Defining tissue specific gene sets using consensus clustering

GEN80436

Stephen Coleman
940309160050

supervised by
Bas Zwaan
Laboratory of Genetics, Wageningen University
and
Chris Wallace
Department of Medicine, Cambridge University

A thesis presented for the degree of
Master's in Bioinformatics

Laboratory of Genetics
Wageningen University

Abstract

Bayesian inference of large and / or high dimensional sets can be limited by the computational cost of Markov chain Monte Carlo (MCMC) methods and the difficulty of sampling the full posterior density for many such methods. I propose consensus clustering as an alternative inferential paradigm. Consensus clustering is a computationally tractable method of implementing approximate Bayesian inference. I show via simulation that this method approximates Bayesian inference well when MCMC methods can successfully describe the target distribution in finite time, that it can describe multiple modes of a complex space, and that the method is quick. I apply this method, carrying out consensus clustering inference of an integrative clustering model with real data. The data used consists of several cell type or tissue specific gene expression datasets. I attempt to recreate a KEGG pathway as a validation of the clustering method. This analysis aims to explore the concept of annotating gene sets with tissue and cell-type specific information. I find encouraging results for this concept that suggest further work could be rewarded in this area.

Acknowledgements

This project is based upon the data collected by The International IBD Genetics Consortium et al. [51] and the implementation of Multiple Dataset Integration coded by Mason et al. [33]. My consensus clustering model is encoded in bash. All analysis and data preparation is done using the R language [50]. The data.table [10], magrittr [2] and optparse [9] packages were integral to the analysis pipeline. In terms of visualisation, all plots other than the convergence diagnostic plots were generated using the pheatmap [28] and ggplot2 [55] packages. The exceptions were generated using the rjags [43] package. The tibble [38] package eased saving and holding of data. This report was generated using LaTeX.

Thanks are due to Paul Kirk of the Cambridge Biostatistics Unit (BSU) for his advice throughout the project and specifically regarding how to sense-check clustering results. Colin Starr of the BSU (maintainer of the MDI-GPU software) and Stuart Rankin of the Cambridge Research Computing Services were always generous with their time and help. More generally, the Wallace Group was incredibly welcoming; my time with them has been both pleasant and enlightening.

Contents

1 Introduction

With the onset of microarrays and RNAseq, producing gene expression data in large quantities for a wide number of genes is increasingly enabled. Unfortunately the large amount of data now available to the genomics community by these methods is difficult to interpret and analyse. Gene Set Enrichment Analysis (GSEA) attempts to overcome some of these issues by using prior knowledge to define groups of genes linked through their biological function [22]. The set is defined using knowledge external to the current analysis; a common method is using the manually annotated discrete pathways available on the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [13].

Analysis of gene sets is advantageous both from the perspective of biological interpretation and statistical power. In terms of the underlying biology, genes do not function in isolation; they participate in pathways, interacting with other genes to carry out specific biological processes. Thus analysis of gene sets is analysis of an object closer to phenotype than the individual gene. Furthermore, it is known from Genome Wide Association Studies that many diseases are polygenic in nature [37]. Subramanian et al. [48] highlight the importance of gene sets, claiming that within a single metabolic pathway an increase of 20% in all the associated gene products may have more impact upon phenotype than a 20-fold increase in a single gene. Thus gene sets have more relevance than individual genes to biological impact. The interpretation of gene sets in the context of pathways is a step closer to phenotype than the information encoded in individual genes. As a deeper biological understanding of the phenotype or disease state is the aim, this encourages the use of gene sets.

In terms of statistical power, in analysing gene sets as a group the degree of perturbation required in the expression of the full gene set due to the disease state / alternative phenotype to be considered significant is much less than that required in analysing each of its constituent members individually [12][56][37].

There is also no obvious detriment in analysing gene sets - no loss of information found. As the gene sets are expected to have correlated expression [54], one expects that if the expression of a gene within the set does change then, if this is not due to noise or stochasticity, the expression of other members of the set should also vary accordingly.

Thus clustering genes into groups known as “gene sets” is natural and useful from both a biological and statistical perspective - it can increase the interpretability and the power of an analysis [39][53].

The problem of how to define gene sets is non-trivial, with many variations present in the literature. There exist many databases of gene sets [1][26][49]. The Molecular Signature Database [48] (MSigDB) is one of the most popular resources for GSEA and encompasses many different gene sets defined under

various criteria or generated from separate resources.

However, none of these definitions of a "set" incorporate tissue specific information. This seems an oversight. Cell-type specific gene pathways are pivotal in differentiating tissue function, implicated in hereditary organ failure, and mediate acquired chronic disease [25]. More and more evidence is being accrued to highlight the cell-type specific level of gene expression [19][40][32]. Thus I propose introducing tissue and cell-type specific annotation to gene sets.

Specifically, I identify gene sets based upon common patterns of expression within cell-type or tissue specific datasets. Correlated expression (or co-expression) between genes is often an indicator that they are:

- Controlled by the same transcription factor;
- Functionally related; or
- Members of the same pathway [54].

As this correlated expression is represented by a common variation across people (or experimental conditions) rather than in the magnitude of expression, I will standardise the expression data as described in section A.2. I describe a small example to highlight my reasoning in section A.2.1.

Previous attempts to achieve cell-type or tissue-specific annotation have used the Genotype Tissue Expression (GTEx) [20] database [31]. Here the profiles are for human donors, post-mortem. One might suspect that the data derived from these cells may not contain the same information as that collected from living, active cells. Furthermore, the GTEx data is across many different tissues (144 are used in [31]), but I focus on cell types relevant to autoimmune disease in general (i.e. white blood cells) and Inflammatory Bowel Disease in particular (intestinal samples).

—- I AM NOT SURE ABOUT THIS ————————————

I am of the opinion that in an exploration of a concept (as described here) one should focus on a smaller dataset to enable analysis and interpretation; if one uses a mountain of data and compares across enough cases one is quite likely to find some sort of evidence for an argument. Defining the argument in advance, restricting the options available for exploration and uncovering evidence in this context is something I find more convincing. Thus, a smaller, restricted dataset is considered.

—- END ——————————————————————————

To describe gene sets within the data, some clustering method is required. Applied on expression values or some transformed variation thereof, groups of genes are created based on some concept of similarity (or alternatively on some concept of dissimilarity or distance). Depending on the choice of transformation and clustering method further questions might arise such as defining the

number of clusters (required for instance with *K*-means clustering) or the type of distance to use (for instance within hierarchical clustering and the methods that integrate this method such as Weighted Gene Correlation Network Analysis). For clustering within a dataset I choose *mixture models* as the method as the number of clusters is inferred from the data and the concept of distance in these models is based upon the likelihood of the Gaussian distributions describing the sub-populations, an intuitive model for continuous data. Specifically I use Bayesian mixture models as these capture uncertainty of membership which is appropriate in this application. A gene's membership might be poorly defined [42]; thus the model uncertainty represents biological uncertainty.

Ideally a model could integrate information about common clustering structure across the tissue and cell-type specific datasets. Such methods are referred to as *integrative clustering methods*. I would like to reduce uncertainty without making assumptions that could impose false structure upon the data or in some other way reduce the signal unique to each tissue. From the field of integrative clustering methods, I choose to use *Multiple Dataset Integration* (MDI) [27] as this method has a minimum number of assumptions (it allows the models on each dataset to become independent if there is no common structure), is Bayesian (and thus has principled quantification of uncertainty) and is an extension of mixture models (a type of clustering model I like for reasons described above).

Bayesian inference relies upon computationally costly *Markov Chain Monte Carlo* (MCMC) methods to sample from the posterior distribution. When the target posterior distribution is multi-modal (which will often be the case when dealing with large and/or high-dimensional datasets), such sampling approaches may fail in practice, since individual chains may get stuck in local modes [52]. There exists many clever sampling methods that aim to overcome this problem such as split-merge steps [8] and Hamiltonian Monte Carlo [11] [23]. However, these methods are both difficult to implement and computationally expensive.

These problems are particularly prevalent in Bayesian clustering methods. The large number of discrete labels encourages a spiky likelihood surface that can trap MCMC chains. Thus convergence can take a space of time beyond realistic constraints. This means that a method capable of overcoming multi-modality and doing so in a useful time-frame (e.g. on the scale of 24 hours) while quantifying uncertainty is highly attractive. I propose using a fast consensus clustering approach for performing approximate Bayesian inference in model-based clustering. Consensus clustering traditionally uses multiple version of the same deterministic clustering method in conjunction with resampling techniques to assess the stability of discovered clusters or else using different initialisations of unstable method such as *K*-means clustering [36]. These implementations are conceptually similar to ensemble methods such as Ran-

dom Forest [3] in that they combine many different models, and to Bagging [4] in the dependence upon the instability of the individual models.

The method of inference I propose involves running a large number of short MCMC chains in parallel, which I then combine using a consensus approach. I show that this provides significant computational gains over traditional MCMC approaches, while still enabling meaningful clustering structures to be uncovered and assessments of uncertainty to be performed. Moreover, I show that this approach is better able to explore complex multi-modal posterior distributions as a consequence of the vast number of parallel chains that are used to establish the consensus.

I show by simulation that for the implementation of consensus clustering described above:

1. It is consistent with converged MDI chains;
2. That is is faster than running multiple long chains of MDI;
3. That the space sampled for possible clusterings is sensible when individual chains become trapped; and
4. That the method is robust to different lengths of model chains.

I then apply the method to real biological data with known pathways present and show encouraging results for uncovering this structure. This gene expression data is from the Correlated Expression and Disease Association Research (CEDAR) datasets. Within the CEDAR cohort there are multiple datasets containing information about the same genes for different tissues or cell types; this allows me to implement exploratory analysis of the clustering structure being tissue or cell-type dependent.

2 Theory

2.1 Bayesian inference

Bayesian inference is an alternative paradigm to frequentist methods that has several attractive properties.

1. Principled error qualification; and
2. Integration of prior knowledge and beliefs.

In this project it is point 1 that makes this framework attractive. As stated previously, model uncertainty can represent biological uncertainty.

The keystone of Bayesian inference is Bayes' rule which defines how one can update a hypothesis as more information is made available. For observations X

and a parameter θ where Θ is the entire sample space for θ:

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int_\Theta p(X|\theta')p(\theta')d\theta'} \tag{1}$$

- $p(\theta|X)$ is referred to as the *posterior* distribution of θ as it is the distribution associated with θ *after* observing X.
- $p(\theta)$ is the *prior* distribution of θ and captures the beliefs about θ before observing X.
- $p(X|\theta)$ is the *likelihood* of X given θ, the probability of data X being generated given our model is true. It is the criterion focused on in the model if a frequentist approach to the inference is used (and hence why the frequentist philosophy treats the data as random); maximising this quantity in our model generates the manifold that best describes the observed data.
- $\int_\Theta p(X|\theta')p(\theta')d\theta'$ is the *normalising constant*. It is also referred to as the marginal likelihood as one marginalises the parameter θ by integrating over its entire sample space.

2.2 Clustering

Given some collection of data $X = (x_1, \ldots, x_n)$, let a *clustering* or partition of the data be defined by:

$$\begin{aligned}
\mathcal{Y} &= \{Y_1, \ldots, Y_K\} && (2)\\
Y_k &= \{x_{1_k}, \ldots, x_{n_k}\} && (3)\\
Y_i \cap Y_j &= \emptyset \ \forall \ i, j \in \{1, \ldots, K\}, \ i \neq j && (4)\\
n_k &= \mid Y_k \mid \geq 1 \ \forall \ k \in \{1, \ldots, K\} && (5)\\
\sum_{k=1}^{K} n_k &= n && (6)
\end{aligned}$$

In short there are K non-empty disjoint sets of data, each of which is referred to as a *cluster* or a *component*, the set of which form a *clustering*. A label $c_i = k$ states that point x_i is assigned to cluster Y_k. We define the collection of labels $c = (c_1, \ldots, c_n)$ as denoting the membership of each point.

2.3 Mixture models

My underlying clustering model is a mixture model. These models assume that the data may be described in terms of K clusters defined by some parametric

distribution, $f(\cdot)$. It is assumed that the clusters represent the structure of underlying sub-populations of the dataset. A common probability density function is chosen to represent each cluster. However, the parameters defining the k^{th} cluster and its associated distribution are learnt from the points assigned to the k^{th} cluster; thus while there is a common density function there are cluster-specific parameters.

More formally, if one is given some data $X = (x_1, \ldots, x_n)$, we assume K unobserved sub-populations generate the data and that insights into these sub-populations can be revealed by imposing a clustering $Y = \{Y_1, \ldots, Y_K\}$ on the data. It is assumed that each of the K clusters can be modelled by a parametric distribution, $f(\cdot)$ with parameters θ_k. We let membership in the k^{th} cluster for the i^{th} individual be denoted by $c_i = k$. The full model density is then the weighted sum of the probability density functions where the weights, π_k, are the proportion of the total population assigned to the k^{th} cluster:

$$p(x_i|c_i = k) = \pi_k f(x_i|\theta_k) \tag{7}$$

$$p(x_i) = \sum_{k=1}^{K} \pi_k f(x_i|\theta_k) \tag{8}$$

For our application, we use a Multivariate Normal (MVN) distribution to describe each subpopulation for the pragmatical reason that the Gaussian distribution is easy to work with.

2.3.1 Dirichlet process

The Dirichlet process is an extension of mixture models where $K = \infty$. This concept is used in the implementation of MDI used by Mason et al. [33], the software I use to run MDI. It is mimicked by using an arbitrarily large K value (technically, a finite Dirichlet process; in the model used by Mason et al. [33], $K = 50$) and allowing the model to learn the number of clusters required. This way the value of K is unfixed and learnt from the data, growing and shrinking as required.

2.3.2 Bayesian mixture models

I use Bayesian mixture models. In this case we have a prior distribution on each of the random variables. This allows us to ensure that there is a non-zero probability of a gene being assigned to an empty cluster (whereas under the frequentist paradigm an empty cluster would have an associated weight of 0, and hence the number of occupied clusters has no probability of growing).

Bayesian inference of this model is implemented using MCMC methods. Specifically, Gibbs sampling is employed which can be summarised as iterating between the following steps (the order of which step comes first is arbitrary):

1. For each of K clusters sample θ_k and π_k from the associated distributions based on current memberships, c_i; and
2. For each of n individuals sample c_i based on the new θ_k and π_k.

For more information on MCMC methods generally and Gibbs sampling specifically, please see section A.3.

The output consists of a matrix n_{iter} rows and p columns (for p genes). The i^{th} row describes the cluster the genes are assigned to in the i^{th} iteration of the Gibbs sampler. To summarise this information I use a posterior similarity matrix (PSM). his is a $n \times n$ matrix where the (i, j) entry is the proportion of times genes i and j are in the same cluster. This means that the PSM is not affected by label switching (a problem in Bayesian model-based clustering) and that it is a symmetric matrix with 1's along the diagonal and all entries in the unit interval. (note that for imputed data I set the diagonal entry to -1 for ease of interpretation, please see figure 25 for an example of this).

From this PSM a single clustering estimate, $\hat{c}$, can be described from the PSM by maximising the posterior expected adjusted Rand index [14]. Other methods such as minimisation of Binder's loss function or minimization of Dahl's criterion are based on the original Rand index, and thus are unadjusted for chance. I prefer use of the adjusted Rand index for reasons mentioned in section A.1 and thus choose to use the method described by Fritsch and Ickstadt [14] utilising the posterior expected adjusted Rand index.

2.4 Multiple dataset integration

Consider the case when one has observed paired datasets $X_1 = (x_{1,1}, \ldots, x_{n,1})$, $X_2 = (x_{1,2}, \ldots, x_{n,2})$, where observations in the ith row of each dataset represent information about the same gene. Ideally, one would like to cluster genes using information common to both datasets. One could concatenate the datasets adding additional covariates for each gene. However, if the two datasets have different clustering structures this would reduce the signal of both clusterings and probably one would dominate. If the two datasets have the same structure but different signal-to-noise ratios this would reduce the signal in the final clustering. In both these cases independent models on each dataset would be preferable. Kirk et al. [27] suggest a method to carry out clustering on both datasets where common information is used but two individual clusterings are

outputted. This method is driven by the allocation prior:

$$p(c_{i1}, c_{i2}|\phi) \propto \gamma_{c_{i1}}\gamma_{c_{i2}}(1 + \phi\mathbb{I}(c_{i1} = c_{i2})) \tag{9}$$

Where:

- c_{ij} is the label of the i^{th} gene in the j^{th} dataset;
- $\gamma_{c_{ij}}$ is the component weight of the cluster associated with label c_{ij} in dataset j. $\gamma_{1l}, \ldots, \gamma_{Kl} \sim \text{Ga}(\alpha/K)$ for $l \in 1, 2$ and K is the number of components present;
- Letting $\pi_{il} = \frac{\gamma_{il}}{\sum_{k=1}^{K}\gamma_{kl}}$ then $\pi_{1l}, \ldots, \pi_{Kl} \sim \text{Dirichlet}(\alpha_k/K, \ldots, \alpha_k/K)$ for $l \in 1, 2$ and K is the number of components present (these correspond to the weights used in the mixture model in section 2.3);
- $\phi \in \mathbb{R} > 0$ is the correlation between datasets;
- $\mathbb{I}(c_{i1} = c_{i2})$ is the indicator function - it takes a value of one if c_{i1} and c_{i2} are equal (i.e. a common allocation across datasets) and 0 otherwise.

Here ϕ controls the strength of association between datasets. Equation (9) states that the probability of allocating individual i to component $c_{i,1}$ in dataset 1 and to component $c_{i,2}$ in dataset 2 is proportional to the proportion of these components within each dataset and up-weighted by ϕ if the gene has the same labelling in each dataset. Thus as ϕ grows the correlation between the clusterings grow and we are more likely to see the same clustering emerge from each dataset. Conversely if $\phi = 0$ we have independent mixture models.

The generalised case for L datasets, $X_1 = (x_{1,1}, \ldots, x_{n,1}), \ldots, X_L = (x_{1,l}, \ldots, x_{n,l})$ for any $L \in \mathbb{N}$ is simply a matter of combinatorics. In this case, (9) extends to:

$$p(c_{i1}, \ldots, c_{iL}|\boldsymbol{\phi}) \propto \left[\prod_{l_1=1}^{L}\gamma_{c_{il_1}l_1}\right]\left[\prod_{l_2=1}^{L-1}\prod_{l_3=l_2+1}^{L}(1 + \phi_{l_2l_3}\mathbb{1}(c_{il_2} = c_{il_3}))\right] \tag{10}$$

Here $\boldsymbol{\phi}$ is the $\binom{L}{2}$-vector of all ϕ_{ij} where ϕ_{12} is the variable ϕ in (9).

Thus MDI is an extension of mixture models to multiple datasets where correlated clustering structure is used to "up weigh" similar clusters across datasets. MDI has been applied to precision medicine, specifically identifying function modules of genes and glioblastoma sub-typing [47], in the past showing its potential as a tool.

2.4.1 Prior distributions

The priors on the MDI model are considered in two steps. First, the priors of the parameters in (10), and second, the parameters specific to the underlying Gaussian densities.

For the parameters in (10):

- The prior on the ϕ_{ij} is $\text{Ga}(1, 0.2)\ \forall\, i, j \in (1, \ldots, L)$;
- $\text{Ga}(\alpha_l / K, 1)$ is the prior on the $\gamma_{k_l l}\ \forall\, l \in (1, \ldots, L), k \in (1, \ldots, K)$;
- The prior on the α_l is $\text{Ga}(2, 4)\ \forall\, l \in (1, \ldots, L)$;

For the Gaussian density, it is assumed that each component is modelled by a Gaussian likelihood of unknown mean and precision. A Normal-Gamma prior is imposed upon these. The density can therefore be represented in closed form and the mean and precision may be marginalised (one of the reasons the Gaussian distribution is easy to work with as referenced in section 2.3). This leaves the following marginal likelihood function for a component with n genes allocated to it:

$$f = \frac{\Gamma(\alpha_n)}{\Gamma(\alpha_0)} \frac{\beta_0^{\alpha_0}}{\beta_n^{\alpha_n}} \left(\frac{\kappa_0}{\kappa_n}\right)^{\frac{1}{2}} (2\pi)^{-\frac{n}{2}} \tag{11}$$

Where:

$$\mu \sim \mathcal{N}(0, (\kappa_0 \lambda)^{-1}) \tag{12}$$
$$\lambda \sim \text{Ga}(\alpha_0, \beta_0) \tag{13}$$
$$\kappa_n = \kappa_0 + n \tag{14}$$
$$\alpha_n = \alpha_0 + \frac{n}{2} \tag{15}$$
$$\beta_n = \beta_0 + \frac{1}{2} \sum_{i=1}^{n} (x_i - \bar{x})^2 + \frac{\kappa_0 n \bar{x}^2}{2\kappa_n} \tag{16}$$

The hyperparameters are set:

$$\alpha_0 = 2 \tag{17}$$
$$\beta_0 = 0.5 \tag{18}$$
$$\kappa_0 = 0.001 \tag{19}$$

These are intended to be broad, uniformed priors (recalling that the data has been standardised - this is particularly relevant to the mean of 0 for the μ parameter). This allows the model to infer from the data without an overtly strong influence from the priors.

2.5 Consensus clustering

In the scenario that MDI struggles to explore the entire posterior distribution from any given initialisation for any realistic number of iterations of MCMC, I

propose use of a "consensus clustering" [36] to perform approximate Bayesian inference. In this scenario I draw samples of clusterings from MCMC chains with different initialisations and use these clusterings to describe the target distribution. In practice this involves running n_{seeds} different chains of MDI for a smaller number of iterations, n_{iter}, and burning out the first $n_{iter} - 1$ iterations. The clustering from the final iteration is then saved for this model.

I then combine the clusterings from all n_{seeds} within a PSM for the n genes. From this PSM a summary clustering may be calculated. The combination of different initialisations enables exploration of multiple maxima in the posterior density and thus provides a more informed clustering than a method liable to become trapped in a single mode.

