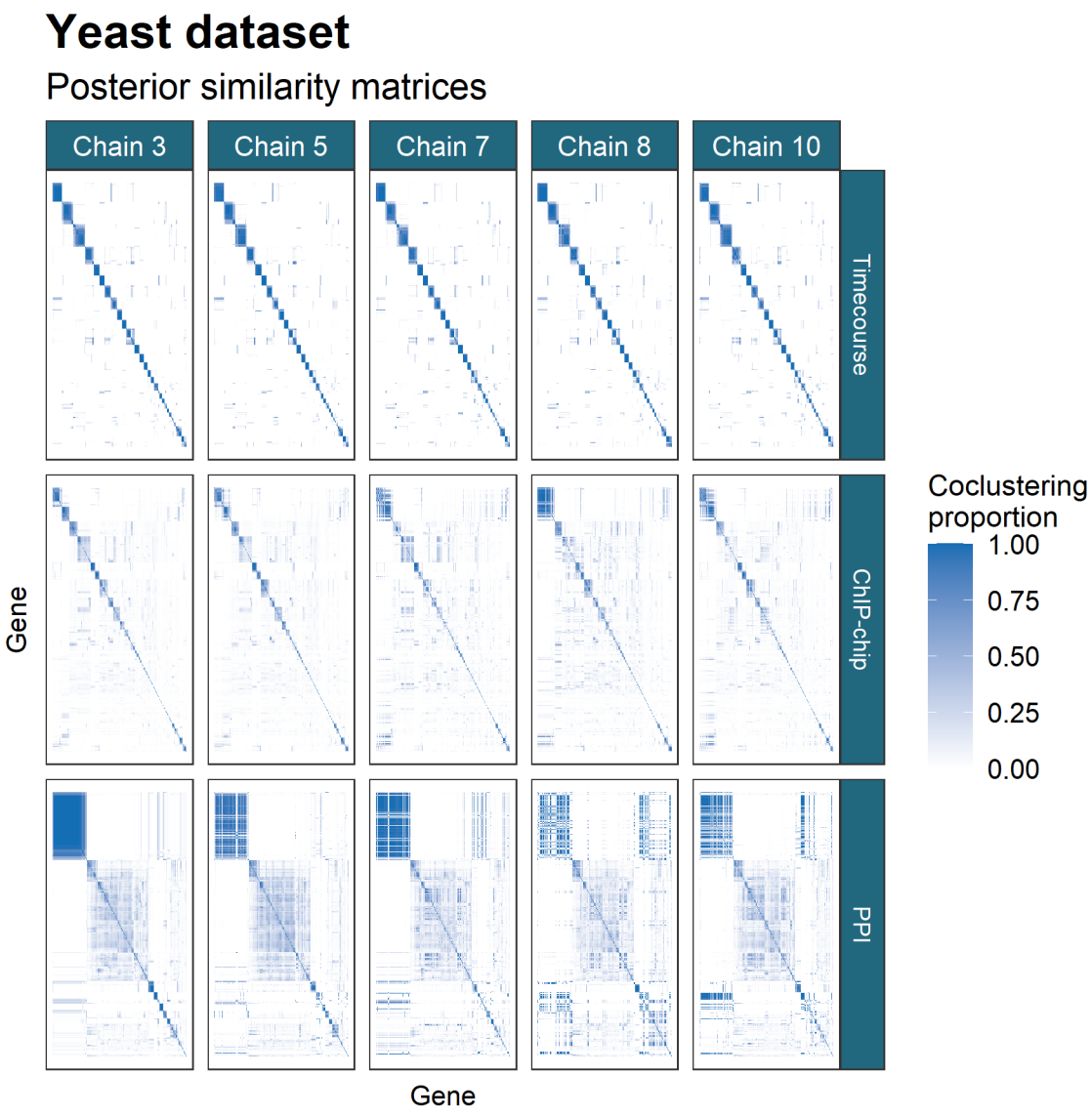

Figure 17: PSMs for each chain within each dataset. The PSMs are ordered by hierarchical clustering of the rows of the PSM for chain 3 in each dataset. There is no marked difference between the matrices for the Timecourse data with disagreement becoming more prominent in the ChIP-chip data and more so again in the PPI dataset.