There have already been numerous applications of consensus clustering [30] [29] [3]. However, none of these previous implementations have used a Bayesian model. I show the validity of my implementation of consensus clustering by means of simulations. In this case I know the true clustering as I have chosen which points are drawn from which sub-populations I can then compare the quality of recorded clusterings generated by a single converged chain of MDI to different version of consensus clustering (i.e. varying n_{iter}). I let the quality of a clustering be defined by its similarity to the ground truth, measured using the *adjusted Rand index* (see section A.1).

3 Case study examples

I first show via simulated data that consensus clustering does produce similar results to a converged single run for MDI.

I then simulate data where individual chains of MDI are expected to struggle to converge and possibly will not converge in finite time. I show that consensus clustering explores a wider space than any individual chain and appears to describe something similar to the space described by the union of the chains.

Finally I apply consensus clustering to two sets of probes for 7 tissue or cell-type specific datasets from the CEDAR dataset; a set of 250 probes with a single KEGG pathway present and a set of 1,000 probes with three KEGG pathways present.

3.1 Simulations

3.1.1 Simulation: Case 1

The data in the first simulation is designed to allow MDI to converge. In this case I take the data generated for the original MDI paper [27]. As this data is

highly separable I add some noise to add some uncertainty to the clustering.

In this case I have 3 datasets (MDItestdata1, MDItestdata2 and MDItestdata3). I use MDItestdata1 as the basis to define new data. I generate two overlapping clusters (cluster A and B) defined by two of the original clusters (cluster 1 and 2). I define cluster A to be generated from a MVN distribution with a mean defined by the weighted means of clusters 1 and 2 and a variance defined by the weighted variance of these same clusters. For cluster A the relative weights are 0.6 and 1 for clusters 1 and 2. Cluster B is defined in the same way, but the weights are reversed such that cluster B is more similar to cluster 1.

3.1.2 Simulation: Case 2

The data in the second simulation is designed to be a testing example in which MDI might struggle to converge. This data is based upon 5 clusters of $n_{clust} = \{25, 50, 75, 100, 150\}$ genes (let n_{clust_i} be the number of genes in each the i^{th} cluster) for $p = 400$ people. Each cluster is defined by a MVN distribution with common variance of 1. I then perturb the clusters, adding a small amount of noise generated from a normal distribution of mean 0 and standard deviation 0.1. This noise makes the clusters less distinct. I generate 3 datasets this way, varying the means defining the clusters between datasets. Specifically each subpopulation is defined by combining p samples for n_{clust_i} genes where each sample is pulled from the (i, j) entry of table 1 (where i is the cluster number and j is the dataset number), and perturbed by a random sample drawn from $\mathcal{N}(0, 0.1)$. One can see that the underlying structure should be easier to uncover in dataset 2 and harder to uncover in dataset 3 (due to the distance between means of sub-populations).

Subpopulation	Dataset 1	Dataset 2	Dataset 3
1	$\mathcal{N}(1,1)$	$\mathcal{N}(1,1)$	$\mathcal{N}(1,1)$
2	$\mathcal{N}(2,1)$	$\mathcal{N}(3,1)$	$\mathcal{N}(1.5,1)$
3	$\mathcal{N}(4,1)$	$\mathcal{N}(6,1)$	$\mathcal{N}(3,1)$
4	$\mathcal{N}(6,1)$	$\mathcal{N}(9,1)$	$\mathcal{N}(4,1)$
5	$\mathcal{N}(8,1)$	$\mathcal{N}(12,1)$	$\mathcal{N}(5,1)$

Table 1: Sub-populations defining the simulated data in case 2.

3.2 CEDAR dataset

I use the gene expression data from the CEDAR cohort [51]. This data is available in a processed form online. This consists of 9 .csv files, one for each tissue

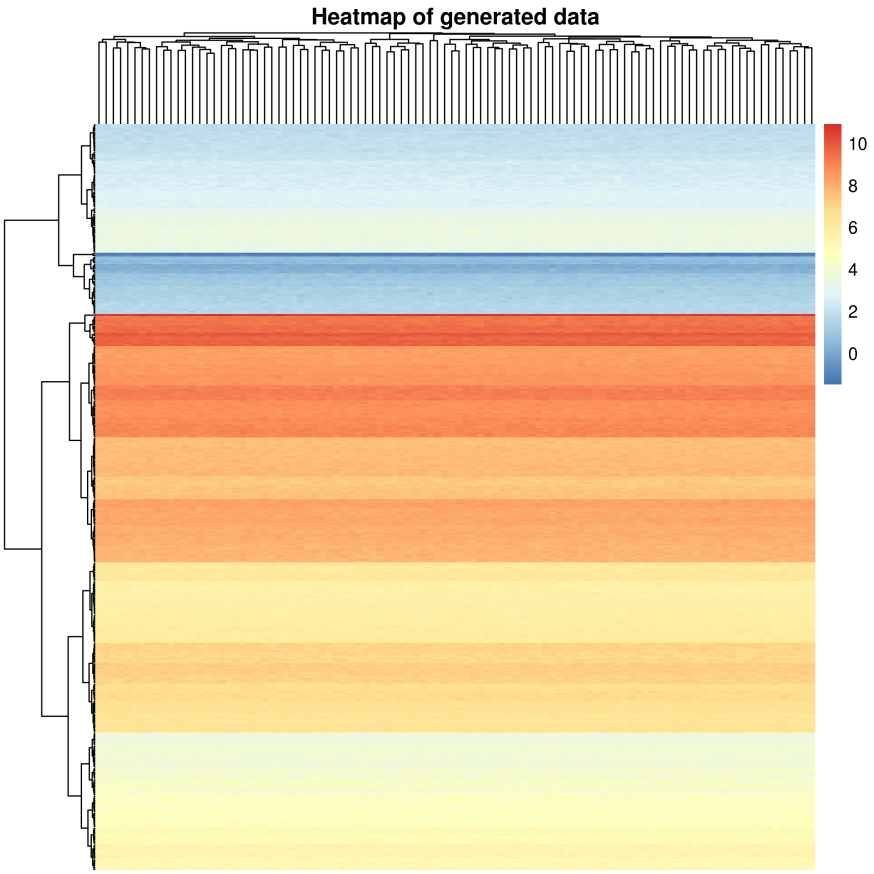

Figure 1: Heatmap of expression data generated for the second simulation as described in section 3.1.2. Note that there are 5 populations present here and that cluster membership and boundaries are not obvious.

/ cell type sampled, of gene expression data for 323 individuals. These are healthy individuals of European descent; the cohort consists of 182 women and 141 men with an average age of 56 years (but ranging from 19 to 86). None of the individuals were suffering from any autoimmune or inflammatory disease and were not taking corticosteroids or non-steroid anti-inflammatory drugs (with the exception of aspirin).

With regards to tissue types, samples from six circulating immune cells types (followed in brackets by the abbreviation for the associated dataset):

- CD4+ T lymphocytes (CD4);
- CD8+ T lymphocytes (CD8);
- CD14+ monocytes (CD14);

- CD15+ granulocytes (CD15);
- CD19+ B lymphocytes (CD19); and
- platelets (PLA).

Data from intestinal biopsies are also present, with samples taken from three distinct locations:

- the ileum (IL);
- the rectum (RE); and
- the colon (TR).

Not every individual is present in every dataset. However, as genes are the object of interest this should not present a problem.

Whole genome expression data were generated using HT-12 Expression Beadchips following the instructions of the manufacturer (Illumina). There are 18,524 probes present between the 9 datasets. It should be noted that there are differing degrees of missingness between the datasets (for instance the platelets dataset has 6,564 probes present in comparison to an average of 12,838 probes present per dataset, see figure 2).

Due to exponential increase in computational cost for each additional dataset, I only use the 7 most informative datasets, dropping PLA and CD15 from the analysis. Furthermore, from a biological perspective I expect PLA to be the least information rich as platelets have no nucleus [57] and therefore any gene expression is an artefact from before they differentiated into platelets. With regards to CD15+ granulocytes, (mast cells, basophils, neutrophils and eosinophils), these are quite distinct from B and T lymphocytes (see figure 3). Based on this I expect there to be less common information pertinent to clustering genes in other datasets. Arguably monocytes are equally distant, but the level of missingness in the CD15 dataset is greater than that in the CD14 dataset; thus CD15 is elimanted from our analysis.

I create two subsets of CEDAR data defined around KEGG pathways and random other probes. This is used to test if the annotated gene sets are identifiable using this method in real data. Implicit in this decision is the assumption that KEGG pathways, which are about protein interactions, can be captured by transcription data, and specifically transcription data with no repeated measurements.

3.2.1 CEDAR: Case 1 - Inositol gene set

I create an example dataset of 250 probes from the CEDAR dataset. This dataset contained 60 probes from the Inositol phosphate metabolism pathway as de-

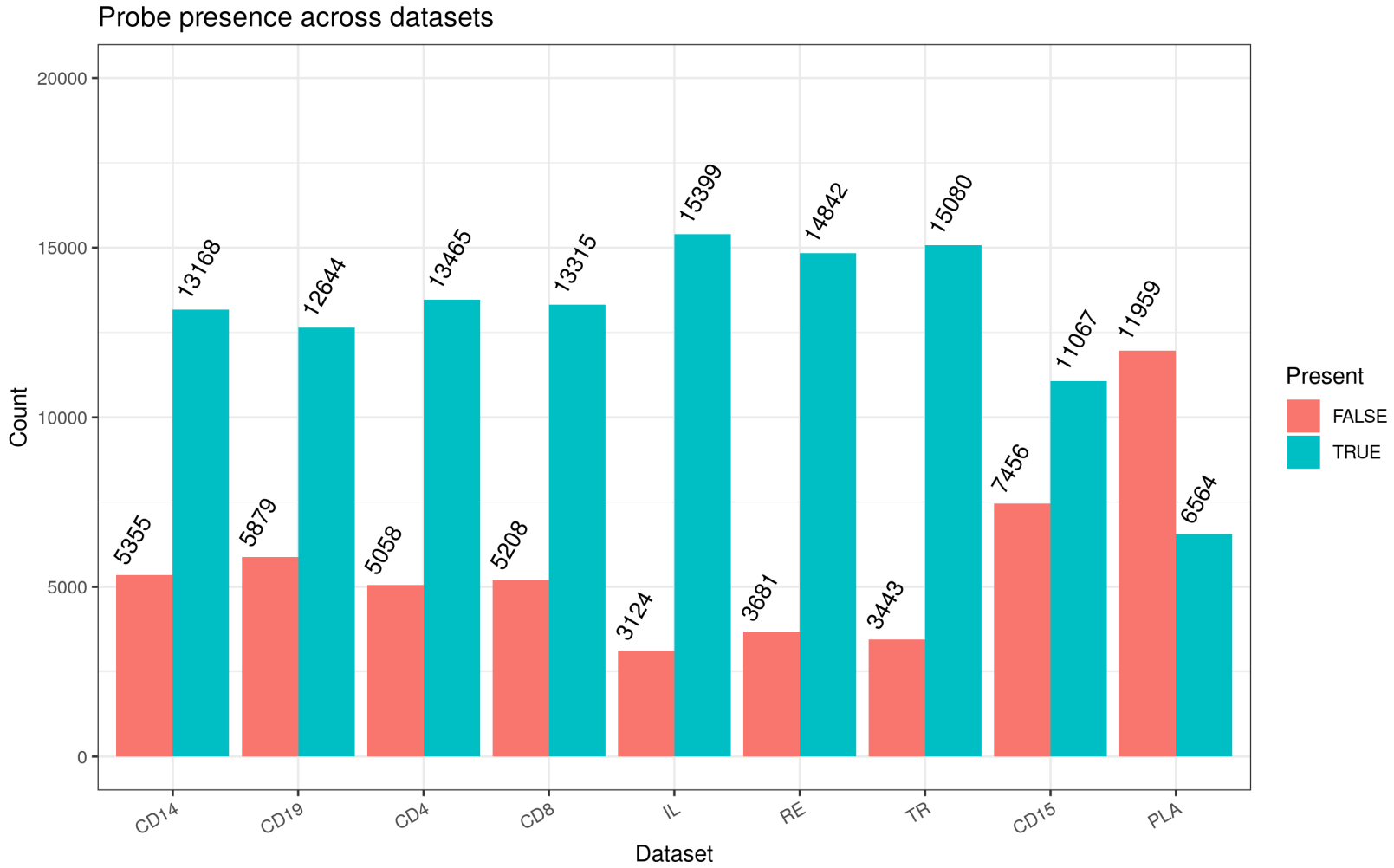

Figure 2: Probe presence across datasets. Note that the number of probes missing is greatest in PLA, followed by CD15.

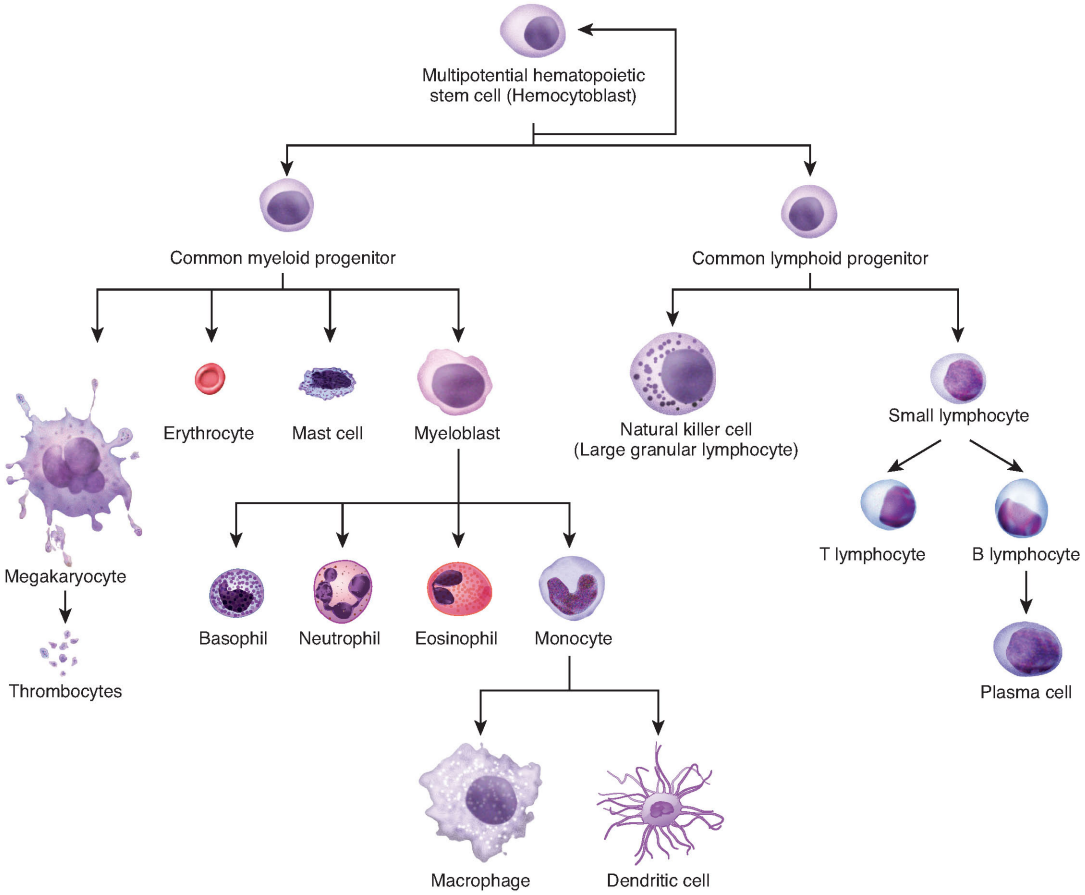

Figure 3: The differentiation of multipotent cells into blood and immune cells. Image courtesy of the OpenStax project [41].

fined in the KEGG database. This scale of dataset allows implementation of MDI across 7 datasets simultaneously.

3.2.2 CEDAR: Case 2 - 1,000 probes

A second dataset of 1,000 probes defined by three KEGG pathways was used to explore the clustering on a larger, more diverse dataset. The pathways used are:

1. Inositol phosphate metabolism (a broad biological pathway);
2. NOD-like receptor signalling pathway (a specific biological pathway with known involvement in IBD [6][15]); and
3. Inflammatory bowel disease (IBD) (a pathological pathway).

The union of these sets corresponds to 169 unique genes (or 237 probes as the mapping from the space of probes to that of genes is non-injective) that are present in the CEDAR dataset. The remaining probes are randomly selected from the total possible space (18,524 probes) less those corresponding to these genes (leaving 18,287 possible candidate probes).

3.2.3 Pipeline

For the CEDAR data, I follow this pipeline to prepare the data for clustering:

1. Transpose the data to have rows associated with gene probes and columns associated with individuals;
2. Remove NAs either imputing values using the minimum expressed value (as missingness is not random) or if above a threshold of missingness removing the column;
3. Standardise the data (see section A.2); and
4. For probes entirely missing from a given dataset I generate expression from a standard normal distribution for each probe. Then these probes are expressed as noise in the dataset and any clustering imposed upon them should be due to information about these probes present in other datasets.

4 Results

4.1 Simulations

4.1.1 Case 1: Proof of consensus

Using the data generated as described in section 3.1.1, I ran 10 separate chains of MDI for 2 million iterations with a thinning factor of 50 (this took approximately 28 hours). Secondly, consensus clustering models of 1,000 different seeds for 4 different lengths of chains were applied to the same data. The time for each individual chain is shown in table 2.

The estimated burn-in time required is checked by means of the effective sample size of MCMC samples drawn (see figure 4).

After implementing the required burn-in across all chains, results are compared by means of an autocorrelation plot, a Geweke plot [18], a Gelman plot [16] and the distribution of adjusted Rand index comparing the clustering at each iteration (or each seed in the consensus clustering) with the clustering defined by the sub-populations used to generate the data. These were shown to

Chain length (n_{iter})	Time (hh:mm:ss)
500	00:02:00
1,000	00:03:30
5,000	00:10:00
10,000	00:20:00
2,000,000	28:30:00

Table 2: Sub-populations defining the simulated data in case 2.

have converged successfully both within chain, by inspecting an autocorrelation plot and a Geweke plot (see figures 5 and 6), and across chains, by means of the Gelman-Rubin convergence statistic (see figure 7). The clustering at each iteration was compared to the the clustering defined by the sub-populations that generated the data and the long chains were compared to each other and consensus clustering for various chain lengths (see figure 8). One can see that the consensus clustering perform very similarly to the individual chains. Finally, in figure 9, the full space of clusterings across all chains is compared to the consensus clustering under the adjusted Rand Index.

4.1.2 Case 2: Overcoming multiple modes

Similar versions of consensus clustering and individual chains were run for the data generated as described in section 3.1.2. These were then compared using the same methods. The burn-in required by the slowest converging chain was 20,000 recorded samples. This was applied to each chain before investigating convergence, but I include an example (figure 11) of the auto-correlation prior to burn-in. One can see an example in both figures 12 and 13 that indicates the individual chain has reached stationarity (the individual chain is no longer exploring new space). However, convergence has not been achieved across chains as is shown in figure 14. The space explored across the ten chains is compared to that explored in each of the consensus clusterings (see figures 15, 16 and 17).

4.2 CEDAR data

4.2.1 Case 1: 250 probes

Consensus clustering with MDI $n_{iter} = 500$ and $n_{seeds} = 1000$ was implemented on 7 datasets. The datasets were defined as described in section 3.2.1. Individual mixture models were also run on each dataset for the same number of seeds and iterations as a comparison. Each chain of MDI took approximately 7 hours and 20 minutes to run. The output was inspected under multiple lenses:

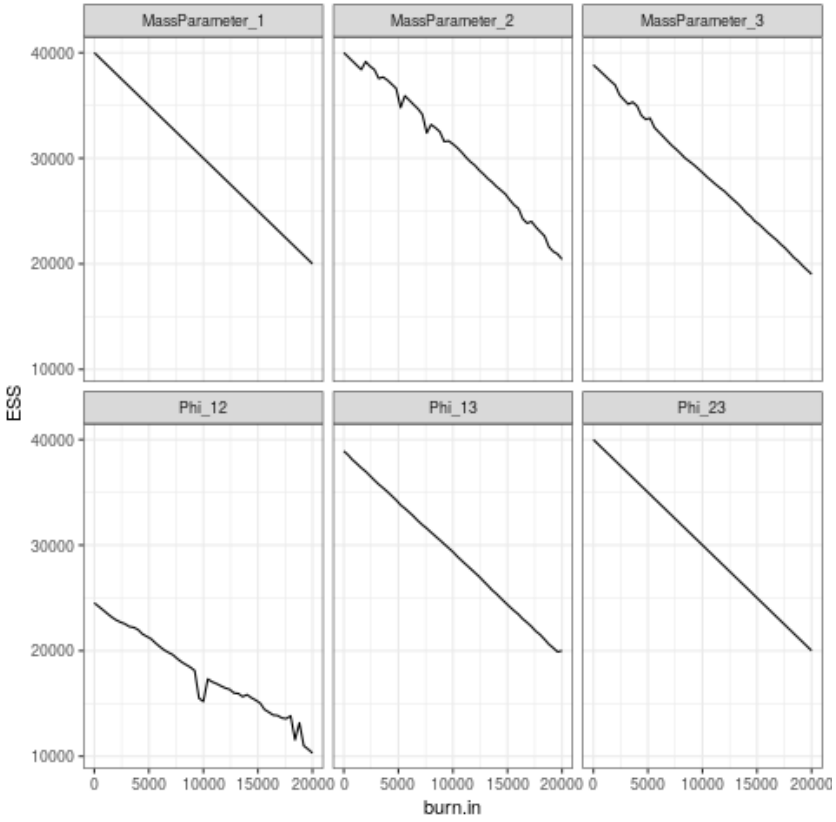

Figure 4: An estimate for burn-in cut-off is estimated by plotting the effective sample size (ESS) against the burn-in at different iterations (shown here for seed 3 in case 1 of the simulations). ESS is the number of independent samples equivalent to the number of autocorrelated samples. The ESS should be maximised at the optimal estimate of the burn-in - thus one looks for the iteration at which ESS is maximised for the most parameters. Here it is a iteration 0. For a more thorough description of ESS, please go to section A.4.3.

1. The adjusted Rand index between i^{th} and $1,000^{th}$ clusterings were plotted for all $i \in \{1, \ldots, 1000\}$ (see an example in figure 18);
2. The mean adjusted Rand index comparing clusterings across datasets were represented in a heatmap (see figure 19);
3. The ϕ_{ij} values were plotted across seeds for all combinations of datasets (see an example in figure 20);
4. The distribution of ϕ_{ij} values were plotted for all combinations of datasets (see an example in figure 21);
5. The mean ϕ value between datasets are represented in a heatmap (see

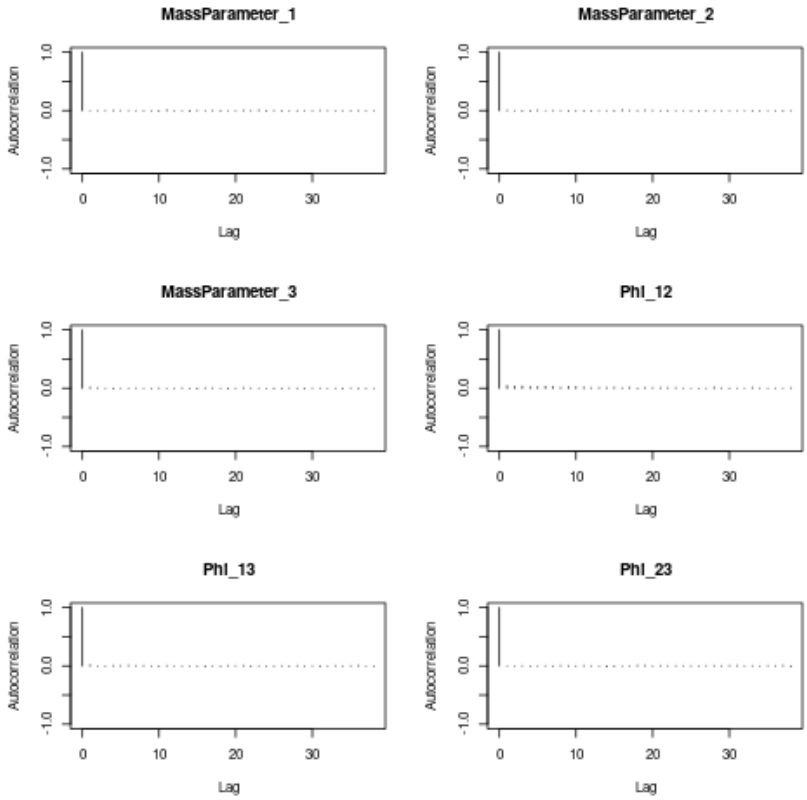

Figure 5: Autocorrelation plot for a long chain with random seed 3 in case 1 of the simulations. The autocorrelation at lag 0 is always 1 (as the observation correlates with itself). If the samples drawn are no longer correlated with samples at any lag visible here (recalling that a lag of 1 corresponds to 50 iterations due to the thinning factor) then this indicates the chain has achieved stationarity. For more discussion of autocorrelation, please read section A.4.3.

figure 22);

6. The number of clusters present in any given seed were plotted for each dataset (see an example in figure 23);
7. The mass parameter for the underlying mixture models were plotted across seeds for each dataset (see an example in figure 24);
8. The PSM for each dataset was plotted as a heatmap (see an example in figure 25); and
9. The comparison of the PSM, the standardised expression data and the correlation matrix were plotted with a common row ordering for each dataset (see an example in figure 26).

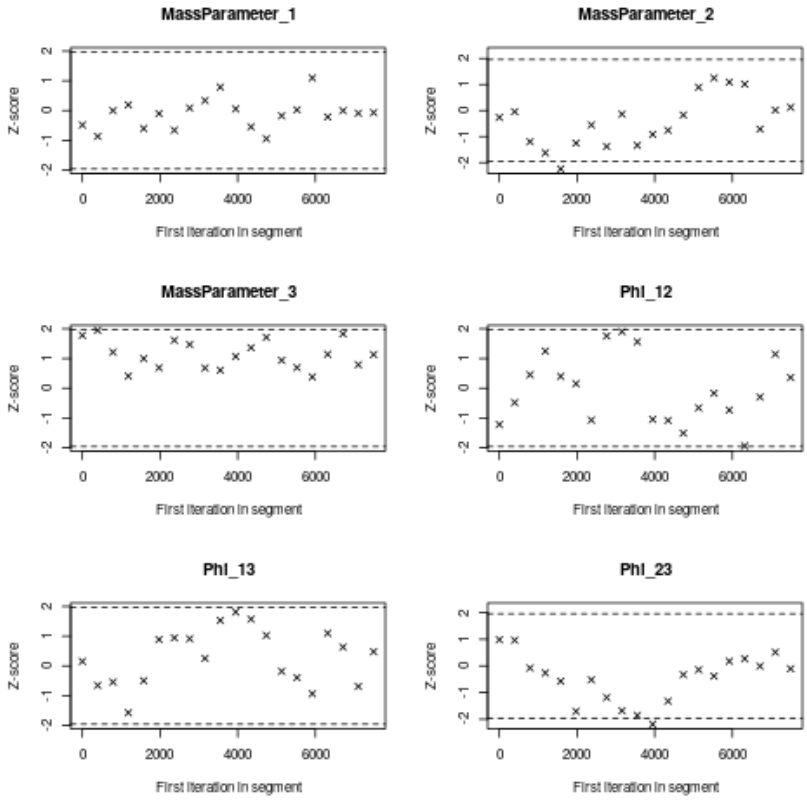

Figure 6: Geweke plot [18] for a long chain with random seed 3 in case 1 of the simulations. This plot is generated using the plot.geweke function from the rjags R package [43]. The samples are split into disjoint bins, with the first set of $\frac{n}{2(n_{bins}-1)}$ samples in the first bin, the second set of $\frac{n}{2(n_{bins}-1)}$ samples in the second, etc. with the final bin containing the final 50% of samples. Geweke's Z-score [18] is repeatedly calculated, comparing the members of each bin to the final 50% of samples with the Z-scores than plotted as above. If the chain has reached stationarity the Z-scores should be described by a standard normal distribution (in which case 95% of recorded values should be contained within the dashed lines). For more information, please see section A.4.3. This is the case here, indicating, particularly in conjunction with figure 5, that the chain has converged.

Note that some of these (such as the ϕ_{ij} plots only apply to MDI, not the mixture models as this is the single dataset case and comparisons across datasets are not possible).

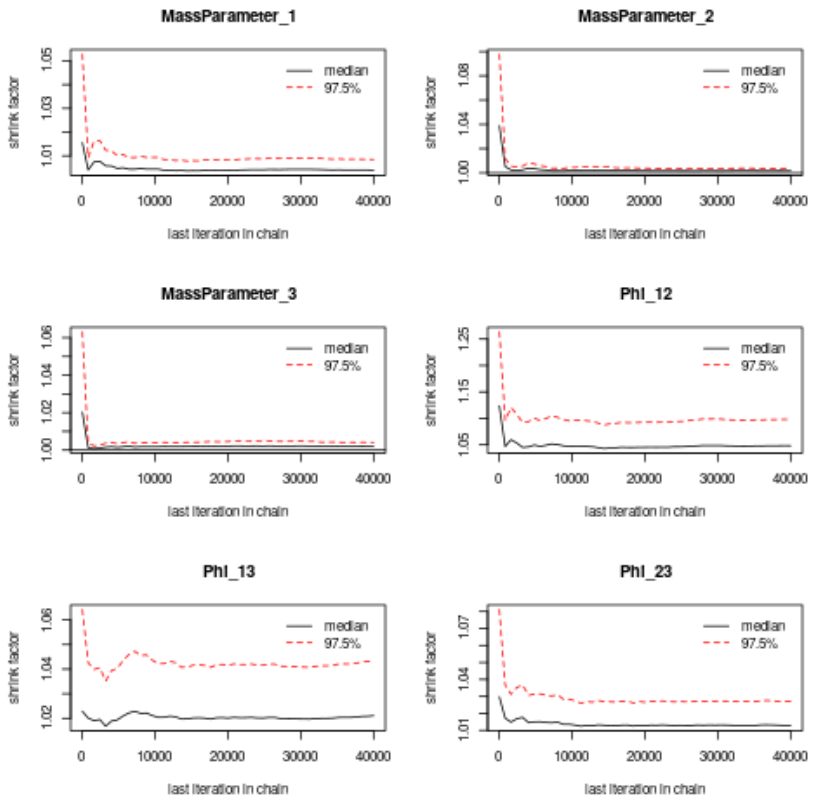

Figure 7: Plot of the shrink factor described by Gelman and Rubin [16] for the continuous variables across chains in case 1 of the simulations. If the chains are converged (and thus describing similar spaces to one another) the values should tend to 1 (as is the case here). For a more thorough description of the shrink factor, please see section A.4.4.

The clustering for the KEGG pathway was inspected using three visualisation techniques:

1. The annotated PSMs of the dataset and the subset of data from the pathway was plotted (see figure 27 for an example of this);
2. For a pathway of m members, I sampled m random genes not in this pathway and found the mean probability of the pairwise alignment of these genes (i.e. the proportion of seeds for which any two of the m genes had the same labelling). Taking n of these samples allowed us to describe the distribution of the mean pairwise alignment probability and compare with the mean pairwise alignment probability of the m genes from the pathway of interest (for an example, please see figure 28); and

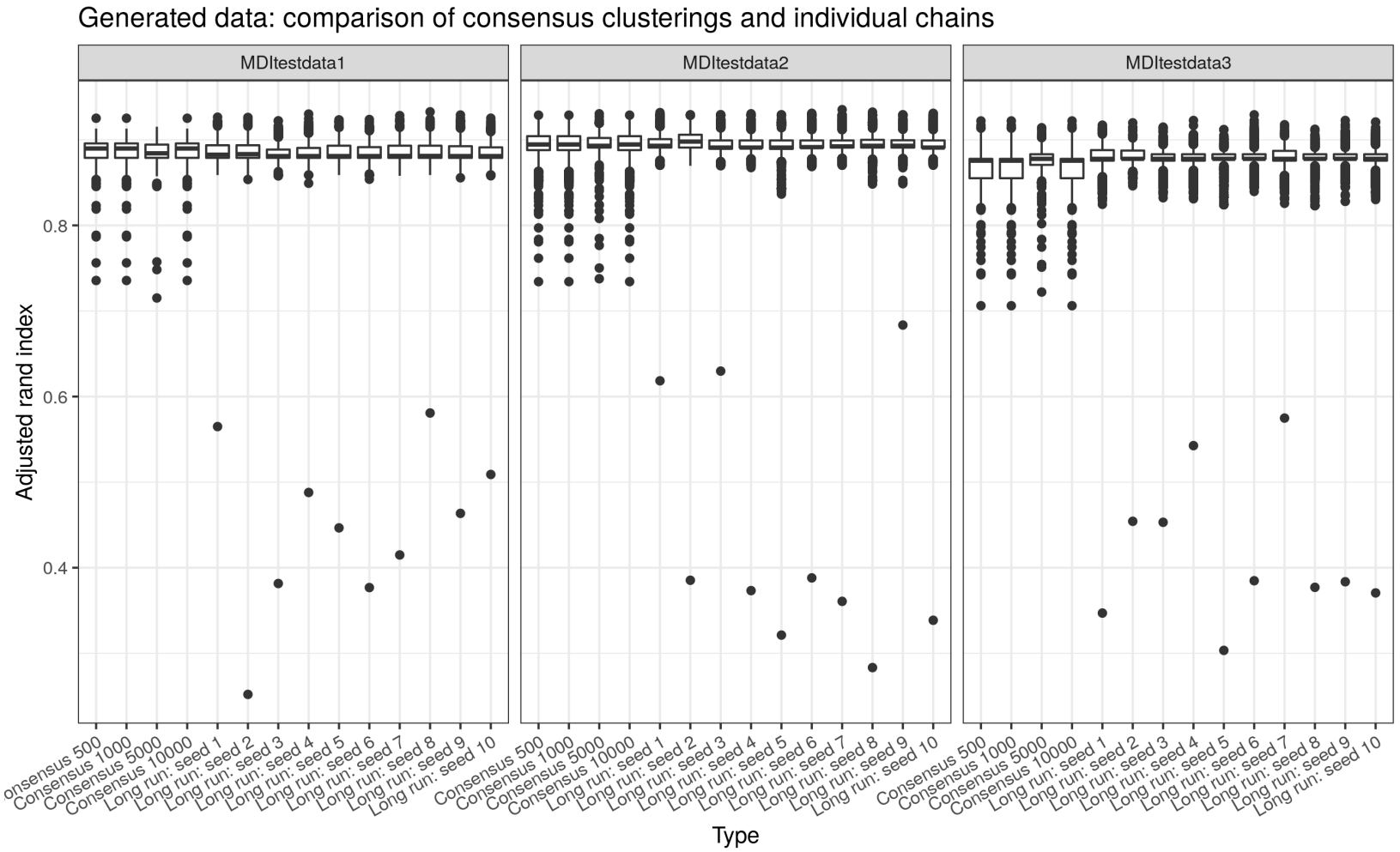

Figure 8: Box plots for distribution of adjusted rand index between the clustering at each iteration to the true clustering for different lengths of consensus clustering and different initialisation of long chains.

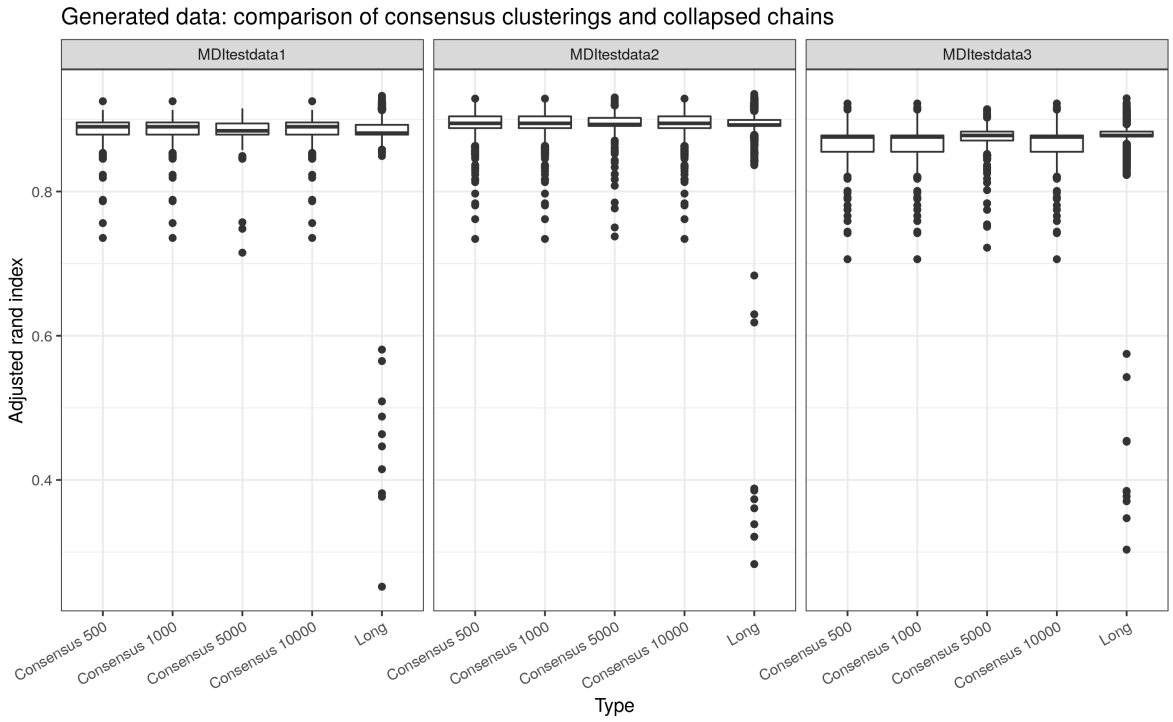

Figure 9: Box plots for distribution of adjusted rand index between the clustering at each iteration to the true clustering for different lengths of consensus clustering and the collapsed long chains.

3. The violin plots of the PSM entries for the pathway were compared to the PSM entries for the remaining genes in the dataset (see the plot depicted in figure 29 for an example of this).

I include a direct comparison of the PSM from MDI to the single dataset case in figure 30.

4.2.2 Case 2: 1,000 probes

Consensus clustering with MDI $n_{iter} = 500$ and $n_{seeds} = 1000$ was implemented on 7 datasets. The datasets were defined as described in section 3.2.2. The analysis followed the same pipeline as described in section 4.2.1.

5 Conclusion

It can be seen that when MDI does converge and successfully samples from the posterior space (section 4.1.1), consensus clustering samples from the same

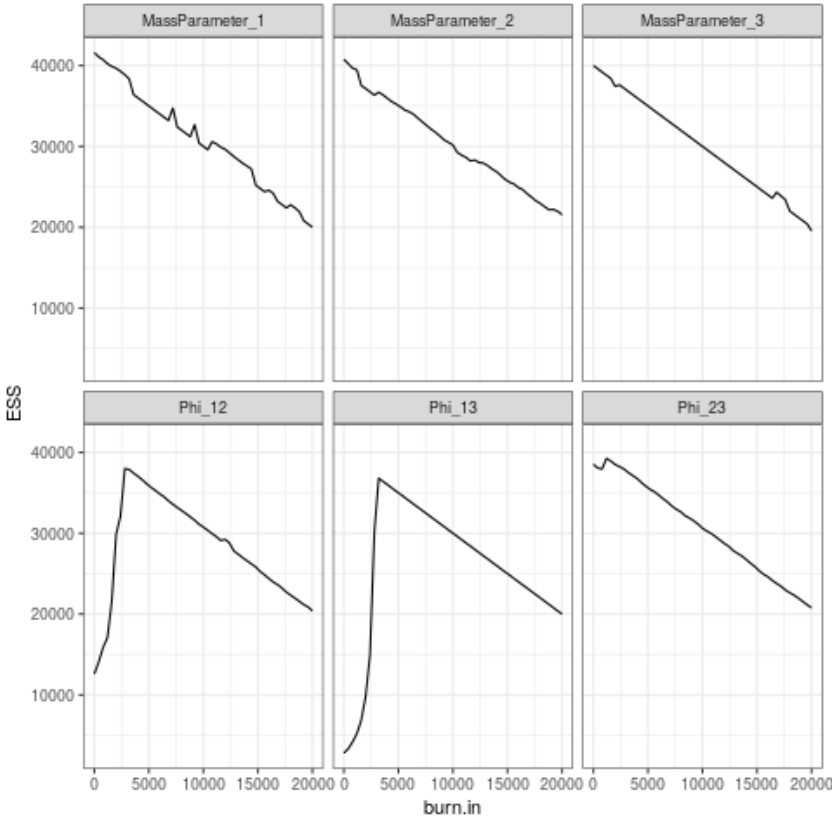

Figure 10: Plot of ESS against burn-in at different iterations for a long chain with random seed 1 in case 2 of the simulations.

space and performs very similarly in terms of describing the underlying structure of the data. The results in section 4.1.2 show that even when MCMC methods struggle to converge, consensus clustering offers a description of the space of interest. Consensus clustering captures multiple modes in a similar distribution to the space described across all chains of MDI (see figure 17). The consensus clustering also appears to be robust in terms of the number of iterations used in each individual chain (each consensus clustering length performs identically). As each chain used in Consensus clustering is independent of the others, the problem is embarrassingly parallel; therefore 1,000 chains of 500 iterations is far quicker to run in parallel than a single long chain. This means that even when multi-modality is not expected to be an issue for MCMC methods that consensus clustering is a useful alternative based purely on the reduction in run-times.

The combination of these results encourages the claim that consensus clustering is a quick, robust solution to the problem of scaling Bayesian cluster-

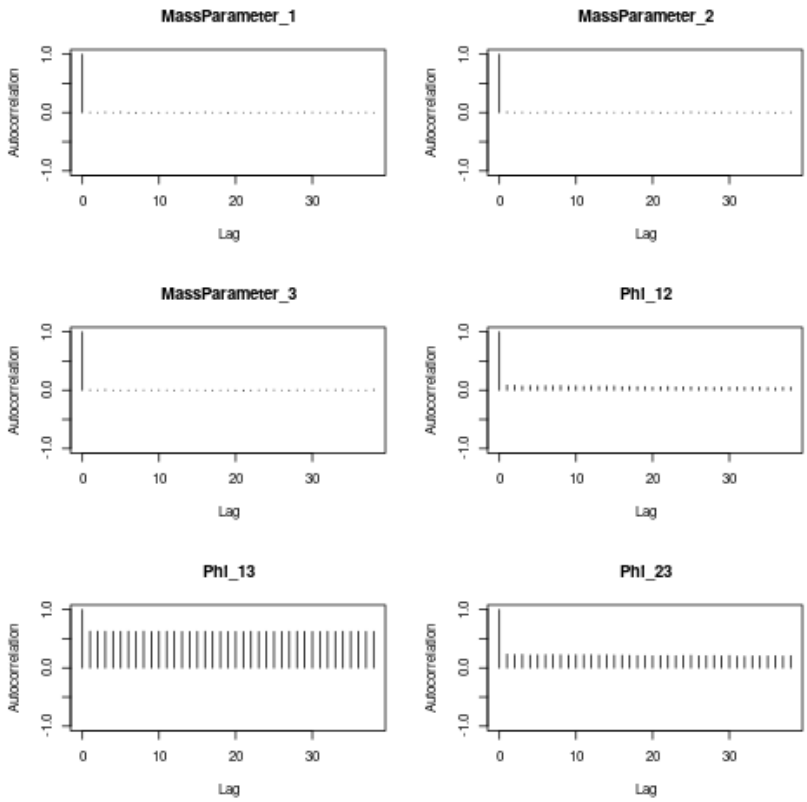

Figure 11: Autocorrelation plot for a long chain with no burn-in. The high lag present for Phi_13 indicates that the chain has not achieved stationarity.

ing methods generally and of multi-modality specifically. Consensus clustering produces a more accurate description of the posterior distribution than any single chain is capable of in a multi-modal high dimensional space within a realistic number of iterations; thus this implementation has many of the advantages of Bayesian inference (the use of priors, quantification of error) but overcomes the limitations of convergence and speed. It should be noted that these limits of Bayesian clustering methods apply to any field where high dimensional, multi-modal datasets are used and thus this implementation of consensus clustering has applications ranging beyond the case studies described here.