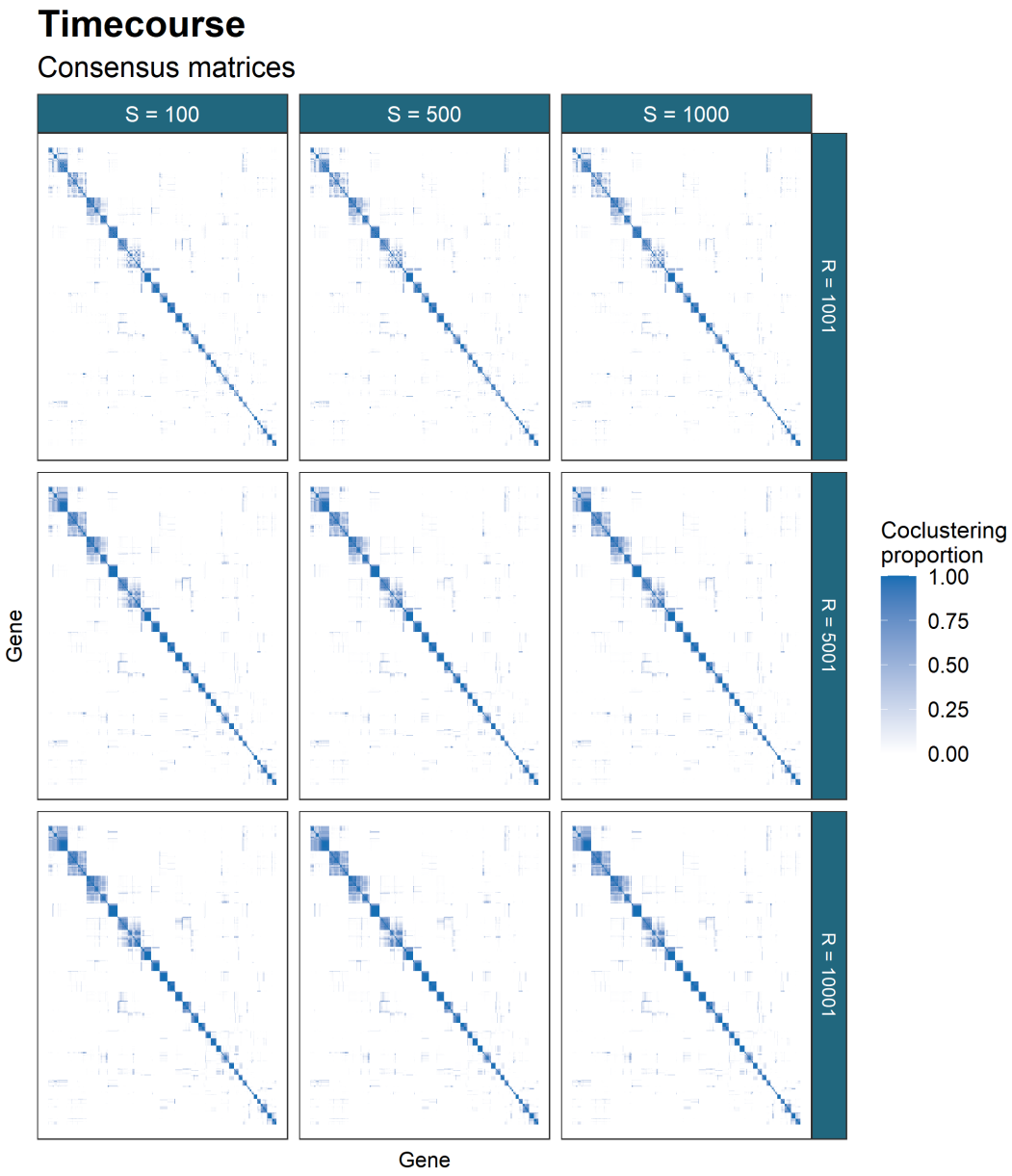

Figure 18: Consensus matrices for different ensembles of MDI for the Timecourse data. This dataset has stable clustering across the different choices of number of chains, S, and chain depth, R, with some components merging as the chain depth increases.