Applied to the CEDAR case studies, consensus clustering has positive results. The agreement between the PSM and the correlation matrix that can be seen in figure 26 is reassuring - it shows that the clustering imposed is in line with the data. Furthermore, the results displayed in figures 28 and 41 are encouraging. It looks like my model has successfully uncovered some of the structure of a pathway. This is supported by the difference between the PSM entries

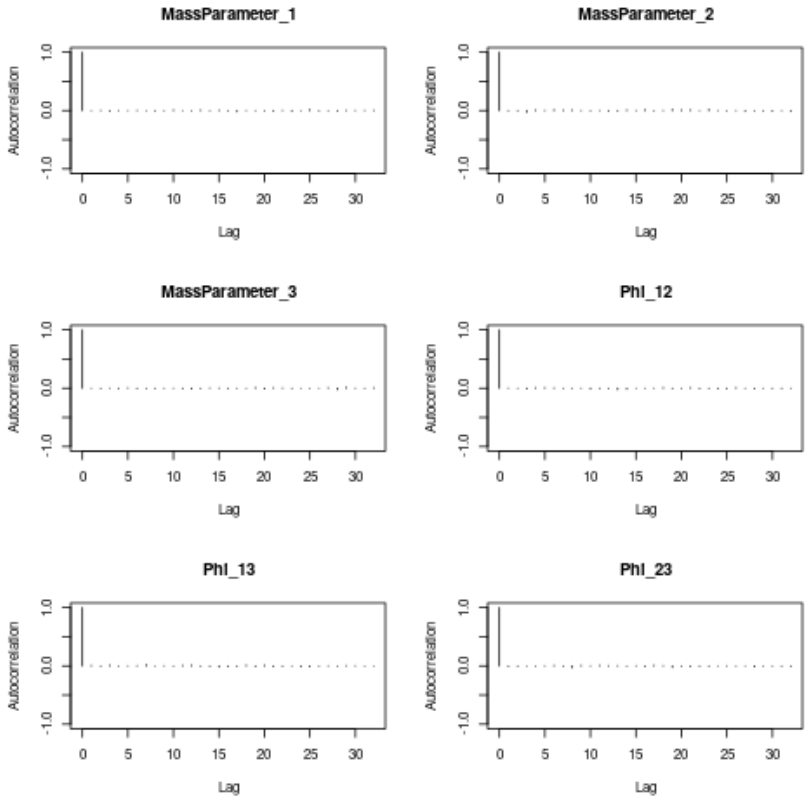

Figure 12: Autocorrelation plot for a long chain with random seed 1 in case 2 of the simulations after a burn-in of 20,000 samples. This result is indicative of stationarity within a chain (low lag values).

for the associated probes compared to the non-pathway probes as can be seen in figure 29. The contrast between figures 41 and 42 is also encouraging of the thesis that tissue annotation is important for pathways, as the IBD pathway is uncovered with some success in the IL dataset, but none at all in the CD14 dataset. This is as one might expect - the colonic samples are pieces of tissue, thus there are a range of cells present here, including auto-immune cells. These auto-immune cells, which mediate IBD, are in the environment where IBD manifests, thus I expect to see the IBD pathway here, whereas the auto-immune cell datasets could be from any location, and thus I do not expect to see a tissue specific disease pathway present.

We can see the benefits of using MDI models in figures 30, 19, 22, 32 and 34. The first, figure 30, shows that MDI is more confident in allocating probes together. This is due to the additional information available to the model through the other datasets. The other plots, figures 19, 22, 32 and 34, show that MDI can

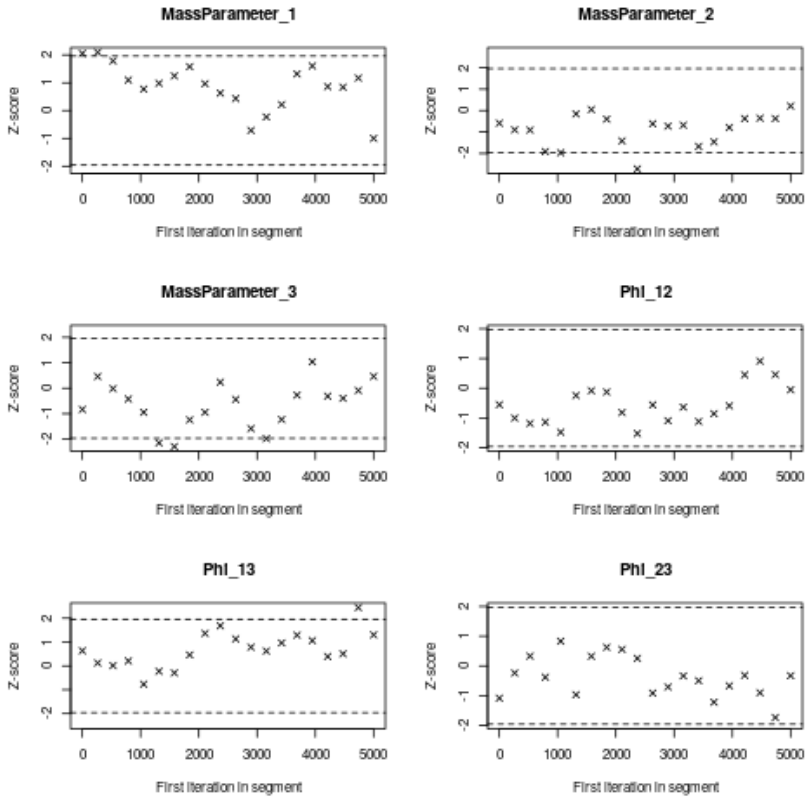

Figure 13: Geweke plot for a long chain with random seed 1 in case 2 of the simulations. 95% of the points are between the dashed lines, indicating stationarity within the chain.

be used to quantify the similarity in structure of the datasets, something individual mixture models cannot do. As one would expect, these show that the three colon samples are quite similar and the CD datasets are similar, with little information shared across these sets of datasets. We can also see that the CD4 and CD8 datasets are highly correlated (the high mean ϕ value across seeds), as one would expect from two types of T lymphocyte.

6 Future work

With regards to consensus clustering there are several avenues to explore:

1. How many seeds are required?
2. How short can the individual chains be?

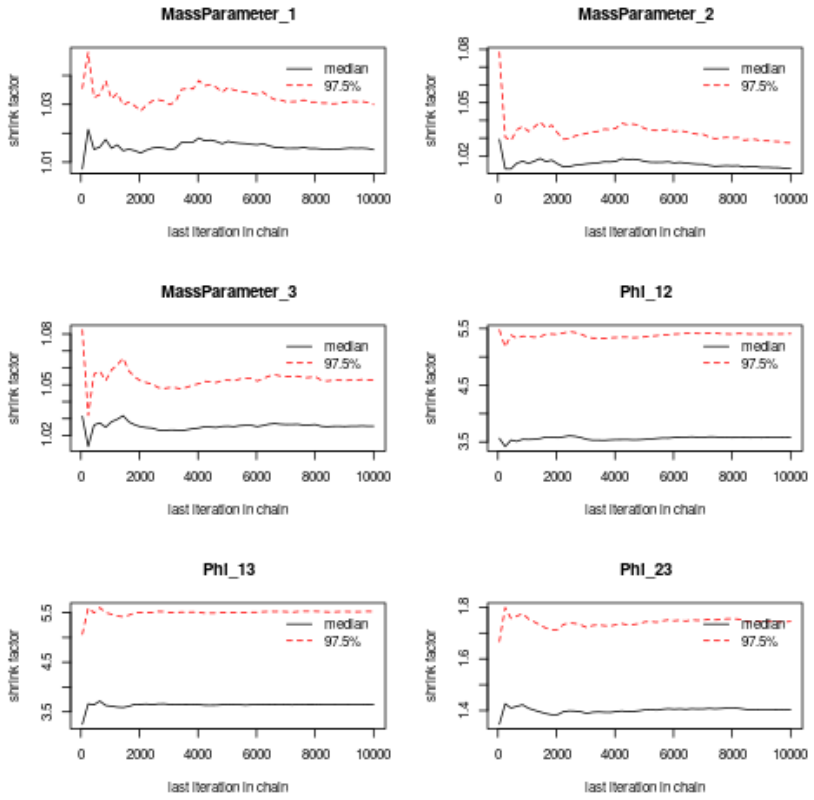

Figure 14: Plot of the Gelman-Rubin shrink factor for the continuous variables across chains in case 2 of the simulations. The behaviour of the shrinkage factor for the ϕ_{ij} parameters indicates that convergence across chains has not been achieved.

3. How does this extend to other clustering methods?

The first two points would be expected to depend on characteristics of the dataset in question, but some suggestions could be drawn from well designed simulations.

For the application of annotating gene sets with tissue and cell-type specific information, the encouraging results from section 4.2 suggest further work to integrate tissue-specific information into definitions of gene sets could be rewarding. Different datasets might also be of use; for instance datasets with repeated measurements or proteomics datasets might offer more information of interest to define gene sets. This is one of the advantages of MDI, the different datasets used do not have to all be the same kind of data as long as the row names are the same. Even different types of data, such as categorical, can be

integrated in the clustering. As MDI allows the ϕ_{ij} parameter to go to 0 if there is no correlation, one can use datasets on thinks might be relevant without a fear of disrupting the signal present in the individual datasets.

References

[1] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000. ISSN 1061-4036, 1546-1718. doi: 10.1038/75556.

[2] Stefan Milton Bache and Hadley Wickham. magrittr: A forward-pipe operator for r, 2014. URL https://CRAN.R-project.org/package=magrittr. R package version 1.5.

[3] Leo Breiman. Random Forests. *Machine Learning*, 45(1573-0565):5–32, January 1. doi: 10.1023/A:1010933404324.

[4] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00058655.

[5] Robert L. Brennan and Richard J. Light. Measuring agreement when two observers classify people into categories not defined in advance. *British Journal of Mathematical and Statistical Psychology*, 27(2):154–163, November 1974. ISSN 00071102. doi: 10.1111/j.2044-8317.1974.tb00535.x.

[6] Lam Carneiro, Jg Magalhaes, I Tattoli, Dj Philpott, and Lh Travassos. Nod-like proteins in inflammation and disease. *The Journal of Pathology*, 214(2): 136–148, January 2008. ISSN 00223417, 10969896. doi: 10.1002/path.2271.

[7] N. G. Cooper, Roger Eckhardt, and Nancy Shera. *From Cardinals to Chaos: Reflection on the Life and Legacy of Stanislaw Ulam*. CUP Archive, February 1989. ISBN 978-0-521-36734-9.

[8] David B Dahl. Sequentially-Allocated Merge-Split Sampler for Conjugate and Nonconjugate Dirichlet Process Mixture Models. page 27, 2005.

[9] Trevor L Davis. *optparse: Command Line Option Parser*, 2019. URL https://CRAN.R-project.org/package=optparse. R package version 1.6.2.

[10] Matt Dowle and Arun Srinivasan. data.table: Extension of 'data.frame', 2019. URL https://CRAN.R-project.org/package=data.table. R package version 1.12.0.

[11] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[12] Frank Dudbridge. Power and Predictive Accuracy of Polygenic Risk Scores. *PLoS Genetics*, 9(3):e1003348, March 2013. ISSN 1553-7404. doi: 10.1371/journal.pgen.1003348.

[13] Brooke L Fridley and Joanna M Biernacka. Gene set analysis of SNP data: Benefits, challenges, and future directions. *European Journal of Human Genetics*, 19(8):837–843, August 2011. ISSN 1018-4813, 1476-5438. doi: 10.1038/ejhg.2011.57.

[14] Arno Fritsch and Katja Ickstadt. Improved criteria for clustering based on the posterior similarity matrix. *Bayesian Analysis*, 4(2):367–391, June 2009. ISSN 1936-0975. doi: 10.1214/09-BA414.

[15] Wendy S. Garrett, Jeffrey I. Gordon, and Laurie H. Glimcher. Homeostasis and Inflammation in the Intestine. *Cell*, 140(6):859–870, March 2010. ISSN 00928674. doi: 10.1016/j.cell.2010.01.023.

[16] Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–472, November 1992. ISSN 0883-4237. doi: 10.1214/ss/1177011136.

[17] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984. ISSN 0162-8828. doi: 10.1109/TPAMI.1984.4767596.

[18] John Geweke. Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments. *Bayesian Statistics*, 4:169–193.

[19] Elin Grundberg, Kerrin S Small, Åsa K Hedman, Alexandra C Nica, Alfonso Buil, Sarah Keildson, Jordana T Bell, Tsun-Po Yang, Eshwar Meduri, Amy Barrett, James Nisbett, Magdalena Sekowska, Alicja Wilk, So-Youn Shin, Daniel Glass, Mary Travers, Josine L Min, Sue Ring, Karen Ho, Gudmar Thorleifsson, Augustine Kong, Unnur Thorsteindottir, Chrysanthi Ainali, Antigone S Dimas, Neelam Hassanali, Catherine Ingle, David Knowles, Maria Krestyaninova, Christopher E Lowe, Paola Di Meglio, Stephen B Montgomery, Leopold Parts, Simon Potter, Gabriela Surdulescu, Loukia Tsaprouni, Sophia Tsoka, Veronique Bataille, Richard Durbin, Frank O Nestle, Stephen O'Rahilly, Nicole Soranzo, Cecilia M Lindgren, Krina T Zondervan, Kourosh R Ahmadi, Eric E Schadt, Kari Stefansson,

George Davey Smith, Mark I McCarthy, Panos Deloukas, Emmanouil T Dermitzakis, and Tim D Spector. Mapping cis- and trans-regulatory effects across multiple tissues in twins. *Nature Genetics*, 44(10):1084–1089, October 2012. ISSN 1061-4036, 1546-1718. doi: 10.1038/ng.2394.

[20] GTEx Consortium. Genetic effects on gene expression across human tissues. *Nature*, 550(7675):204–213, October 2017. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature24277.

[21] W K Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. page 14.

[22] Boris P. Hejblum, Jason Skinner, and Rodolphe Thiébaut. Time-Course Gene Set Analysis for Longitudinal Gene Expression Data. *PLOS Computational Biology*, 11(6):e1004310, June 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004310.

[23] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[24] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985. ISSN 0176-4268, 1432-1343. doi: 10.1007/BF01908075.

[25] Wenjun Ju, Casey S. Greene, Felix Eichinger, Viji Nair, Jeffrey B. Hodgin, Markus Bitzer, Young-suk Lee, Qian Zhu, Masami Kehata, Min Li, Song Jiang, Maria Pia Rastaldi, Clemens D. Cohen, Olga G. Troyanskaya, and Matthias Kretzler. Defining cell-type specificity at the transcriptional level in human disease. *Genome Research*, 23(11):1862–1873, November 2013. ISSN 1088-9051. doi: 10.1101/gr.155697.113.

[26] Minoru Kanehisa, Yoko Sato, Miho Furumichi, Kanae Morishima, and Mao Tanabe. New approach for understanding genome variations in KEGG. *Nucleic Acids Research*, 47(D1):D590–D595, January 2019. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gky962.

[27] 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, December 2012. ISSN 1460-2059, 1367-4803. doi: 10.1093/bioinformatics/bts595.

[28] Raivo Kolde. Pheatmap: Pretty Heatmaps, 2018. R package version 1.0.10.

[29] Andrea Lancichinetti and Santo Fortunato. Consensus clustering in complex networks. *Scientific Reports*, 2(1):336, December 2012. ISSN 2045-2322. doi: 10.1038/srep00336.

[30] Tao Li and Chris Ding. Weighted Consensus Clustering. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 798–809. Society for Industrial and Applied Mathematics, April 2008. ISBN 978-0-89871-654-2 978-1-61197-278-8. doi: 10.1137/1.9781611972788.72.

[31] John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, Barbara Foster, Mike Moser, Ellen Karasik, Bryan Gillard, Kimberley Ramsey, Susan Sullivan, Jason Bridge, Harold Magazine, John Syron, Johnelle Fleming, Laura Siminoff, Heather Traino, Maghboeba Mosavel, Laura Barker, Scott Jewell, Dan Rohrer, Dan Maxim, Dana Filkins, Philip Harbach, Eddie Cortadillo, Bree Berghuis, Lisa Turner, Eric Hudson, Kristin Feenstra, Leslie Sobin, James Robb, Phillip Branton, Greg Korzeniewski, Charles Shive, David Tabor, Liqun Qi, Kevin Groch, Sreenath Nampally, Steve Buia, Angela Zimmerman, Anna Smith, Robin Burges, Karna Robinson, Kim Valentino, Deborah Bradbury, Mark Cosentino, Norma Diaz-Mayoral, Mary Kennedy, Theresa Engel, Penelope Williams, Kenyon Erickson, Kristin Ardlie, Wendy Winckler, Gad Getz, David DeLuca, Daniel MacArthur, Manolis Kellis, Alexander Thomson, Taylor Young, Ellen Gelfand, Molly Donovan, Yan Meng, George Grant, Deborah Mash, Yvonne Marcus, Margaret Basile, Jun Liu, Jun Zhu, Zhidong Tu, Nancy J Cox, Dan L Nicolae, Eric R Gamazon, Hae Kyung Im, Anuar Konkashbaev, Jonathan Pritchard, Matthew Stevens, Timothèe Flutre, Xiaoquan Wen, Emmanouil T Dermitzakis, Tuuli Lappalainen, Roderic Guigo, Jean Monlong, Michael Sammeth, Daphne Koller, Alexis Battle, Sara Mostafavi, Mark McCarthy, Manual Rivas, Julian Maller, Ivan Rusyn, Andrew Nobel, Fred Wright, Andrey Shabalin, Mike Feolo, Nataliya Sharopova, Anne Sturcke, Justin Paschal, James M Anderson, Elizabeth L Wilder, Leslie K Derr, Eric D Green, Jeffery P Struewing, Gary Temple, Simona Volpi, Joy T Boyer, Elizabeth J Thomson, Mark S Guyer, Cathy Ng, Assya Abdallah, Deborah Colantuoni, Thomas R Insel, Susan E Koester, A Roger Little, Patrick K Bender, Thomas Lehner, Yin Yao, Carolyn C Compton, Jimmie B Vaught, Sherilyn Sawyer, Nicole C Lockhart, Joanne Demchok, and Helen F Moore. The Genotype-Tissue Expression (GTEx) project. *Nature Genetics*, 45(6):580–585, June 2013. ISSN 1061-4036, 1546-1718. doi: 10.1038/ng.2653.

[32] T Maniatis, S Goodbourn, and J. Fischer. Regulation of inducible and

tissue-specific gene expression. *Science*, 236(4806):1237–1245, June 1987. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.3296191.

[33] Samuel A. Mason, Faiz Sayyid, Paul D.W. 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), January 2016. ISSN 1544-6115, 2194-6302. doi: 10.1515/sagmb-2015-0055.

[34] Nicholas Metropolis and S. Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335, September 1949. ISSN 01621459. doi: 10.2307/2280232.

[35] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.1699114.

[36] Stefano Monti. Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. page 28.

[37] Michael A. Mooney and Beth Wilmot. Gene set analysis: A step-by-step guide. *American Journal of Medical Genetics Part B: Neuropsychiatric Genetics*, 168(7):517–527, October 2015. ISSN 15524841. doi: 10.1002/ajmg.b.32328.

[38] Kirill Müller and Hadley Wickham. tibble: Simple data frames, 2019. URL https://CRAN.R-project.org/package=tibble. R package version 2.1.1.

[39] A. C. Nica and E. T. Dermitzakis. Expression quantitative trait loci: Present and future. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 368(1620):20120362–20120362, May 2013. ISSN 0962-8436, 1471-2970. doi: 10.1098/rstb.2012.0362.

[40] Chin-Tong Ong and Victor G. Corces. Enhancer function: New insights into the regulation of tissue-specific gene expression. *Nature Reviews Genetics*, 12(4):283–293, April 2011. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg2957.

[41] OpenStax. *Anatomy & Physiology*. OpenStax CNX, February 2016.

[42] Yered Pita-Juárez, Gabriel Altschuler, Sokratis Kariotis, Wenbin Wei, Katjuša Koler, Claire Green, Rudolph E. Tanzi, and Winston Hide. The Pathway Coexpression Network: Revealing pathway relationships. *PLOS*

Computational Biology, 14(3):e1006042, March 2018. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006042.

[43] Martyn Plummer. *rjags: Bayesian Graphical Models using MCMC*, 2018. URL https://CRAN.R-project.org/package=rjags. R package version 4-8.

[44] William N. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[45] BD Ripley and MD Kirkland. Iterative simulation methods. *Journal of Computational and Applied Mathematics*, 31(1):165–172, 1990.

[46] Brian D Ripley. *Stochastic simulation*, volume 316. John Wiley & Sons, 2009.

[47] Richard S. Savage, Zoubin Ghahramani, Jim E. Griffin, Paul Kirk, and David L. Wild. Identifying cancer subtypes in glioblastoma by combining genomic, transcriptomic and epigenomic data. *arXiv:1304.3577 [q-bio, stat]*, April 2013.

[48] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, October 2005. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0506580102.

[49] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, Lars J Jensen, and Christian von Mering. STRING v11: Protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Research*, 47(D1):D607–D613, January 2019. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gky1131.

[50] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, 2018.

[51] The International IBD Genetics Consortium, Yukihide Momozawa, Julia Dmitrieva, Emilie Théâtre, Valérie Deffontaine, Souad Rahmouni, Benoît Charloteaux, François Crins, Elisa Docampo, Mahmoud Elansary, Ann-Stephan Gori, Christelle Lecut, Rob Mariman, Myriam Mni, Cécile Oury, Ilya Altukhov, Dmitry Alexeev, Yuri Aulchenko, Leila Amininejad, Gerd Bouma, Frank Hoentjen, Mark Löwenberg, Bas Oldenburg, Marieke J.