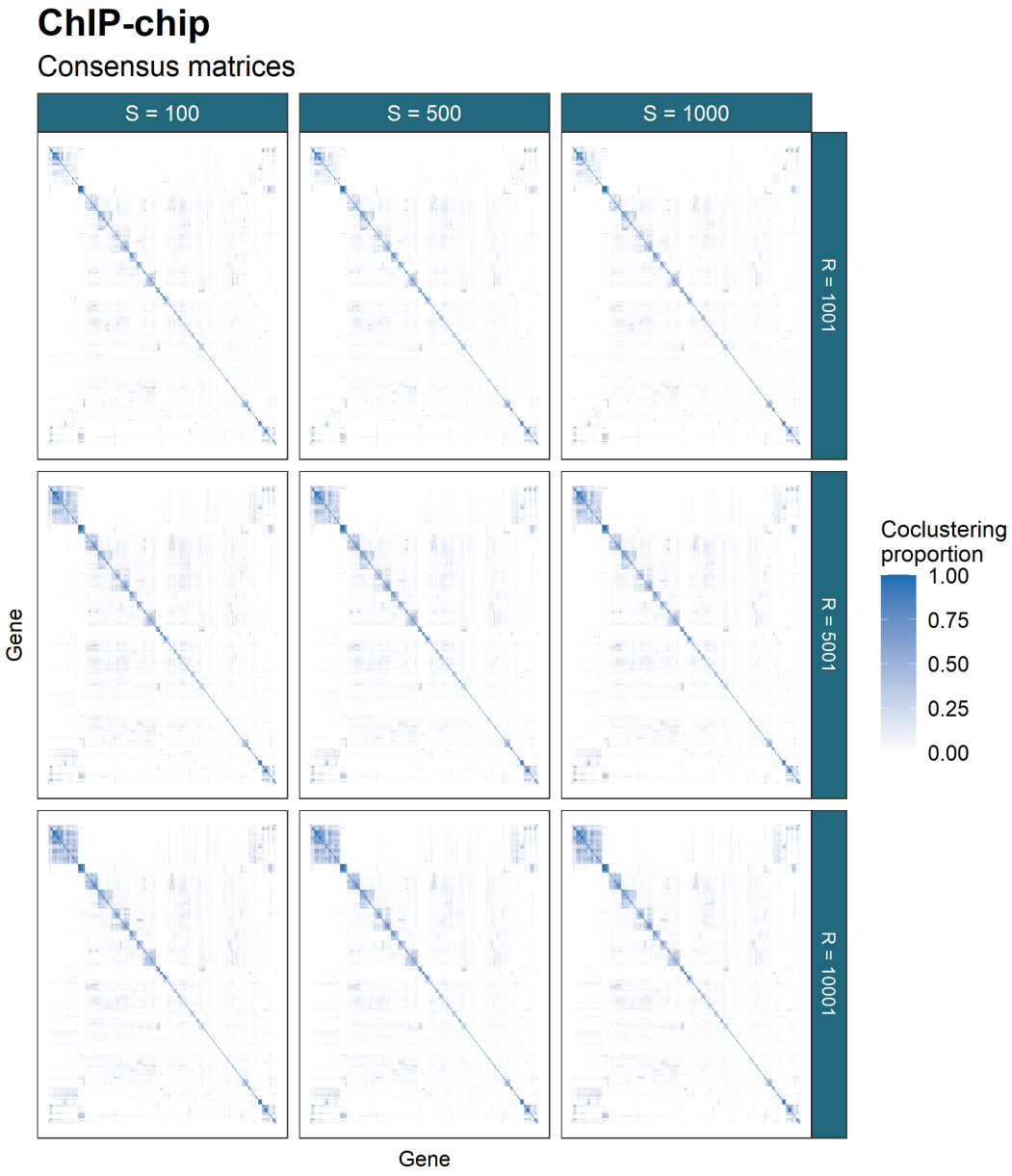

Figure 19: The ChIP-chip dataset is more sparse than the Timecourse data. In keeping with the results from the simulations for mixture models, deeper chains are required for better performance. It is only between $R = 5,001$ and $R = 10,001$ that no change in the clustering can be observed and the result is believed to be stable. In this dataset the number of chains used, S, appears relatively unimportant, with similar results for $S = 100, 500, 1000$.

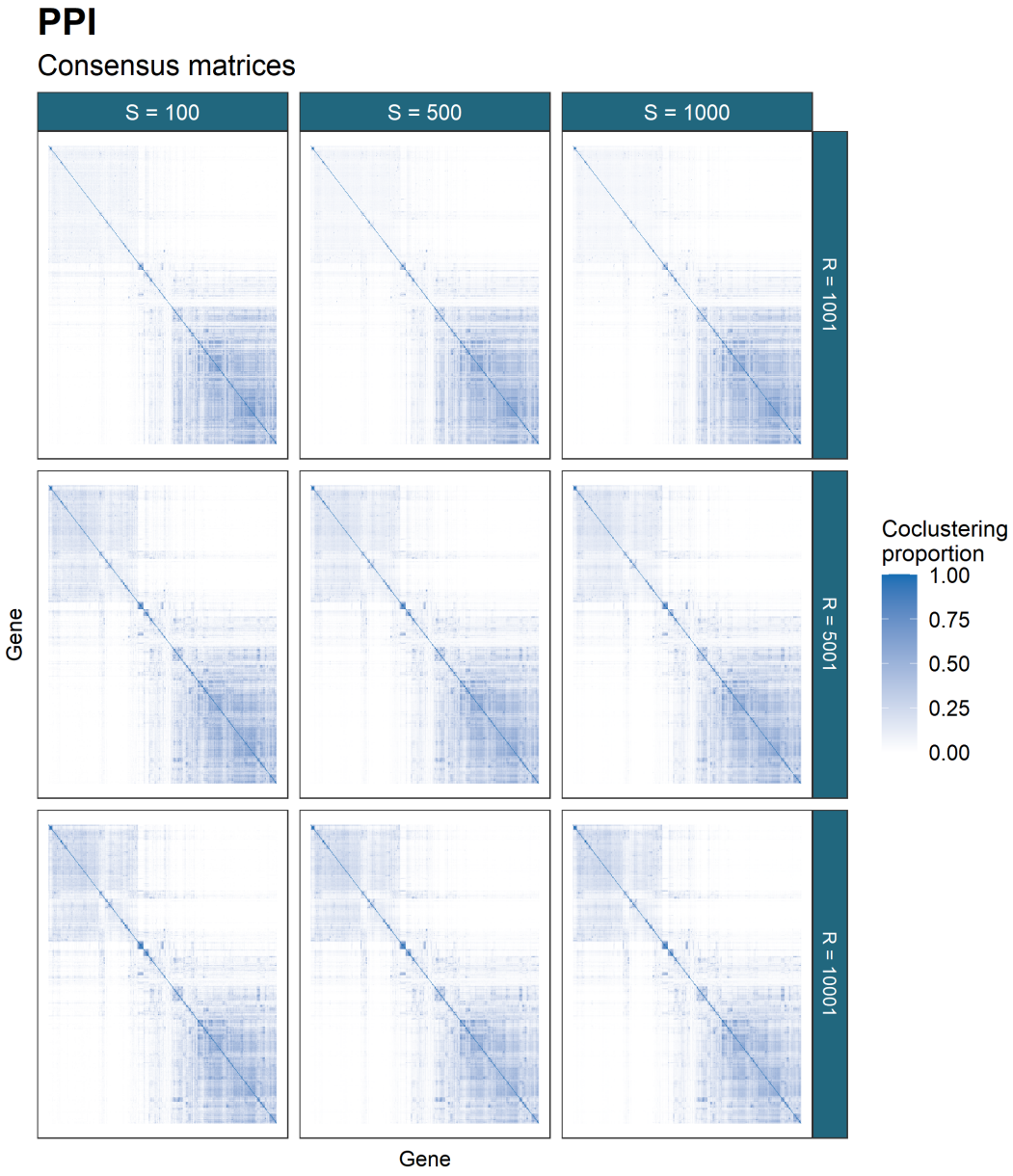

Figure 20: The PPI dataset has awkward characteristics for modelling. A wide, sparse dataset it is chain depth that we found to be the most important parameter for the ensemble. Similar to the results in figure 19, the matrices only stabilise from $R = 5001$ to $R = 10001$.