Pierik, Andrea E. vander Meulen-de Jong, C. Janneke van der Woude, Marijn C. Visschedijk, Mark Lathrop, Jean-Pierre Hugot, Rinse K. Weersma, Martine De Vos, Denis Franchimont, Severine Vermeire, Michiaki Kubo, Edouard Louis, and Michel Georges. IBD risk loci are enriched in multigenic regulatory modules encompassing putative causative genes. *Nature Communications*, 9(1):2427, December 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-04365-8.

[52] Hakon Tjelmeland and Bjorn Kare Hegstad. Mode Jumping Proposals in MCMC. *Scandinavian Journal of Statistics*, 28(1):205–223, March 2001. ISSN 0303-6898, 1467-9469. doi: 10.1111/1467-9469.00232.

[53] Urmo Võsa, Annique Claringbould, Harm-Jan Westra, Marc Jan Bonder, Patrick Deelen, Biao Zeng, Holger Kirsten, Ashis Saha, Roman Kreuzhuber, Silva Kasela, Natalia Pervjakova, Isabel Alvaes, Marie-Julie Fave, Mawusse Agbessi, Mark Christiansen, Rick Jansen, Ilkka Seppälä, Lin Tong, Alexander Teumer, Katharina Schramm, Gibran Hemani, Joost Verlouw, Hanieh Yaghootkar, Reyhan Sönmez, Andrew A. Andrew, Viktorija Kukushkina, Anette Kalnapenkis, Sina Rüeger, Eleonora Porcu, Jaanika Kronberg-Guzman, Johannes Kettunen, Joseph Powell, Bernett Lee, Futao Zhang, Wibowo Arindrarto, Frank Beutner, BIOS Consortium, Harm Brugge, i2QTL Consortium, Julia Dmitrieva, Mahmoud Elansary, Benjamin P. Fairfax, Michel Georges, Bastiaan T Heijmans, Mika Kähönen, Yungil Kim, Julian C. Knight, Peter Kovacs, Knut Krohn, Shuang Li, Markus Loeffler, Urko M Marigorta, Hailang Mei, Yukihide Momozawa, Martina Müller-Nurasyid, Matthias Nauck, Michel Nivard, Brenda Penninx, Jonathan Pritchard, Olli Raitakari, Olaf Rotzschke, Eline P. Slagboom, Coen D.A. Stehouwer, Michael Stumvoll, Patrick Sullivan, Peter A.C. 't Hoen, Joachim Thiery, Anke Tönjes, Jenny van Dongen, Maarten van Iterson, Jan Veldink, Uwe Völker, Cisca Wijmenga, Morris Swertz, Anand Andiappan, Grant W. Montgomery, Samuli Ripatti, Markus Perola, Zoltan Kutalik, Emmanouil Dermitzakis, Sven Bergmann, Timothy Frayling, Joyce van Meurs, Holger Prokisch, Habibul Ahsan, Brandon Pierce, Terho Lehtimäki, Dorret Boomsma, Bruce M. Psaty, Sina A. Gharib, Philip Awadalla, Lili Milani, Willem H. Ouwehand, Kate Downes, Oliver Stegle, Alexis Battle, Jian Yang, Peter M. Visscher, Markus Scholz, Gregory Gibson, Tõnu Esko, and Lude Franke. Unraveling the polygenic architecture of complex traits using blood eQTL meta-analysis. Preprint, Genomics, October 2018.

[54] Matthew T. Weirauch. Gene Coexpression Networks for the Analysis of DNA Microarray Data. In Matthias Dehmer, Frank Emmert-Streib, Armin

Graber, and Armindo Salvador, editors, *Applied Statistics for Network Biology*, pages 215–250. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, April 2011. ISBN 978-3-527-63807-9 978-3-527-32750-8. doi: 10.1002/9783527638079.ch11.

[55] Hadley Wickham. ggplot2: Elegant graphics for data analysis, 2016. URL https://ggplot2.tidyverse.org.

[56] Naomi R. Wray, Sang Hong Lee, Divya Mehta, Anna A.E. Vinkhuyzen, Frank Dudbridge, and Christel M. Middeldorp. Research Review: Polygenic methods and their application to psychiatric traits. *Journal of Child Psychology and Psychiatry*, 55(10):1068–1087, October 2014. ISSN 00219630. doi: 10.1111/jcpp.12295.

[57] James Homer Wright. The histogenesis of the blood platelets. *Journal of Morphology*, 21(2):263–278, July 1910. ISSN 0362-2525, 1097-4687. doi: 10.1002/jmor.1050210204.

A Additional theory

A.1 Rand index

A popular metric for comparing the similarity of two clusterings of the data is the *Rand index* [44]. If one assumes that all points are of equal importance in determining clusterings, then in combination with the discrete nature of clusters and the fact that a cluster is defined as much by what it does not contain as that which it does, Rand [44] proposes a metric to measure similarity between clusterings. Between clusterings Y and Y' for any two points x_i and x_j there can exist one of a number of scenarios regarding their labelling. Let γ_{ij} be a measure between the two points x_i and x_j. For the two points, they can have:

1. the same label in both clusterings ($c_i = c_j \wedge c'_i = c'_j$) ($\gamma_{ij} = 1$);
2. different labels in both ($c_i \neq c_j \wedge c'_i \neq c'_j$) ($\gamma_{ij} = 1$); or
3. the same label in one but not in the other ($c_i \neq c_j \wedge c'_i = c'_j \vee c_i = c_j \wedge c'_i \neq c'_j$) ($\gamma_{ij} = 0$).

Thus Rand [44] proposed counting the number of times any two points have one of 1 or 2 from list A.1 and finding the proportion of these compared to the number of all possible point combinations. More formally, this is:

$$A\binom{n}{2}^{-1} = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \gamma_{ij} \tag{20}$$

This can be envisioned as a $K \times K'$ contingency table of the count of overlapping points, as shown in table 3. Table 3 uses the following notation:

- n_{ij} is the number of points that have membership in Y_i in clustering Y and Y'_j in clustering Y';
- $n_{\cdot j}$ is the number of points in cluster Y'_j in clustering Y';
- $n_{i\cdot}$ is the number of points in cluster Y_i in clustering Y; and
- $n_{\cdot\cdot} = n$ is the number of points in clusterings Y and Y'.

One can restate equation 20 in terms of the notation from table 3 [5]:

$$A = \binom{n}{2} + \sum_{i=1}^{K} \sum_{j=1}^{K'} n_{ij}^2 - \frac{1}{2}\left(\sum_{i=1}^{K} n_{i\cdot}^2 + \sum_{j=1}^{K'} n_{\cdot j}^2\right) \tag{21}$$

$$= \binom{n}{2} + 2\sum_{i=1}^{K} \sum_{j=1}^{K'} \binom{n_{ij}}{2} - \left[\sum_{i=1}^{K} \binom{n_{i\cdot}}{2} + \sum_{j=1}^{K'} \binom{n_{\cdot j}}{2}\right] \tag{22}$$

$Y \diagdown Y'$	Y'_1	Y'_2	$\cdots$	$Y'_{K'}$	Sums
Y_1	n_{11}	n_{12}	$\cdots$	$n_{1K'}$	$n_{1\cdot}$
Y_2	n_{21}	n_{22}	$\cdots$	$n_{2K'}$	$n_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Y_K	n_{K1}	n_{K2}	$\cdots$	$n_{KK'}$	$n_{K\cdot}$
Sums	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot K'}$	$n_{\cdot\cdot} = n$

Table 3: Contingency table used by Rand [44] to calculate a measure of similarity between clusterings Y and Y'.

Hubert and Arabie [24] extend the Rand index to account for chance. They include a null hypothesis and assume that there is a probability of some points having a γ value of 1 by chance. Consider the scenario where a point x_i has the same label as another point x_j under clustering Y. For another clustering Y', there a non-zero is a probability $c'_i = c'_j$ purely by chance and does not represent a similarity between Y and Y'. If one generates two clusterings Y and Y' by sampling from the integers in the closed interval $[1, K]$ (i.e. by sampling from discrete uniform distribution $\mathcal{U}\{1, K\}$), then the contingency table generated is constructed from the generalised hyper-geometric distribution [24]. It can be shown that the expected number of points with common membership in both clusters is non-zero. Specifically:

$$\mathbb{E}\left[\sum_{i=1}^{K}\sum_{j=1}^{K}\binom{n_{ij}}{2}\right] = \frac{\sum_{i=1}^{K}\binom{n_{i\cdot}}{2}\sum_{j=1}^{K}\binom{n_{\cdot j}}{2}}{\binom{n}{2}} \tag{23}$$

This is the product of the number of distinct pairs that can be formed from rows and the number of distinct pairs that can be constructed from columns, divided by the total number of pairs.

For a particular cell of the contingency table, the expected number of entries of the type described in point 1, is the product of number of pairs in its row and in its column divided by the total number of possible pairs:

$$\mathbb{E}\left[\binom{n_{ij}}{2}\right] = \frac{\binom{n_{i\cdot}}{2}\binom{n_{\cdot j}}{2}}{\binom{n}{2}} \tag{24}$$

One can see that as each component of equation 21 is some transformation of $\sum_{i,j}\binom{n_{ij}}{2}$, one can directly state the expected value of the Rand index by combining equations 21 and 24:

$$\mathbb{E}\left[A\binom{n}{2}^{-1}\right] = 1 + 2\sum_{i=1}^{K}\binom{n_{i\cdot}}{2}\sum_{j=1}^{K'}\binom{n_{\cdot j}}{2}\binom{n}{2}^{-2} - \left[\sum_{i=1}^{K}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{K'}\binom{n_{\cdot j}}{2}\right]\binom{n}{2}^{-1} \tag{25}$$

Defining an index corrected for chance as:

$$\text{Corrected index} = \frac{\text{Index} - \text{Expected index}}{\text{Maximum index} - \text{Expected index}} \tag{26}$$

Assuming a maximum value of 1 for the Rand index then gives a corrected Rand index:

$$\frac{\sum_{i=1}^{K}\sum_{j=1}^{K'}\binom{n_{ij}}{2} - \sum_{i=1}^{K}\binom{n_{i\cdot}}{2}\sum_{j=1}^{K'}\binom{n_{\cdot j}}{2}\binom{n}{2}^{-1}}{\frac{1}{2}\left[\sum_{i=1}^{K}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{K'}\binom{n_{\cdot j}}{2}\right] - \sum_{i=1}^{K}\binom{n_{i\cdot}}{2}\sum_{j=1}^{K'}\binom{n_{\cdot j}}{2}\binom{n}{2}^{-1}} \tag{27}$$

This quantity described in equation 27 is defined as the *adjusted Rand index* and I use it as my measure of choice for similarity between clusterings.

I describe an explicit example motivating the adjusted Rand index in section A.1.1.

A.1.1 Motivating example: adjusted Rand index

Consider the case of n labels Y and Y' generated from $\mathcal{U}\{1,3\}$ where n is some arbitrarily large number and $\mathcal{U}\{x,y\}$ is the uniform distribution over the interval $[x,y]$. Then as n tends to infinity we can expect that our contingency table has entries of $\frac{n}{9}$ in each cell. If one calculates the Rand index on these random partitions where any similarity is purely by chance one finds, it comes to (approximately) 0.56. This suggests there is some similarity between Y and Y', but this is misleading as we know any similarity is stochastic. In the same scenario the adjusted Rand index between the partitions is 0. This seems preferable. Based on this, one could argue that the Rand index has inflated values. Consider the case that we have n points in total, but we let the first $\frac{7n}{16}$ have a common label (say $(c_1,\ldots,c_{n_1}) = 1$ for $n_1 = \frac{7n}{16}$) and then draw the remaining $\frac{9n}{16}$ points from $\mathcal{U}\{1,3\}$. Then, as n tends to infinity, our contingency table tends to that described in table 4. One feels that the high Rand index for such a clustering, 0.64, is misleading in its magnitude. In such a scenario we feel one has to consider this 0.64 in the context of the 0.56 for a purely random similarity - this is difficult to do without explicitly checking what the Rand index is for a random partitioning for a given K and K'. Thus the use of the full unit interval in comparing similarity by a corrected index such as the adjusted Rand index requires less vigilance on the part of the analyst. In the second scenario, the adjusted Rand index is 0.28.

$Y \diagdown Y'$	Y'_1	Y'_2	Y'_3	Sums
Y_1	$\frac{n}{2}$	$\frac{n}{16}$	$\frac{n}{16}$	$\frac{10n}{16}$
Y_2	$\frac{n}{16}$	$\frac{n}{16}$	$\frac{n}{16}$	$\frac{3n}{16}$
Y_3	$\frac{n}{16}$	$\frac{n}{16}$	$\frac{n}{16}$	$\frac{3n}{16}$
Sums	$\frac{10n}{16}$	$\frac{3n}{16}$	$\frac{3n}{16}$	$\frac{16n}{16} = n$

Table 4: Contingency table for the non-random clustering described in section A.1.1.

A.2 Standardisation

For a p-vector of observations, $X_i = (x_{i1}, \ldots, x_{1p})$, standardisation of X_i be defined as the mapping from X_i to $X'_i = (x'_{i1}, \ldots, x'_{ip})$ defined by the *sample mean*, $\bar{x}_i$, and sample standard deviation, s_i:

$$\bar{x}_i = \frac{1}{p}\sum_{j=1}^{p} x_{ij} \tag{28}$$

$$s_i^2 = \frac{1}{p-1}\sum_{j=1}^{p}\left(x_{ij} - \bar{x}_i\right)^2 \tag{29}$$

$$x'_{ij} = \frac{x_{ij} - \bar{x}_i}{s_i} \;\forall\; j \in (1, \ldots, p) \tag{30}$$

We refer to X'_i as the standardised form of X. If we are given a dataset $X = (X_1, \ldots, X_n)$ where each X_i is a p-vector of observations of the form referred to above, then in referring to the standardised form of X, we mean the dataset $X' = (X'_1, \ldots, X'_n)$ where each X'_i is the standardised form of X_i.

Standardisation moves the values observed for each X_i to a common scale where each vector has an observed mean and standard deviation of 0 and 1 respectively.

A.2.1 Motivating example: Standardising gene expression data

If one considers table 5 which contains an example of expression data for some genes A, B, C, D and E across people 1 to 4. One can see that genes A and C have similar patters in variation across the people, as do genes B and D. Gene E is not consistent with any other gene here. However, as this relative variation is of interest rather than the magnitude of expression, one can see that standardising the data is required.

If one were to cluster the data as represented in table 5, one would place genes A and B in one cluster and genes C, D and E in another as their absolute

Genes	Person 1	Person 2	Person 3	Person 4
A	5.1	5.2	4.9	5.0
B	5.1	4.9	5.2	5.4
C	1.4	1.5	1.2	1.3
D	1.4	1.2	1.5	1.7
E	1.4	1.5	1.4	1.5

Table 5: Example gene expression data.

Genes	Person 1	Person 2	Person 3	Person 4
A	0.39	1.16	-1.16	-0.39
B	-0.24	-1.20	0.24	1.20
C	0.39	1.16	-1.16	-0.39
D	-0.24	-1.20	0.24	1.20
E	-0.87	0.87	-0.87	0.87

Table 6: Example standardised gene expression data.

expression levels are similar (as can be seen in figure 43). However, if the expression level of each gene is standardised as per section A.2, the data is then as represented in table 6. The data are now on the same scale and thus the characteristic that will determine a clustering is the variation of expression across people. As we want genes with similar patterns of variation (i.e. that are co-expressed) this enables us to cluster under our objective of defining gene sets. In this case genes A and C are one cluster, genes B and D another with gene E in a cluster alone, as can be seen in figure 44. As this is the type of data we wish to cluster across, we therefore most standardise our expression data before clustering can be implemented.

A.3 Gibbs sampling

Gibbs sampling is based on the concept of *Monte Carlo integration* and *Markov chains*. It is a special case of the *Metropolis-Hastings algorithm*. Gibbs sampling is used to sample directly from the posterior distribution of the model's random variables. We briefly describe the emphasized terms below before describing Gibbs sampling in more detail.

A.3.1 Monte Carlo integration

The original Monte Carlo method was developed by Stanislaw Ulam, a Polish mathematician while he worked at Los Alamos on the Manhattan Project.

Along with Nicholas Metropolis [7], he developed the method as a means to solving integrals by use of random number generation [34].

Suppose there is some complex integral we wish to solve on some interval, (a, b):

$$\int_a^b h(x)dx \tag{31}$$

If we can decompose $h(x)$ into the product of some more simple function $f(x)$ and a probability density $p(x)$ where both are defined over (a, b) we can then state:

$$\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = \mathbb{E}_{p(x)}\left[f(x)\right] \tag{32}$$

We assume that we can approximate this expectation of $f(x)$ over $p(x)$ by drawing N random variables $x = (x_1, \ldots, x_N)$ from $p(x)$ (by the Law of Large Numbers); thus (32) becomes:

$$\int_a^b h(x)dx = \mathbb{E}_{p(x)}\left[f(x)\right] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{33}$$

This is *Monte Carlo integration*.

A.3.2 Markov chains

Consider a random variable X observed at discrete times $t = (t_0, \ldots, t_n)$, with the observation at t_i denoted $X(t_i)$. Let S be the state space of possible values X can take. X is said to have the *Markov property* if the transition probabilities between states depend only on the current state, i.e. for any states $s_i, s_j, s_k \in S$:

$$\mathbb{P}(X(t_{n+1}) = s_j | X(t_0) = s_k, \ldots, X(t_n) = s_i) = \mathbb{P}(X(t_{n+1}) = s_j | X(t_n) = s_i) \tag{34}$$

Thus prediction depends only on the information in the present; the process is said to be memory-less as the past does not affect future outcomes. In this case we refer to X as a *Markov process*. A *Markov chain* refers to a sequence of random variables $(X_0, \ldots, X_n)$ generated by a Markov process.

Let $\mathbf{P}$ be the matrix of $|S| \times |S|$ where entry (i, j) is the transition probability from state s_i to s_j. We define the n-step transition probability p_{ij}^n as the probability that the process is in state s_j given it was in state s_i at a remove of n steps, i.e.:

$$p_{ij}^n = \mathbb{P}(X(t_{i+n} = s_j | X(t_i) = s_i) \tag{35}$$

A Markov chain is said to be *irreducible* if $p_{ij}^n > 0 \ \forall \ i, j \in \mathbb{N}$. This means that there always exists a possible path from any state s_i to every other state s_j. If this is true the states are said to communicate. If the number of steps between two states is not required to be the multiple of some integer than the chain is considered aperiodic.

For some state $s \in S$ denote $\mathbb{P}(X(t_{n+1}) = s)$ by π_s. The chain has the *reversible* property if for any states $x, y \in S$ the detailed balance (36) holds:

$$\mathbb{P}(X(t_{n+1}) = x | X(t_n) = y)\pi_x = \mathbb{P}(X(t_{n+1}) = y | X(t_n) = x)\pi_y \tag{36}$$

This is sufficient condition for a unique, stationary distribution. That is the probability of being in any given state for the process is independent of the starting condition given sufficient time and the transition probabilities have approached some limiting value.

More formally, a stationary distribution, π, is a row-vector with $|S|$ entries defined by:

$$\pi_j \geq 0 \, \forall \, j \in (1, \ldots, |S|) \tag{37}$$

$$\sum_{j=1}^{|S|} \pi_j = 1 \tag{38}$$

That is invariant under the operation of the transition matrix $\mathbf{P}$ upon it. That is:

$$\pi = \pi \mathbf{P} \tag{39}$$

Thus the distribution, π, remains unchanged in the Markov chain as time progresses.

If for any initial state π_0, the chain has the property:

$$\lim_{n \to \infty} \pi_0 \mathbf{P}^n = \pi \tag{40}$$

For some transition matrix $\mathbf{P}$ and stationary distribution π, then the chain is said to converge to π. Once the chain is sampling from π, then as this distribution is stationary, the samples drawn should be independent of the previous draws as can be seen in (39). For this reason auto-correlation is one of the tests of convergence within a Markov chain.

A.3.3 Markov-Chain Monte Carlo methods

Markov-Chain Monte Carlo (MCMC) methods developed as a method to obtain samples from some complex distribution $p(x)$ for the decomposition suggested in (32). Our goal in the following is to draw samples from some distribution $p(\theta)$ where we have some distribution $f(\theta)$ such that:

$$p(\theta) = \frac{f(\theta)}{K} \tag{41}$$

For some constant K where K may not be known and is often difficult to compute.

The Metropolis-Hastings algorithm [21] is a popular MCMC algorithm. It is an extension to the original Metropolis algorithm which allows for asymmetry in state probabilities (i.e. that $p_{ij} \neq p_{ji}$). The Metropolis algorithm [34] [35] generates a sequence of draws from $p(\theta)$ using the following steps:

1. Initialise with some arbitrary value θ_0 with the condition that $f(\theta_0) > 0$ and also choose some probability density $q(\theta_1|\theta_2)$ as the *jumping distribution* or *proposal density*. For the Metropolis algorithm one demands that this is symmetric (i.e. $q(\theta_1|\theta_2) = q(\theta_2|\theta_1)$).
2. For each iteration, t:
   (a) Using the current value θ_{t-1}, sample a candidate point, θ^*, from $q(\theta^*|\theta_{t-1})$.
   (b) Calculate the *acceptance ratio* for the new value θ^*:

$$\alpha = \frac{p(\theta^*)}{p(\theta_{t-1})} = \frac{f(\theta^*)}{f(\theta_{t-1})} \tag{42}$$

   Note that as the proportionality constant is the same for all θ that this is an equivalence rather than proportional relationship.
   (c) Accept the new value θ^* with probability equal to $\min(\alpha, 1)$. Generate a number u from the uniform distribution on $[0, 1]$ and accept if $\alpha \geq u$, else reject.