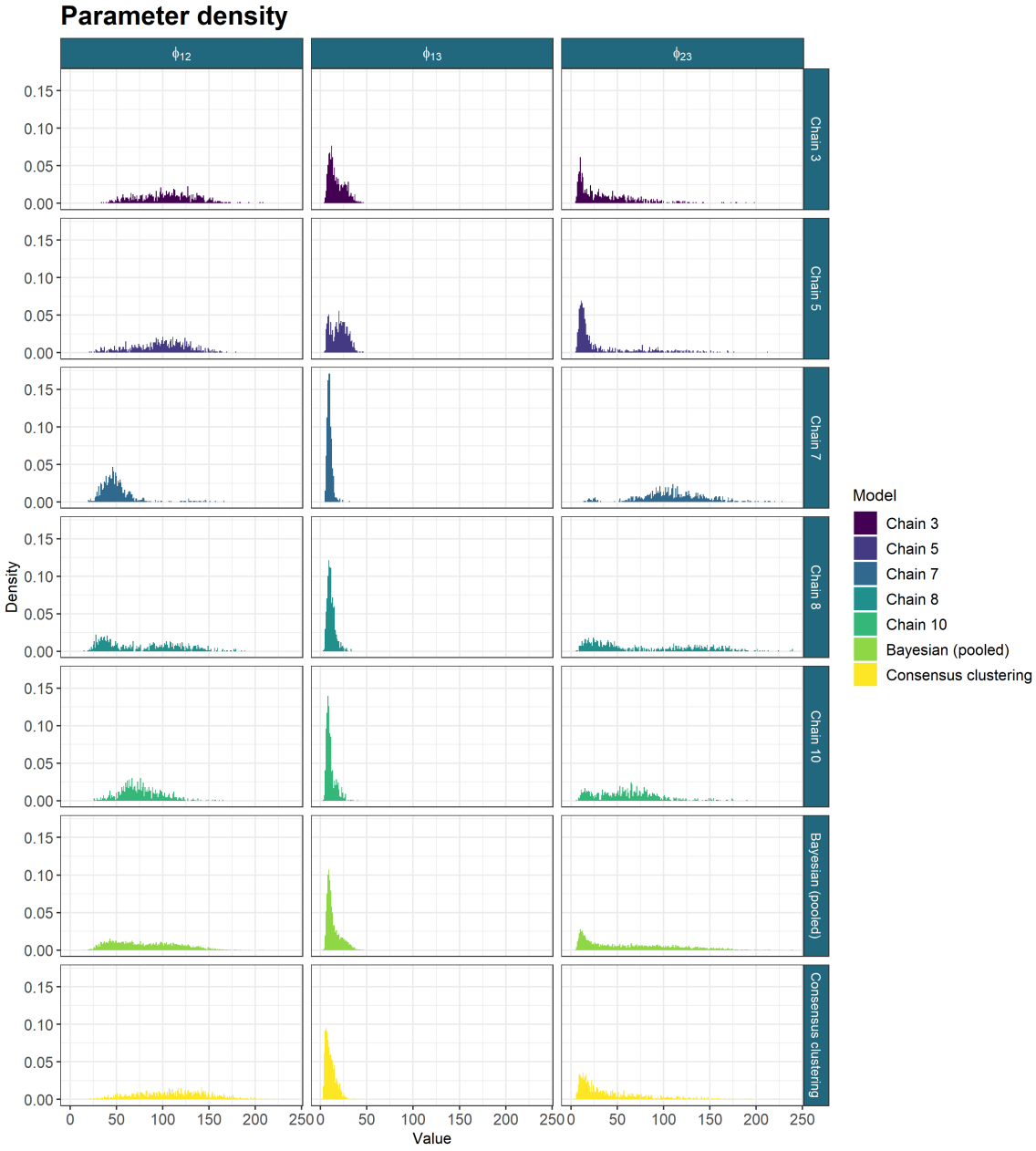

Figure 21: The sampled values for the ϕ parameters from the long chains, their pooled samples and the consensus using 1000 chains of depth 10,001. The long chains display a variety of behaviours. Across chains there is no clear consensus on the nature of the posterior distribution. The samples from any 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.

Evidence for this can also be seen in the coherent clusters of time series from the Timecourse dataset in figure 22.

4.3 GO term over-representation

To validate the predicted clusters we tested if they contained a higher concentration of specific Gene Ontology (GO) terms than would be expected by chance. We estimated clusterings from the PSMs of the chains kept from section 4.1 visualised in figure 17 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 chance. We corrected the p-values using the Benjamini-Hochberg correction (Benjamini and Hochberg, 1995) and defined significance by a threshold of 0.01. We plot the over-represented GO terms for the different clusterings within each dataset using the three different ontologies of "Molecular function" (**MF**), "Biological process" (**BP**) and "Cellular component" (**CC**) (figures 23, 24 and 25 respectively).

We find that the Consensus clustering finds terms common to all of the long chains. It also finds some terms with low p-values common to a majority of chains (such as DNA helicase activity in the MF ontology for the Timecourse dataset) and a small number of GO terms unique to itself. These terms that no long chain find are normally related to other terms already enriched either in the Consensus clustering or a number of the long chains. For example, we believe that uncovering transmembrane transporter activity and transporter activity terms in the Timecourse dataset due to an unstable or inconsistent clustering as these are related to terms found across 3 of the chains and by Consensus clustering (specifically transferase activity and phosphotransferase). We believe that Consensus clustering of MDI results in a meaningful clustering.

Furthermore, we also note that the Bayesian chains have very significant disagreements between eachother; there is no consensus on the results with many terms enriched in one or two chains (see the behaviour in any ontology for the ChIP-chip and PPI datasets).

We argue that the final partition from Consensus clustering is more consistent than any of the individual long chains, agreeing where the chains agree and providing sensible differences to any given chain. As the chains are not converged and there is no clear consensus, a full analysis would be difficult to defend using any one of these long chains. We believe that Consensus clustering

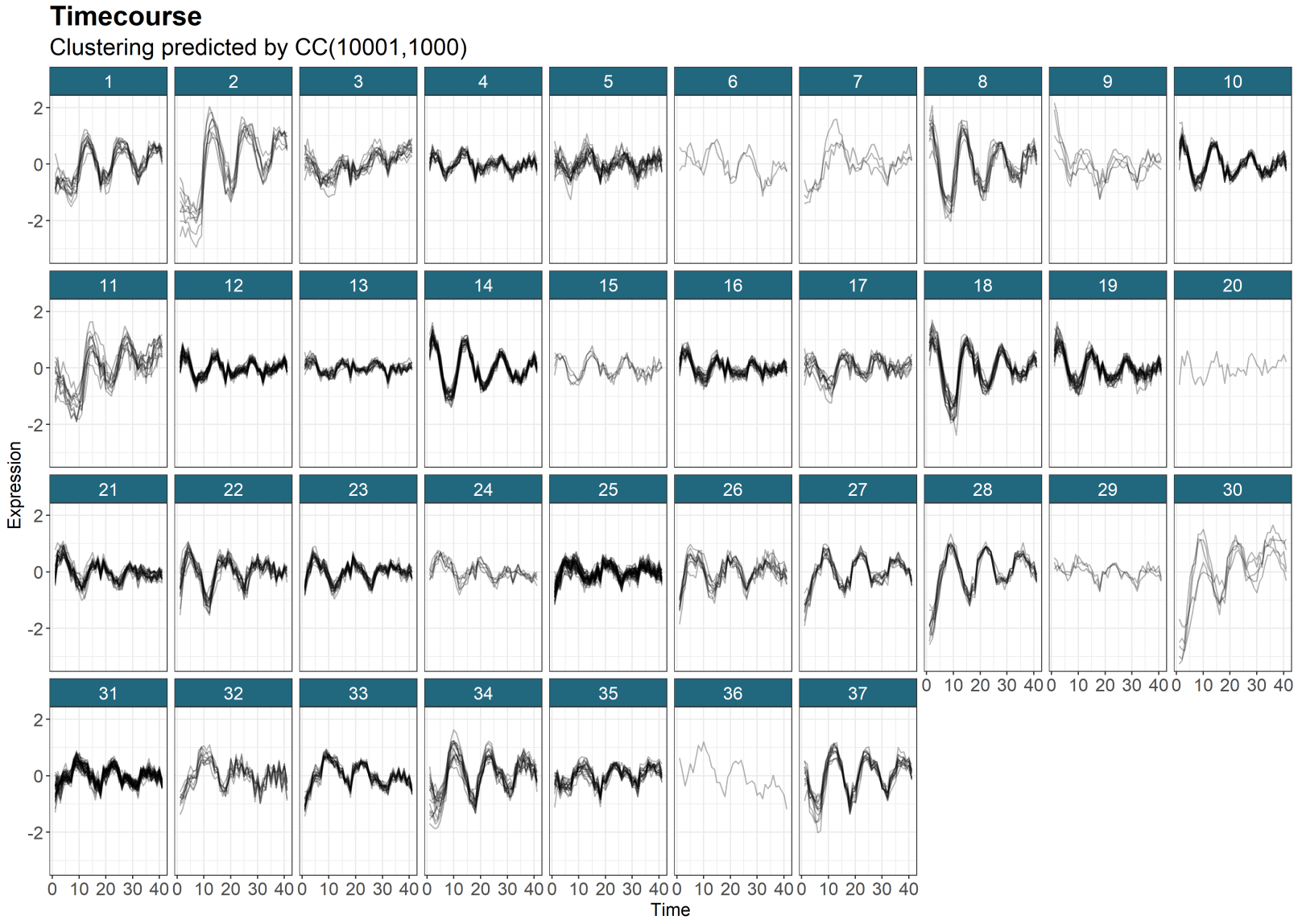

Figure 22: Gene expression across time for the predicted clusters in the Timecourse dataset for Consensus clustering of MDI with $R = 10001$ and $S = 1000$.

does offer a solution to problems preventing the use of complex models such as MDI in their currently implemented forms.

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 Ṣenbabaoğlu, George Michailidis, and Jun Z Li. A reassessment of consensus clustering for class discovery. *bioRxiv*, page 002642, 2014a.

Yasin Ṣenbabaoğlu, George Michailidis, and Jun Z Li. Critical limitations of consensus clustering in class discovery. *Scientific reports*, 4(1):1–13, 2014b.

Arno Fritsch. *mcclust: Process an MCMC Sample of Clusterings*, 2012. URL `https://CRAN.R-project.org/package=mcclust`. R package version 1.0.

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

Evelina Gabasova, John Reid, and Lorenz Wernisch. Clusternomics: Integrative context-dependent clustering for heterogeneous datasets. *PLoS computational biology*, 13(10):e1005781, 2017.

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.