This generates a Markov chain $(\theta_0, \ldots, \theta_k, \ldots, \theta_n)$ as each iteration is conditionally independent of all others given the sample from the iteration preceding it. A stationary distribution is reached after a *burn-in* period of k steps (for some $k \in \mathbb{N}$) and all following samples come from $p(\theta)$ (i.e. the vector $(\theta_{k+1}, \ldots, \theta_n)$ are samples from $p(\theta)$). Knowing k is a non-trivial issue; we often assume some arbitrary large number of burn-in iterations erring on the side of caution. The samples generated are highly correlated with other samples from within a close

range of iterations. To avoid recording this duplicate information, often only every lth sample is recorded (called *thinning*) for some small l.

Hastings [21] extends this method to allow asymmetric proposal densities, in which case the acceptance ratio changes to:

$$\alpha = \min\left(\frac{f(\theta^*)q(\theta^*|\theta_{t-1})}{f(\theta_{t-1})q(\theta_{t-1}|\theta^*)}, 1\right) \tag{43}$$

Geman and Geman [17] use a special case of the Metropolis-Hastings algorithm, taking $\alpha = 1 \ \forall \ \theta^*$, accepting all proposed values. This is known as a *Gibbs sampler*.

These methods are useful in a Bayesian context as one is interested in a rather complex distribution, the posterior, and know two simpler quantities, the prior and the likelihood, that the posterior is proportional to (as shown in (1)). Thus one can use MCMC methods to sample directly from the posterior distribution without directly calculating the normalising constant.

A.4 Convergence

Assessment of convergence of Markov chains is a non-trivial issue. Apparent convergence can occur quite quickly when using a Gibbs sampler. Ripley and Kirkland [45] refer to a case when convergence was apparently achieved within several minutes, but then the algorithm jumped after a week of runtime. The following sub-sections briefly describe several heuristic measures to inspect convergence, but none are considered an absolute guarantee.

A.4.1 Effective sample size

The Effective sample size (ESS) [46] is an attempt to convert the number of MCMC samples, which can be highly autocorrelated, into an equivalent number of independent samples; consider it as an exchange rate. For n samples and the letting $\rho(i)$ denote the correlation between the n^{th} sample and the sample drawn at a lag of i:

$$ESS = \frac{n}{1 + 2\sum_{i=1}^{\infty}\rho(i)} \tag{44}$$

If the level of autocorrelation present is 0, then $ESS = n$; if the autocorrelation decreases sufficiently slowly with regards to the lag that the sum in the denominator diverges, then $ESS = 0$. One can use this to estimate the burn-in required to use samples only from the stationary distribution. One should use the burn-in at which ESS is maximised across all variables, as one wants to maximise

the amount of samples available while also ensuring one is sampling from the correct distribution.

A.4.2 Auto-correlation

The autocorrelation, $\rho(i)$, is the correlation between the current sample and the sample at a lag of i. $\rho(0) = 1$ as the observation correlates perfectly with itself. If the samples drawn are no longer correlated with previous samples this indicates the chain has achieved stationarity based upon (39).

A.4.3 Geweke Z-score

Geweke plot [18] for a long chain with random seed 3 in case 1 of the simulations. If the chain has reached stationarity the Z-scores should be described by a standard normal distribution (in which case 95% of recorded values should be contained within the dashed lines). This is the case here, indicating, particularly in conjunction with figure 5, that the chain has converged.

Geweke [18] proposed a statistic based upon time-series methods to test the convergence. If the object of interest is a mean (which applies in the case of the continuous parameters of the MDI model) then this method is appropriate. The concept is based upon the belief that if the samples are all from the same distribution (i.e. the whole chain has converged), then the mean of the early samples should be similar to that from the later samples. To do this the samples are considered within two sets; the first 10% and the last 50% of the recorded samples. The convergence statistic, Z is the difference of the means associated with the samples from each set divided by the asymptotic standard error of their difference. As the number of samples recorded goes to infinity, the sampling distribution of Z goes to $\mathcal{N}(0,1)$ if the chain has converged. Therefore the Geweke plot is inspected for to see if many of the Z-scores occupy the extreme tails of the $\mathcal{N}(0,1)$ distribution.

A.4.4 Gelman-Rubin shrink factor

The approach described by Gelman and Rubin [16] compares sequences across chains. If these are all sampling from the same distribution this is taken as evidence for convergence. This method tries to check if the samples from different chains are more different than would be expected based upon their internal variability; it does this by an analysis of the variance. Convergence is diagnosed when the between-sequence variance B is no larger than the within-sequence variance W.

Consider a m parallel chains, each of length n of the variable X. Denote the chain specific samples by $x_i = (x_i 1, \ldots, x_i n)$. Then:

$$B = \frac{n}{m-1} \sum_{i=1}^{m} (\bar{x}_i - \bar{x})^2 \tag{45}$$

$$W = \frac{1}{m} \sum_{i=1}^{m} s_i^2 \tag{46}$$

where $\bar{x}_i$ is the sample mean and s_i is the sample standard deviation of the i^{th} chain and $\bar{x}$ is the sample mean across chains (with these values calculated as per (28)).

From these variance components, it is expected that W should be an estimate of $Var(X)$; a second estimate is constructed:

$$\hat{Var}(X) = \frac{n-1}{n} W + \frac{1}{n} B \tag{47}$$

If the chains are stationary, this quantity $\hat{Var}(X)$ is an unbiased estimate of $Var(X)$, but if the starting points are over-dispersed, then it is an over-estimate.

As W should under-estimate $Var(X)$ for any finite value of n (as the individual chains should not explore the full target distribution and thus will under-estimate the variance present), there are two quantities estimating $Var(X)$ that should also impose the upper and lower bounds on $Var(X)$. Thus Gelman and Rubin [16] proposed using the ratio of these quantities to estimate across-chain convergence. Specifically the estimated potential scale reduction or *shrink factor*:

$$\sqrt{\hat{R}} = \sqrt{\frac{\hat{Var}}{W}} \tag{48}$$

As the simulation converges, $\sqrt{\hat{R}}$ falls to 1. This means that the parallel Markov chains are essentially overlapping. If the shrink factor is high, then one should proceed with further simulations.

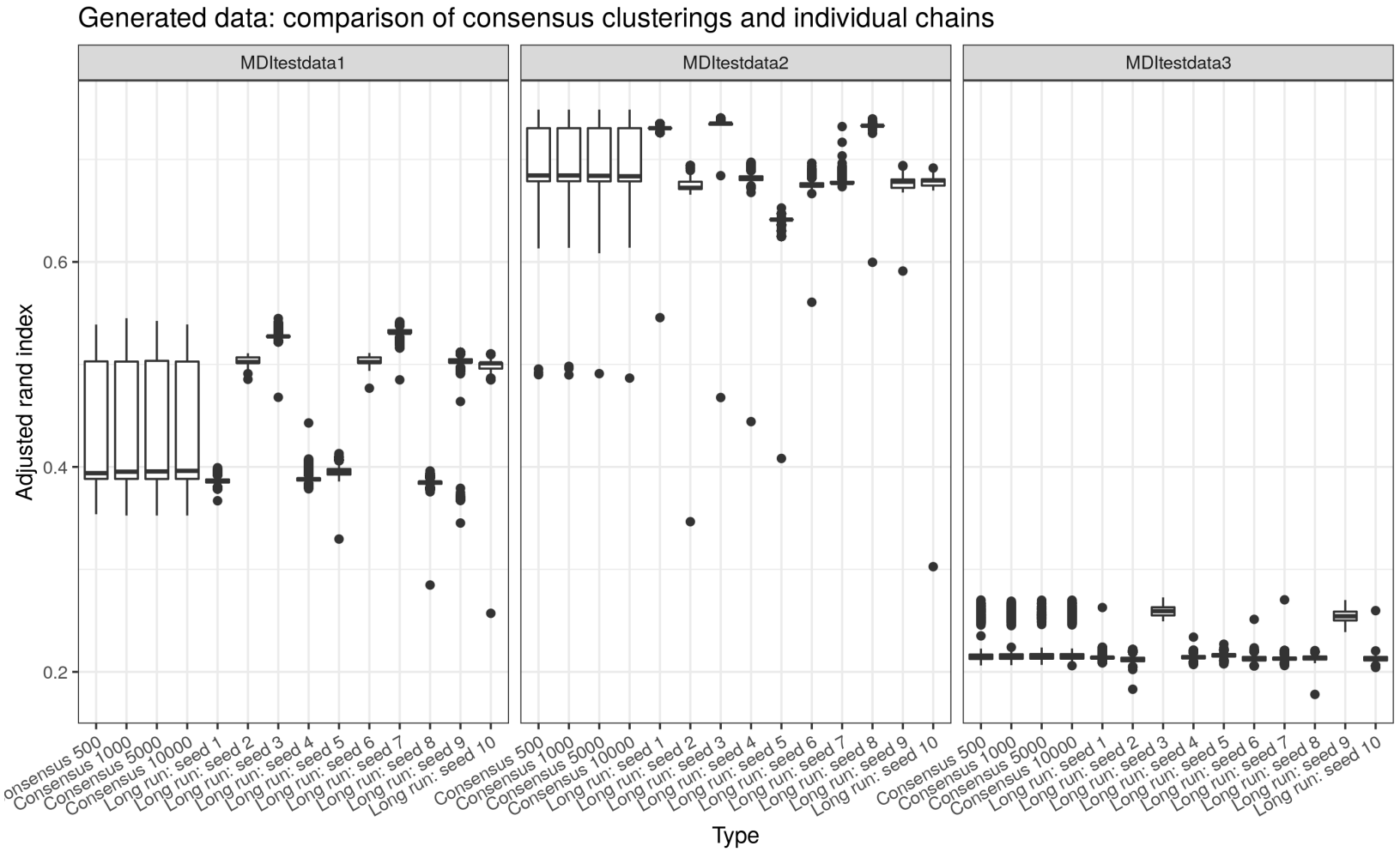

Figure 15: Box plots for distribution of adjusted rand index between the clustering at each iteration to the true clustering for different lengths of consensus clustering and different initialisation of long chains.

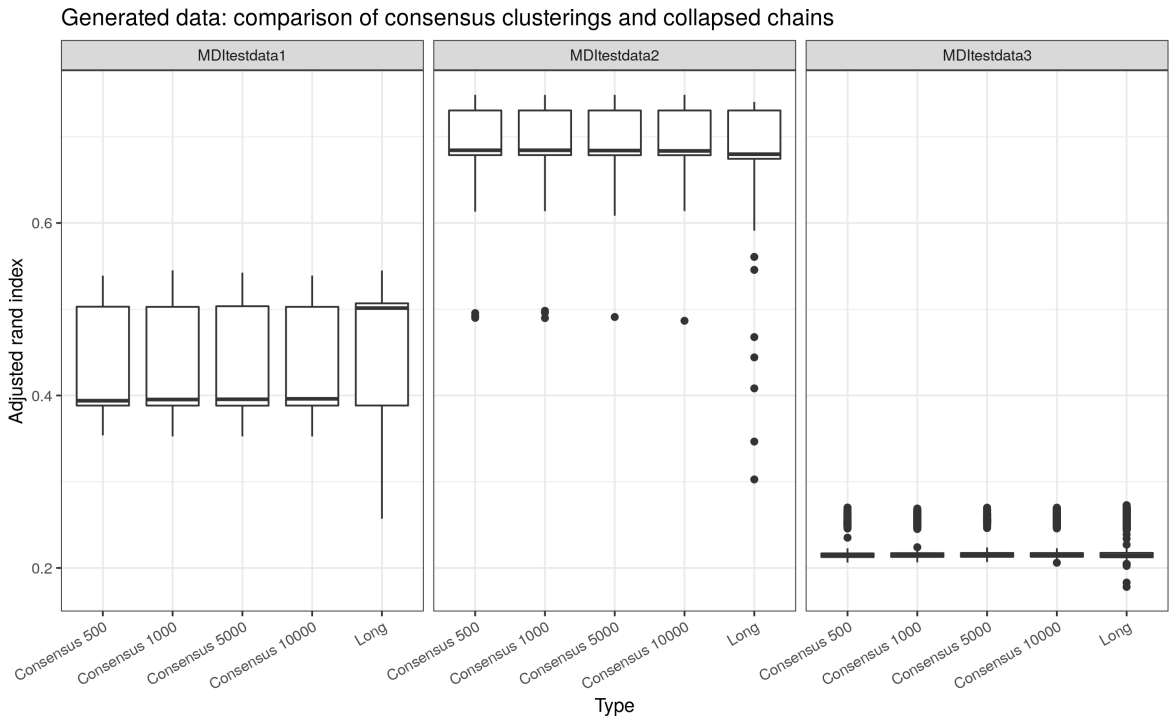

Figure 16: Box plots for distribution of adjusted rand index between the clustering at each iteration to the true clustering for different lengths of consensus clustering and the collapsed long chains.

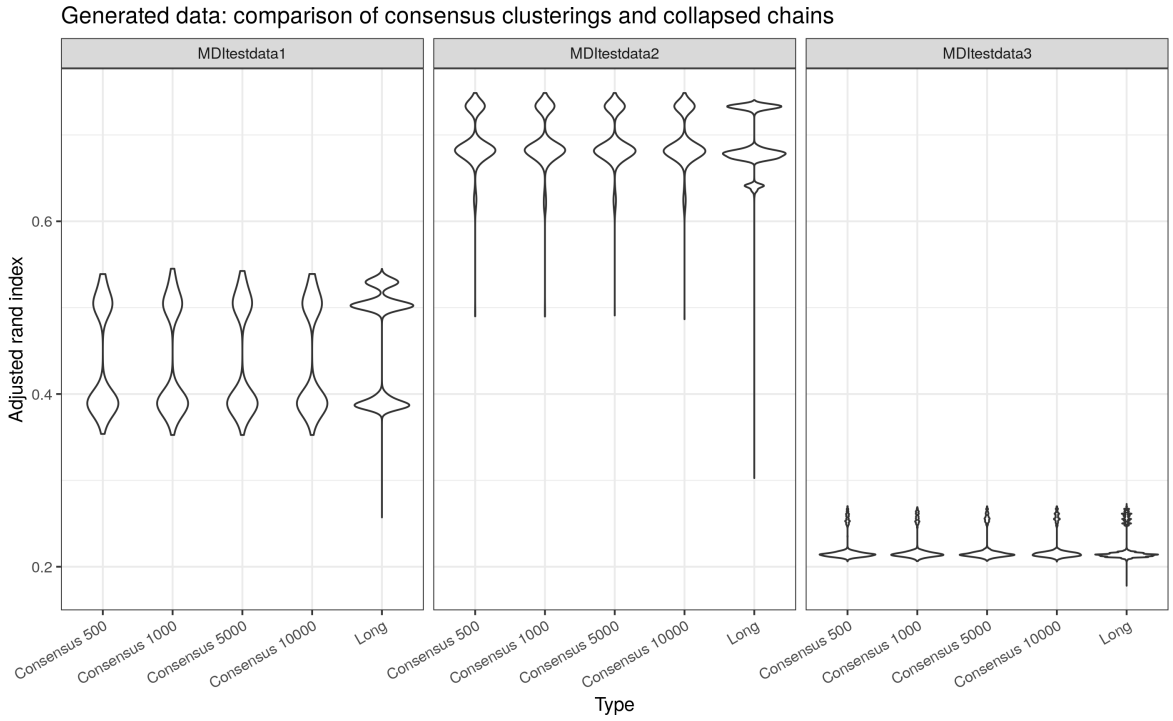

Figure 17: Violin plots for distribution of adjusted rand index between the clustering at each iteration to the true clustering for different lengths of consensus clustering and the collapsed long chains. We can see that the consensus clustering approximates the modes described across chains quite well.

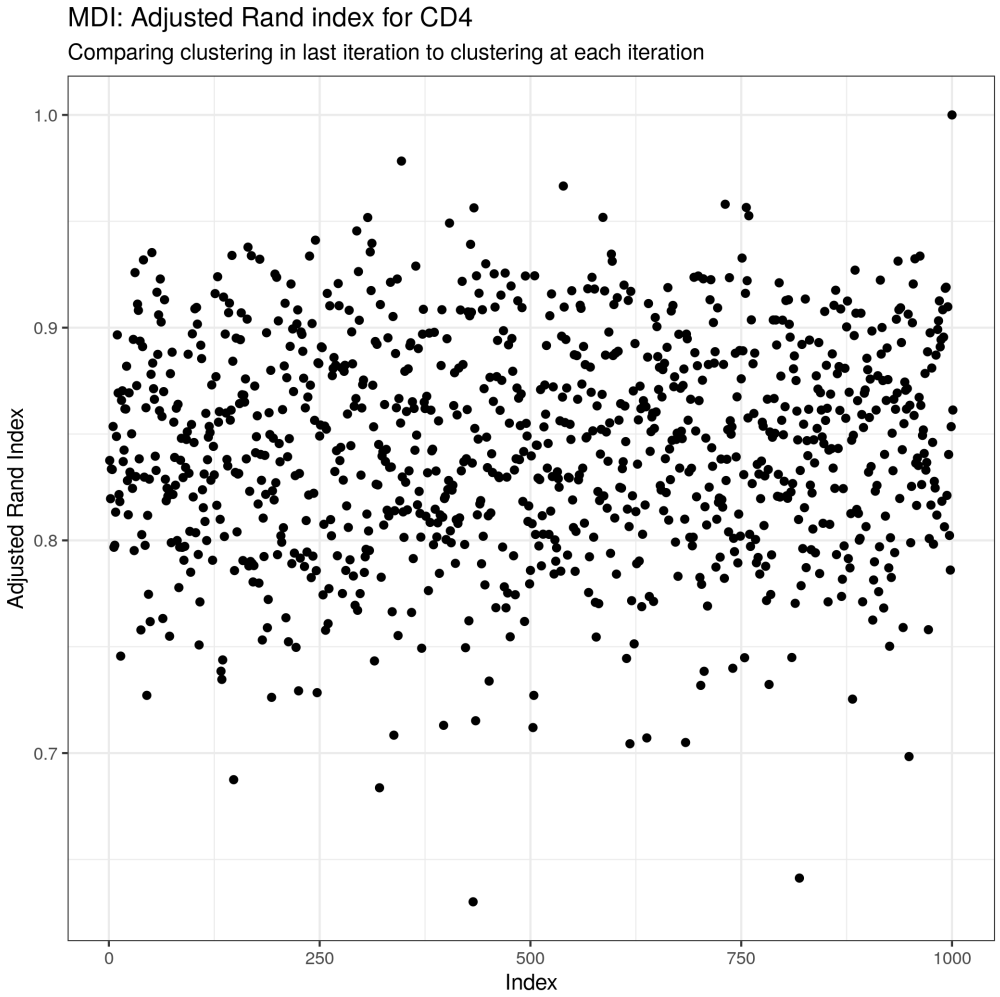

Figure 18: Plot of the adjusted Rand index between the clustering in each seed to that in the 1,000th for CD14.

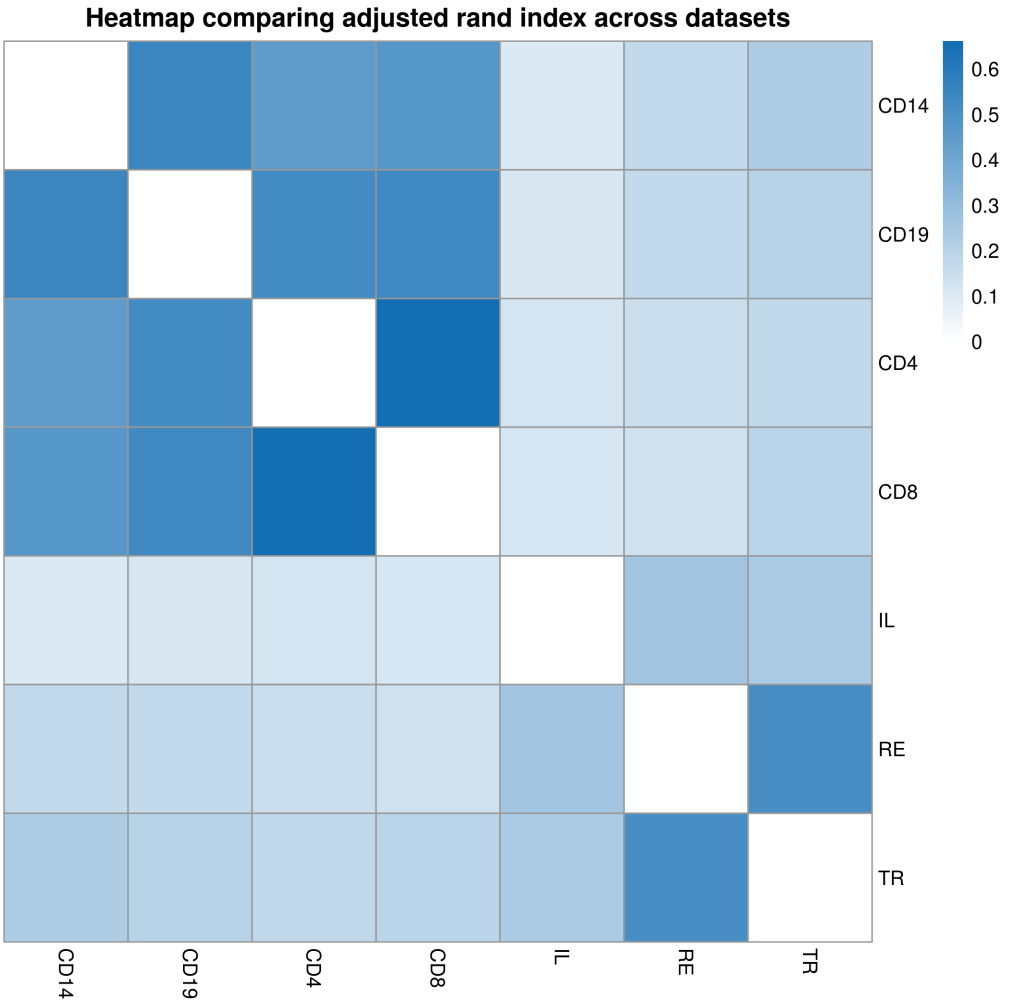

Figure 19: Heatmap of the mean adjusted Rand index comparing the clustering across datasets for each seed.