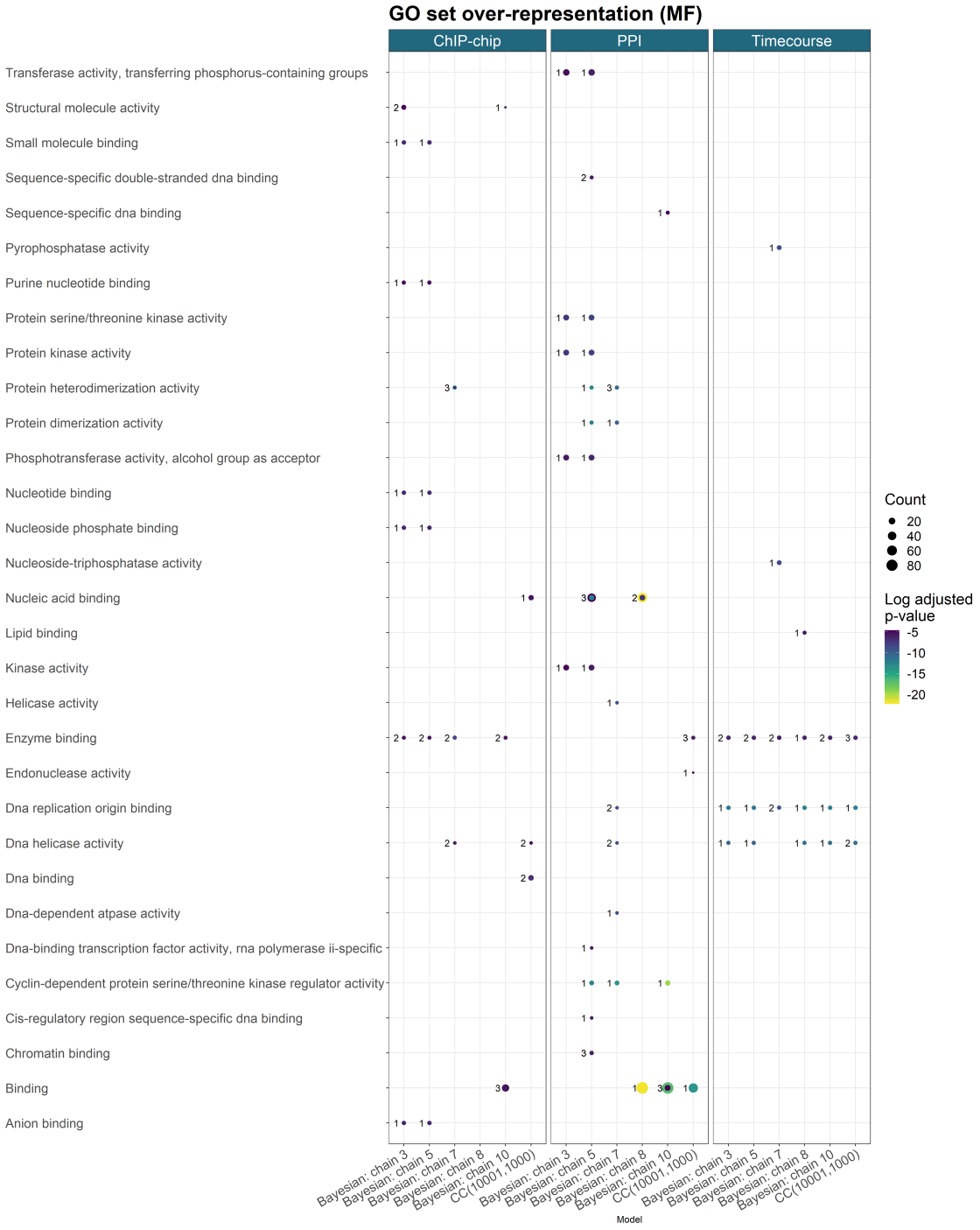

Figure 23: GO term over-representation for the Molecular function ontology for each dataset from the final clustering of each method.