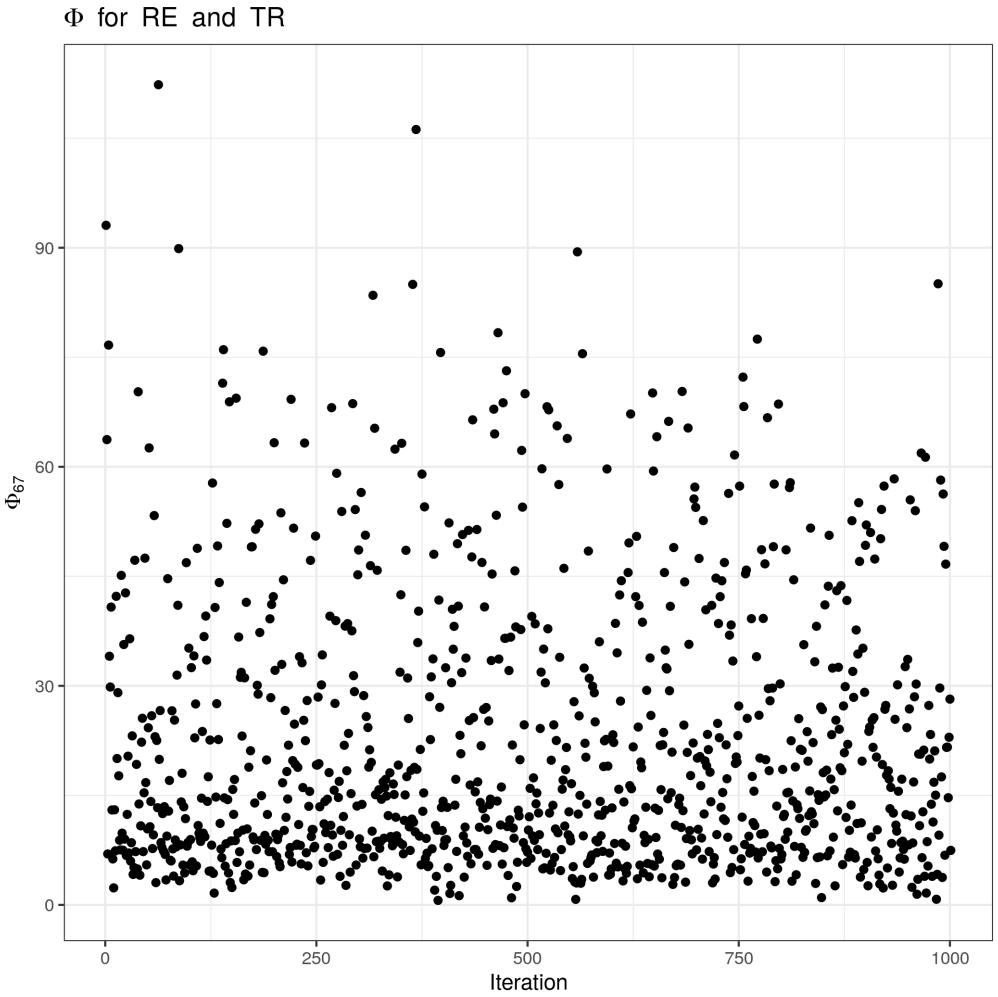

Figure 20: Plot of the ϕ_{67} values across all seeds, between the RE and TR datasets (note that the high values indicate a high clustering correlation).

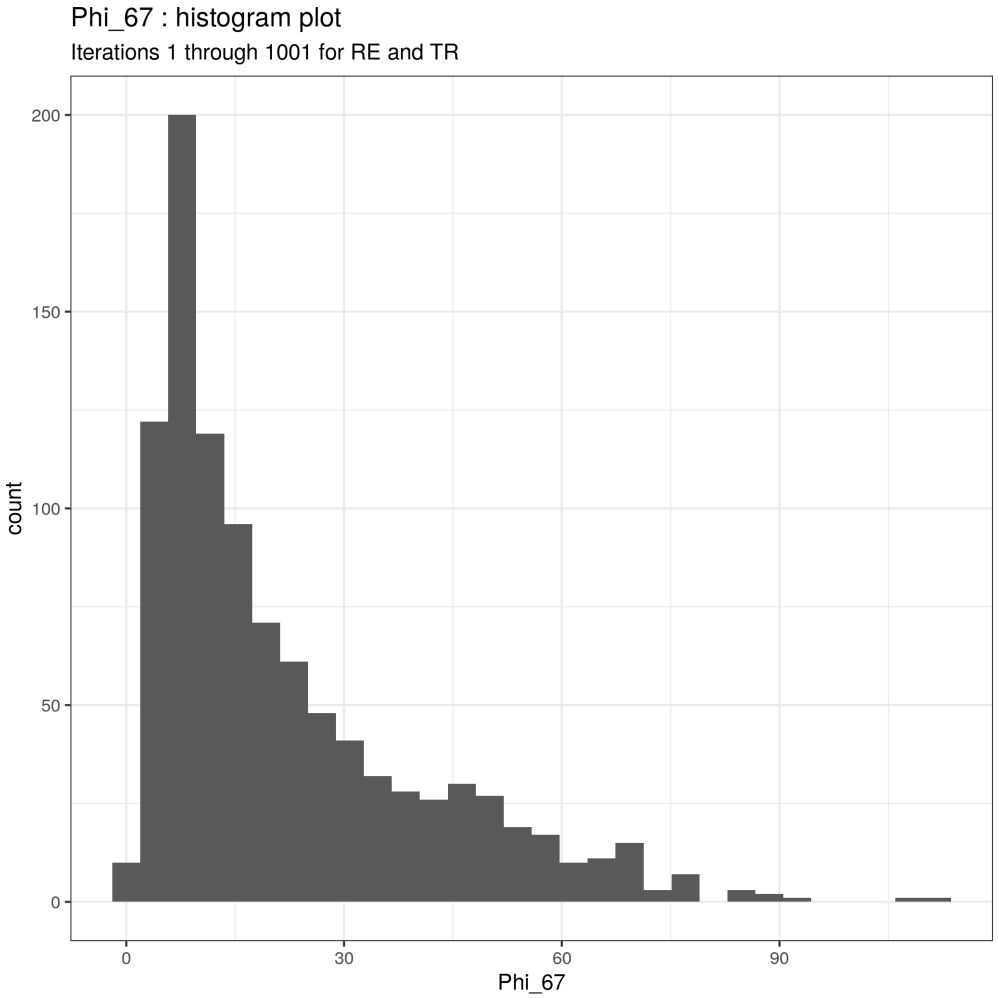

Figure 21: Histogram of the distribution of ϕ_{67} values across seeds(between the RE and TR datasets).

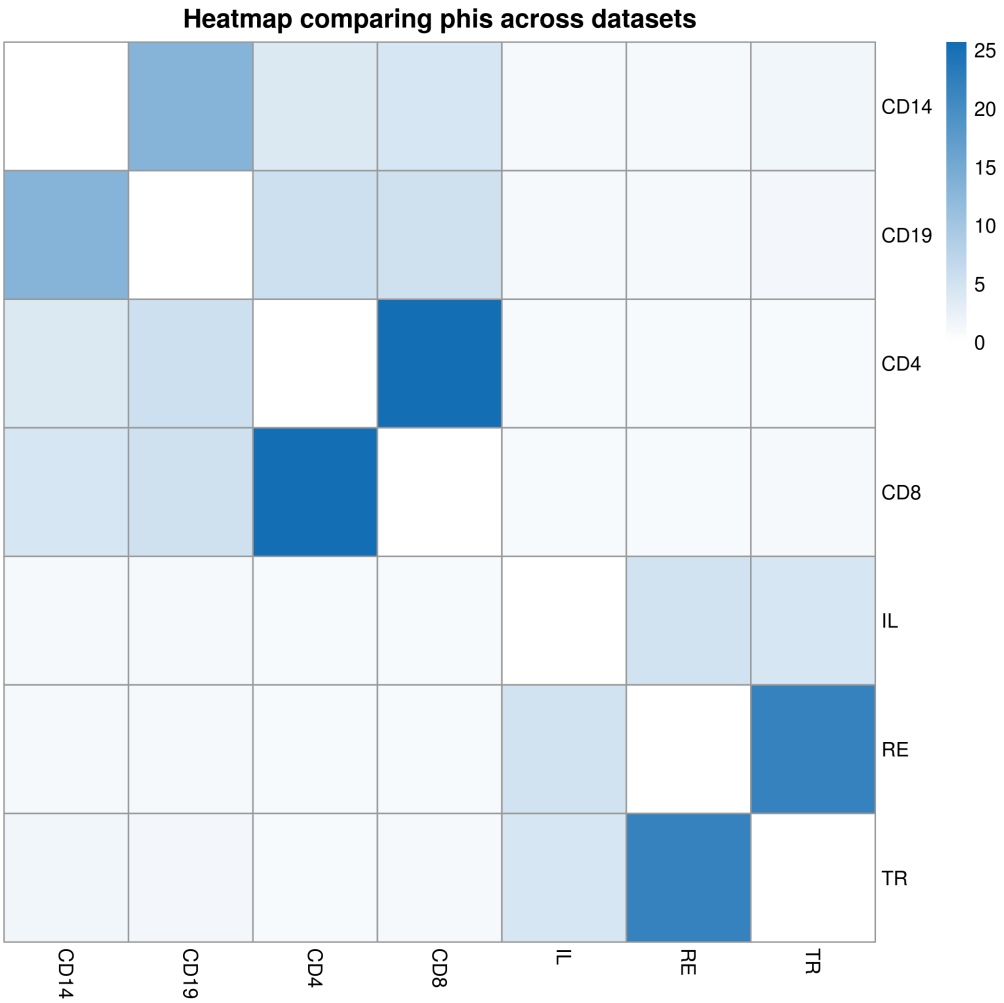

Figure 22: Plot of the mean ϕ_{ij} values across all seeds, between the all datasets.

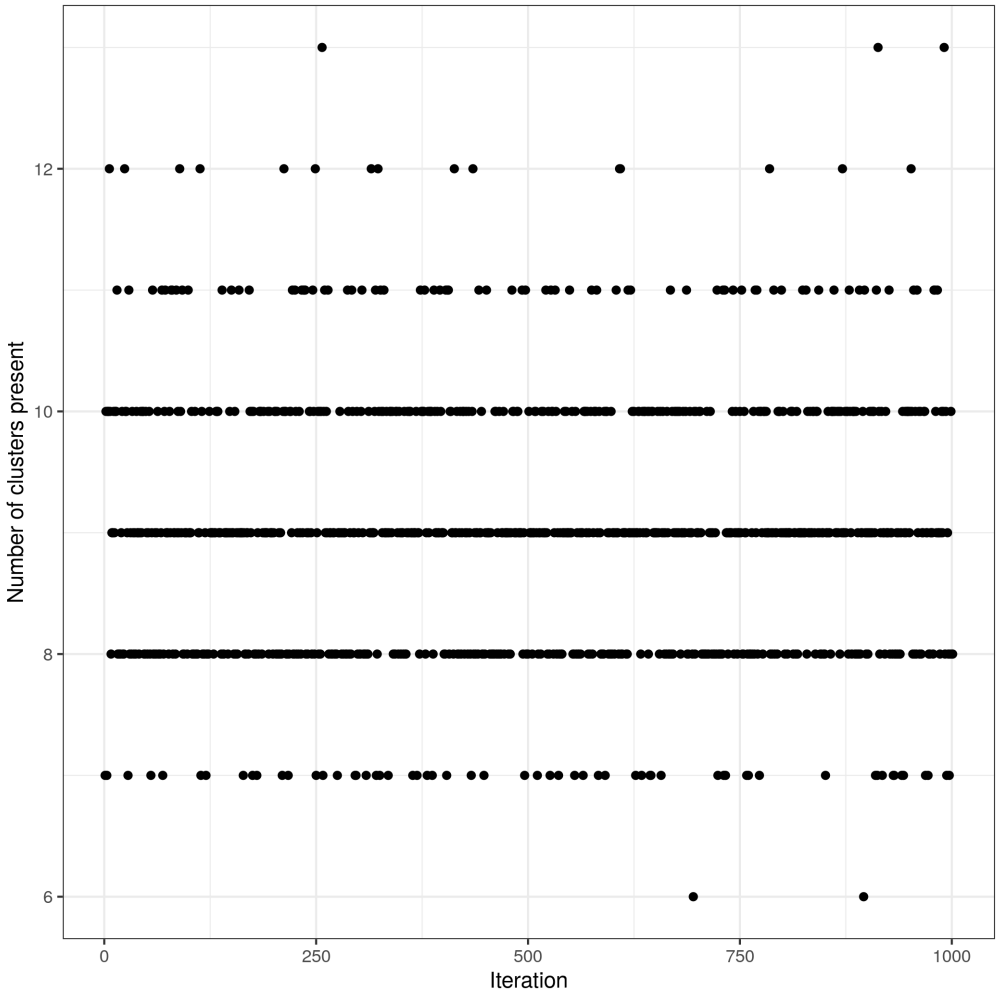

Figure 23: Plot of the number of clusters present in each seed for the CD4 dataset.

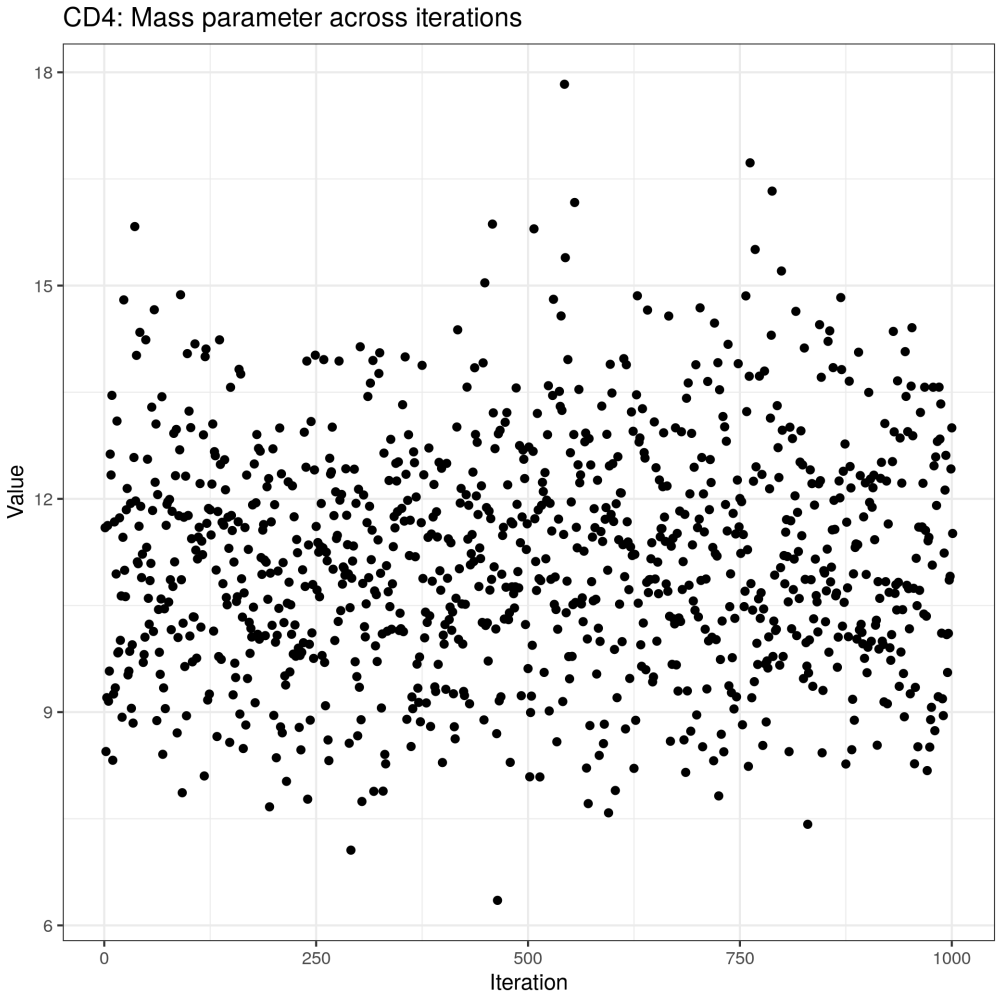

Figure 24: Plot of the mass parameter (α) for the Dirichlet process for the CD4 dataset of MDI.

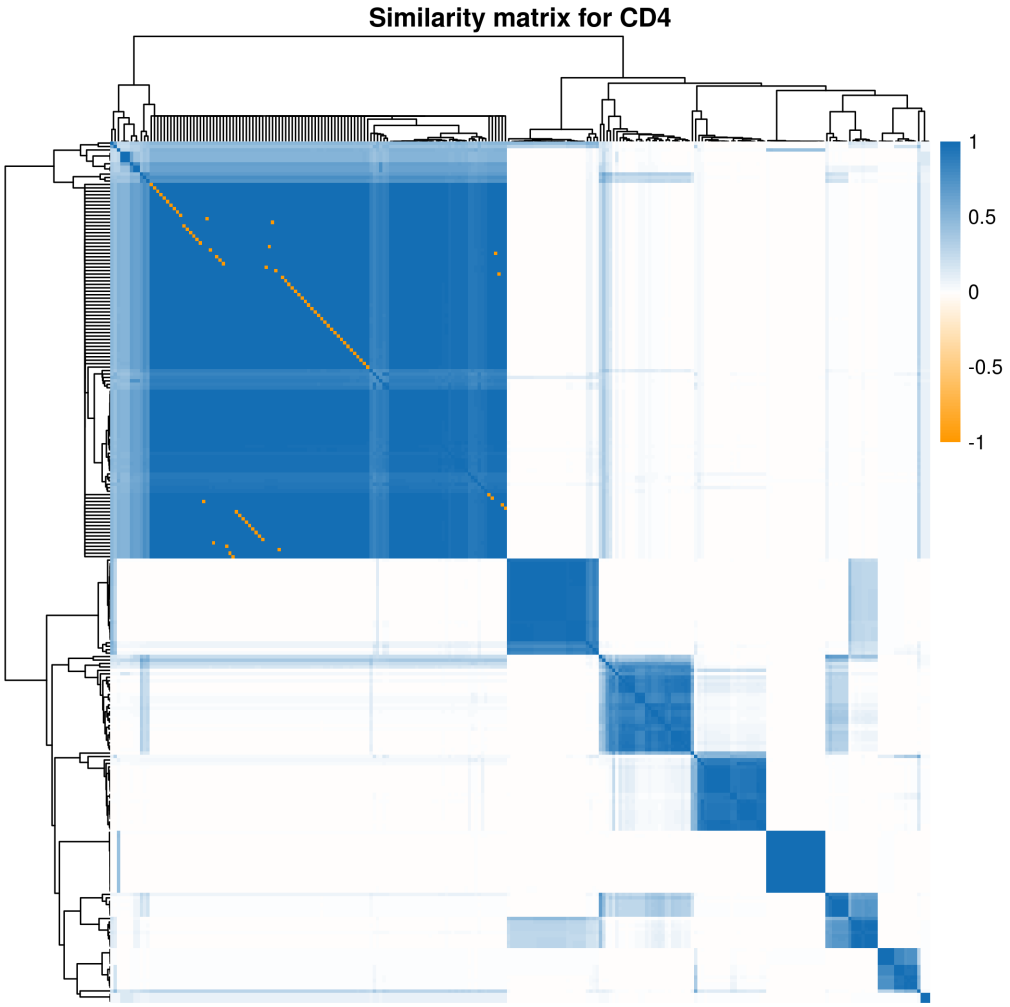

Figure 25: Heatmap of the PSM for the CD4 dataset from the consensus clustering of MDI.

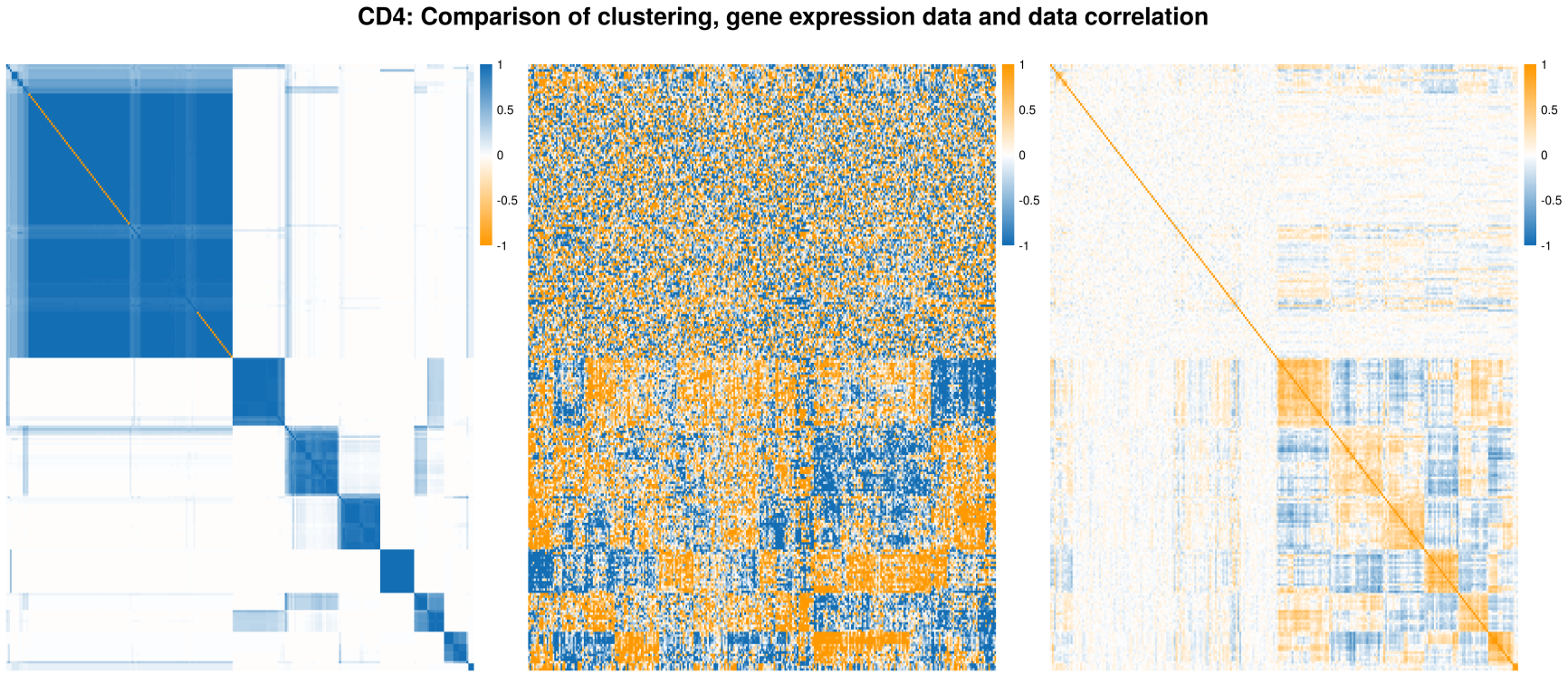

Figure 26: Heatmap of the PSM for the CD4 dataset from the consensus clustering of MDI.