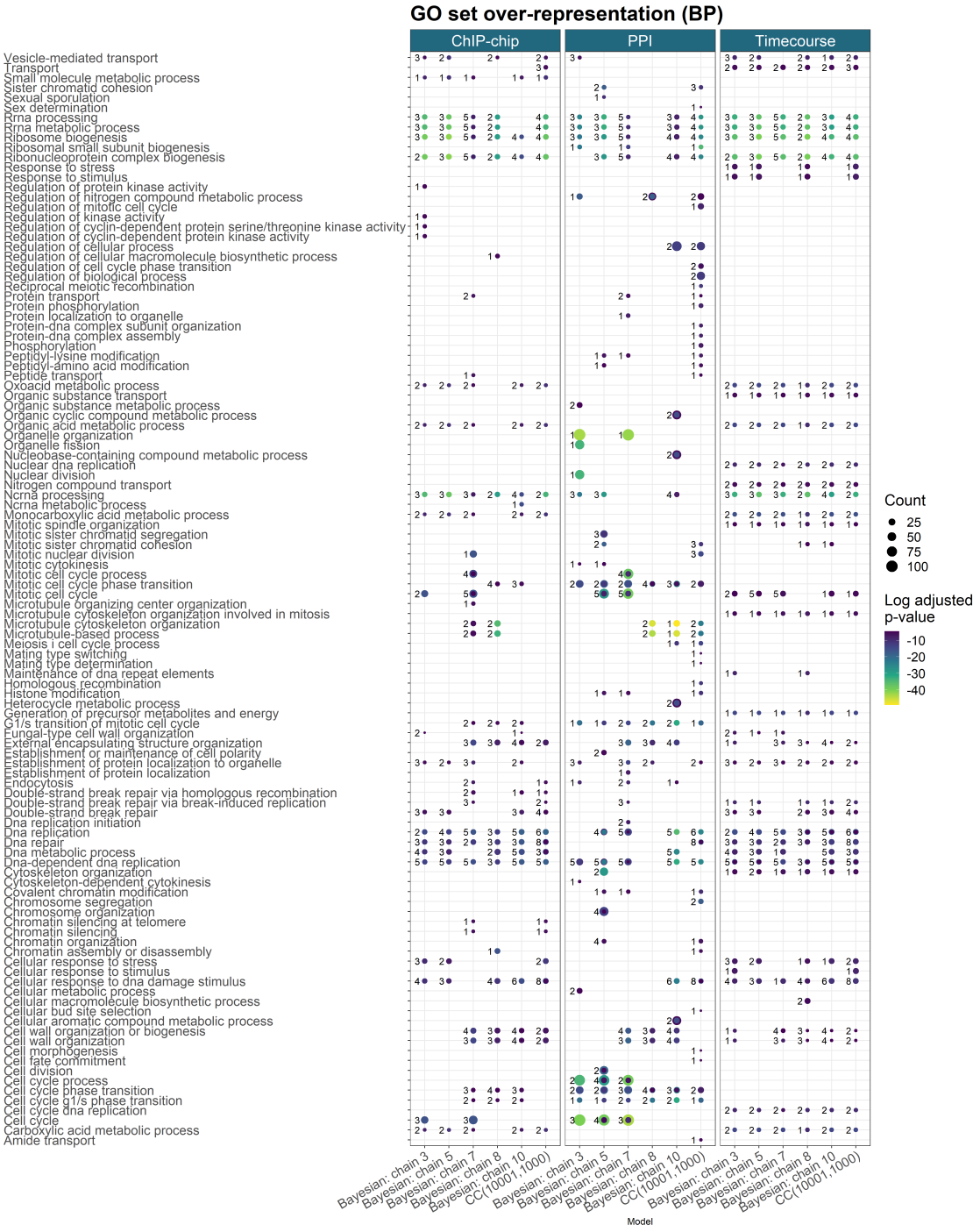

Figure 24: GO term over-representation for the Biological process ontology for each dataset from the final clustering of each method.

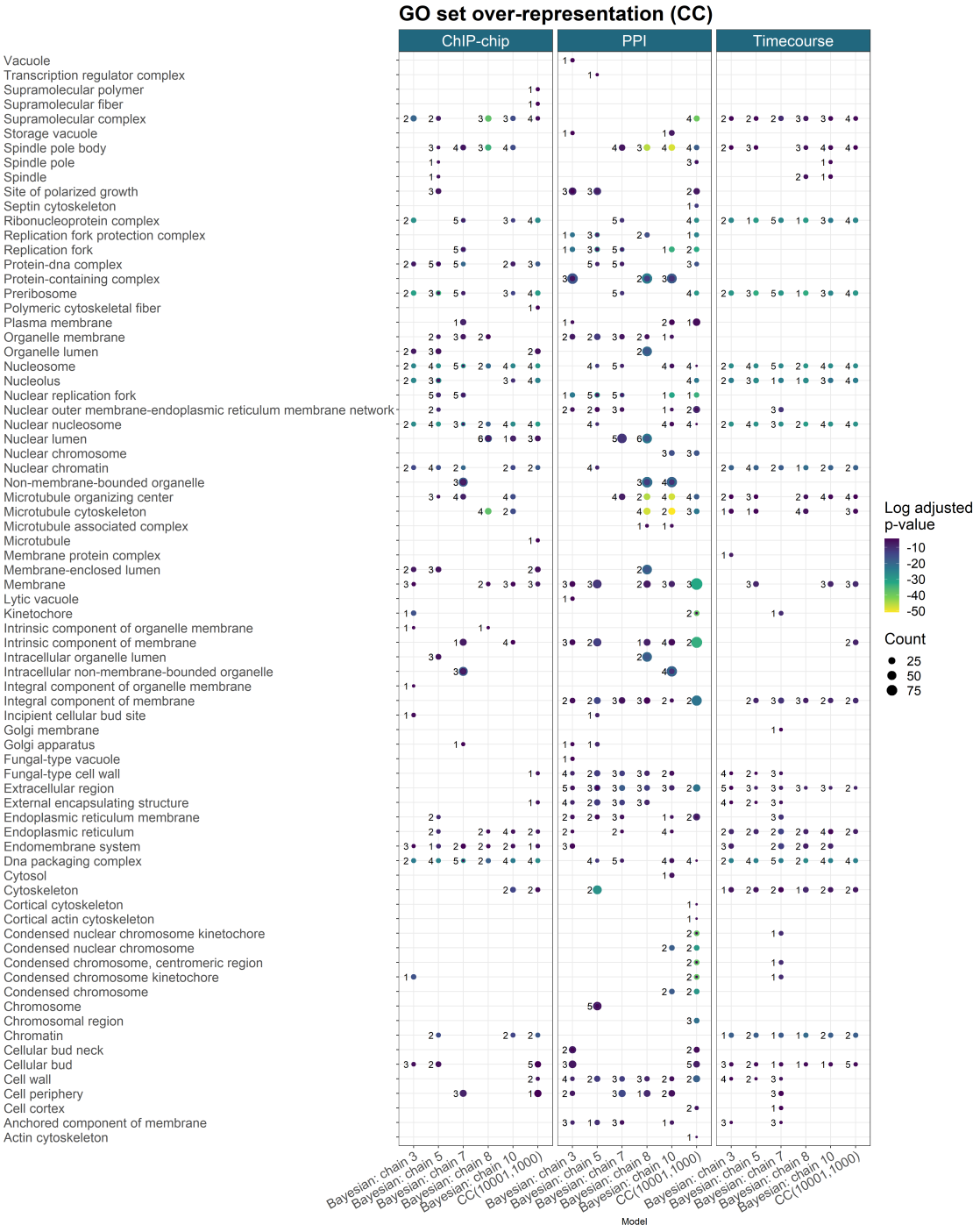

Figure 25: GO term over-representation for the Cellular component ontology for each dataset from the final clustering of each method.

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 gp-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. clusterprofiler: 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.