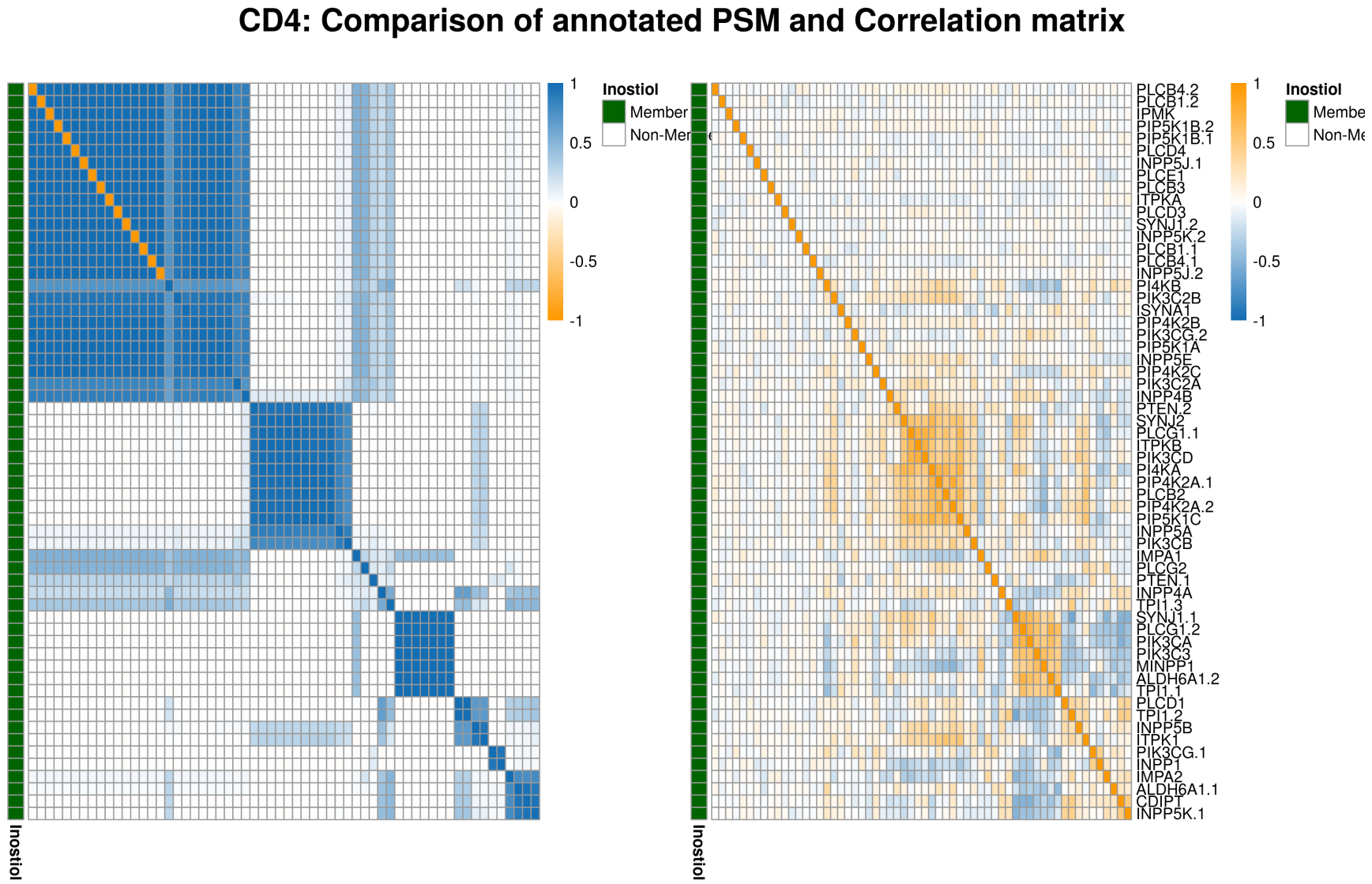

Figure 27: Heatmap of the PSM and expression data for the Inositol genes for the CD4 datasets from the consensus clustering of MDI.

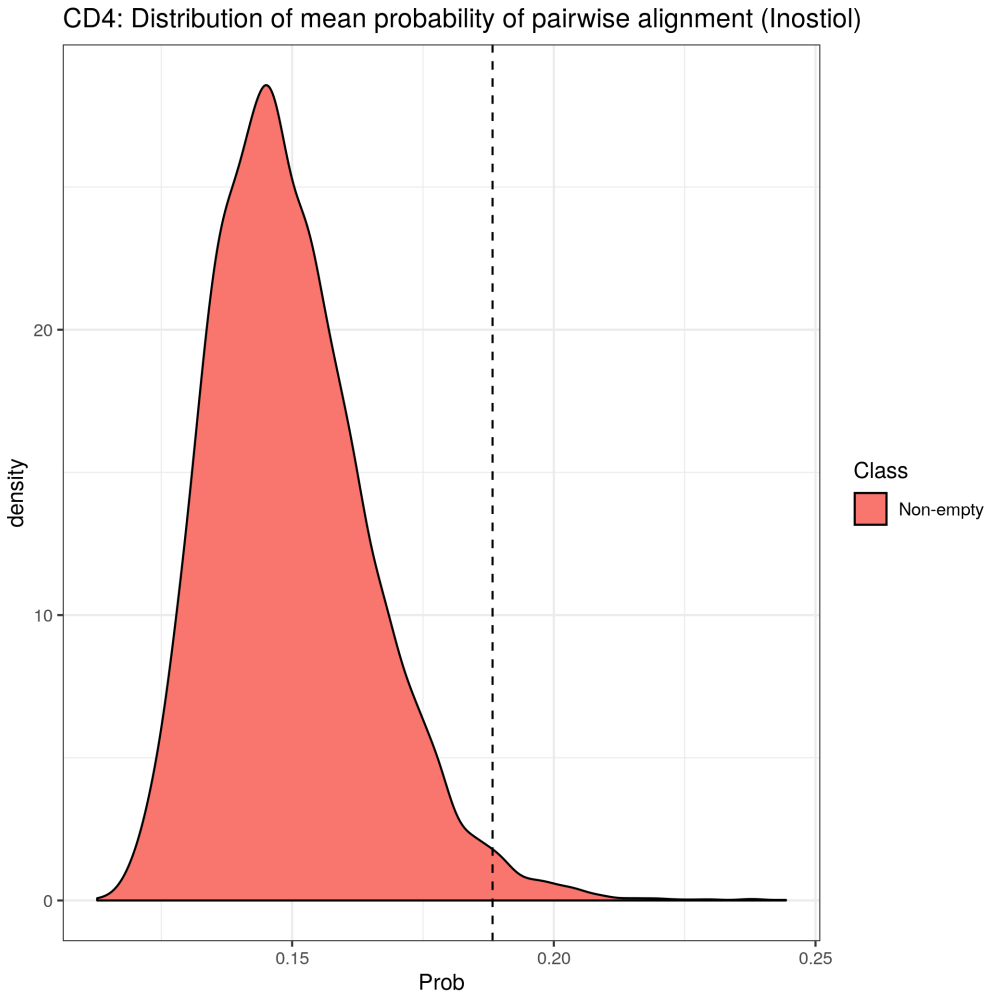

Figure 28: Plot of the distribution of the mean probability of pairwise alignment for a random sample of 60 genes (to coincide with the number of genes associated with the Inositol pathway present) with a dashed line indicating the mean probability of pairwise alignment for the Inositol genes.

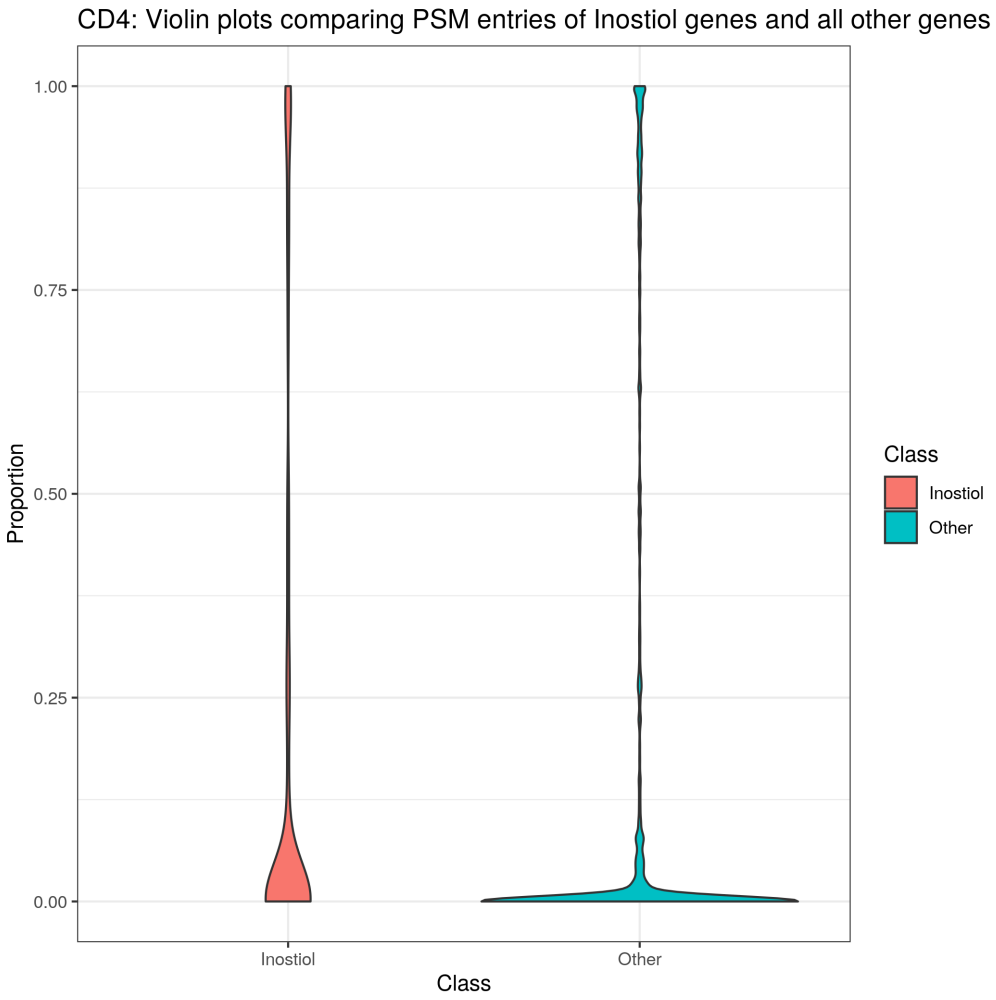

Figure 29: Violin plot of the PSM entries for the Inositol genes and the genes not belonging to this pathway for the CD4 datasets from the consensus clustering of MDI.

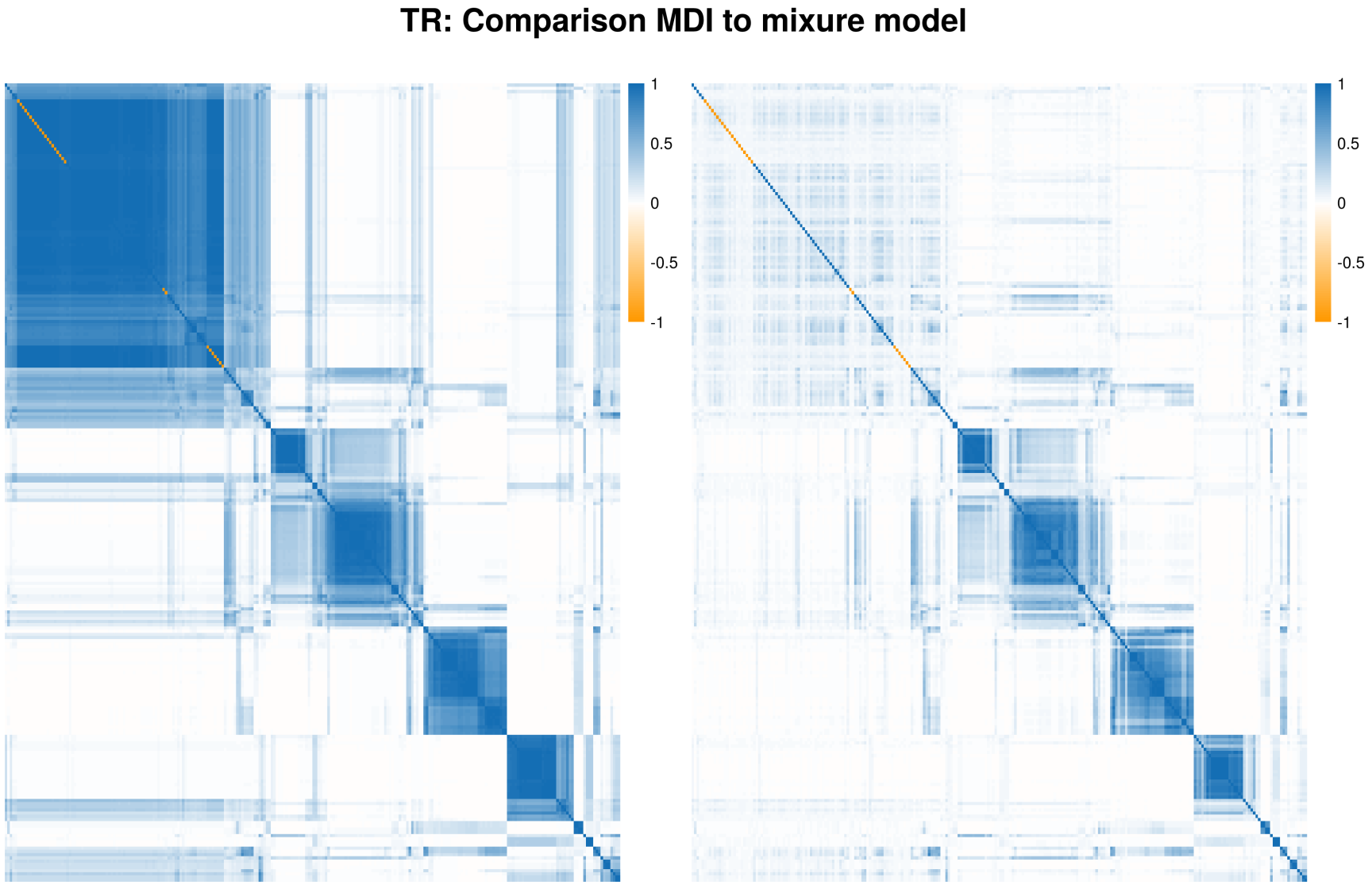

Figure 30: Comparison of the PSMs generated by applying consensus clustering using an MDI model and individual mixture models using only the TR dataset.

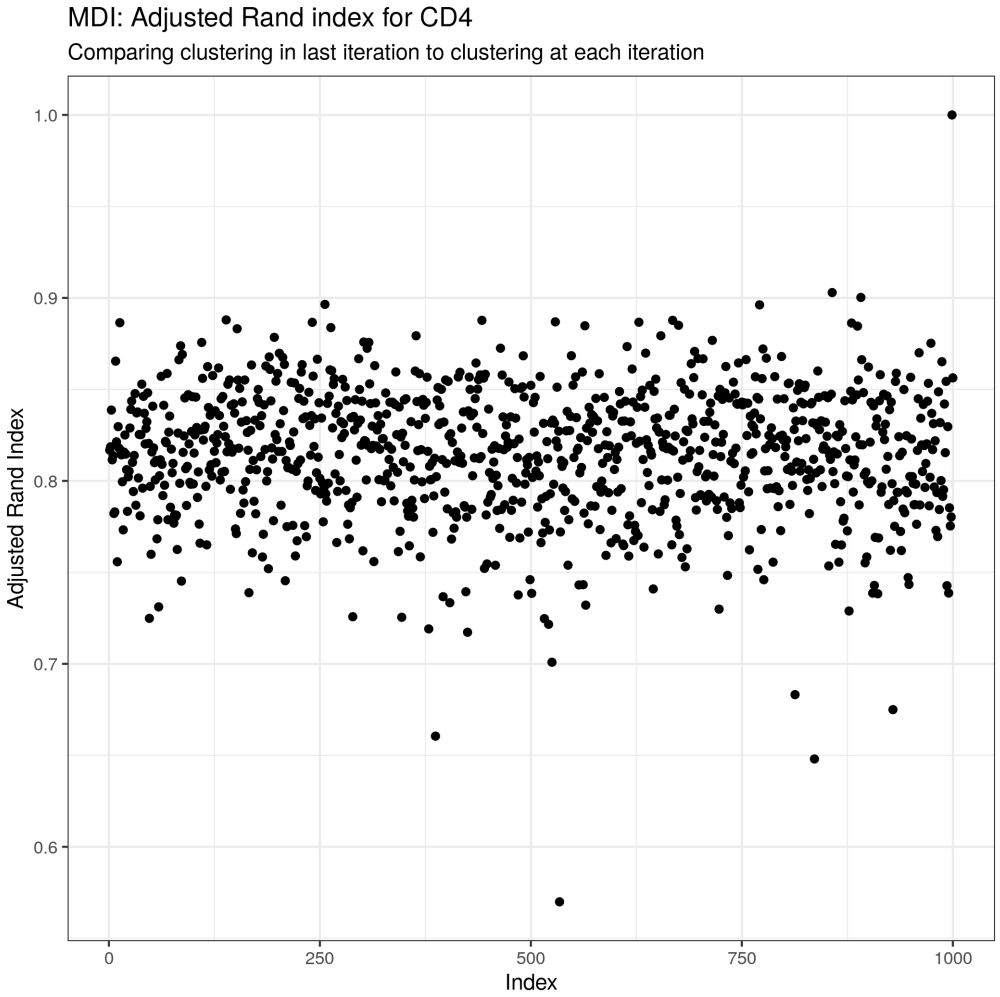

Figure 31: Plot of the adjusted Rand index between the clustering in each seed to that in the 1,000th for CD14.

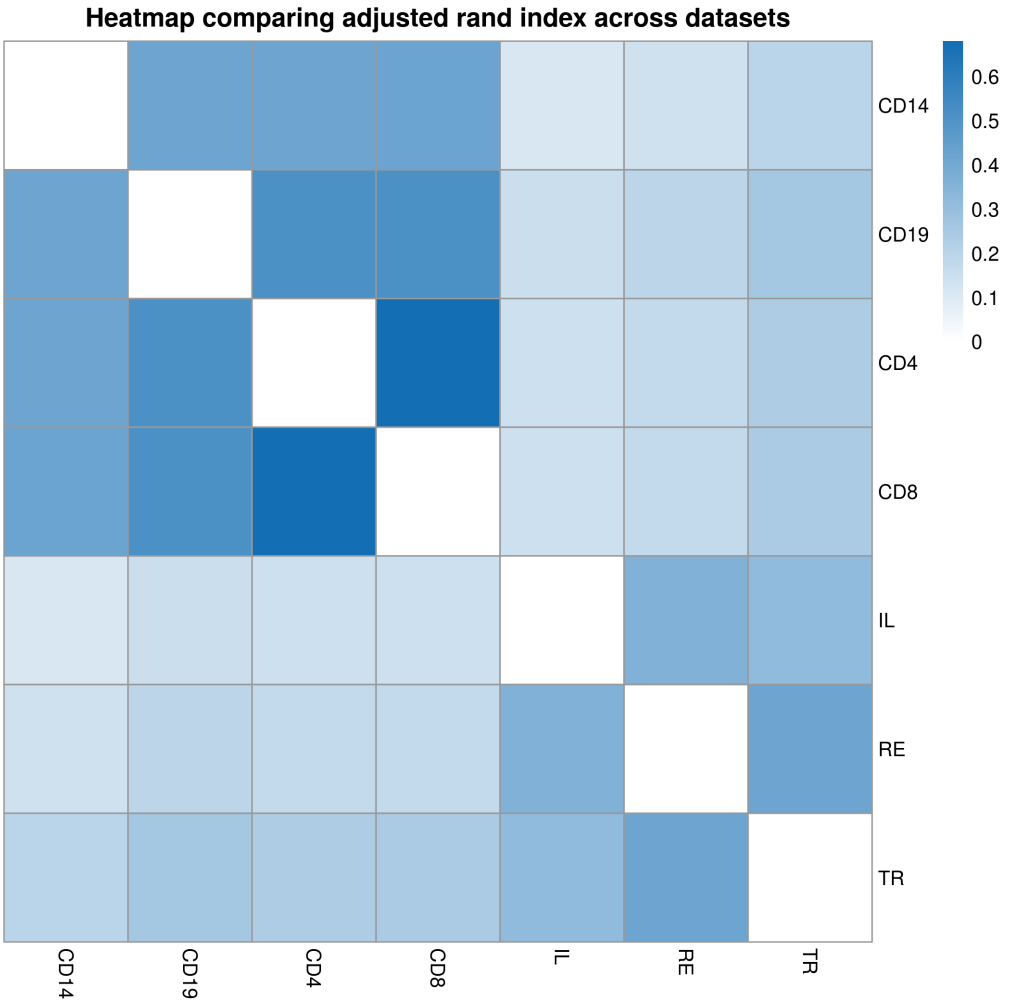

Figure 32: Heatmap of the mean adjusted Rand index comparing the clustering across datasets for each seed.

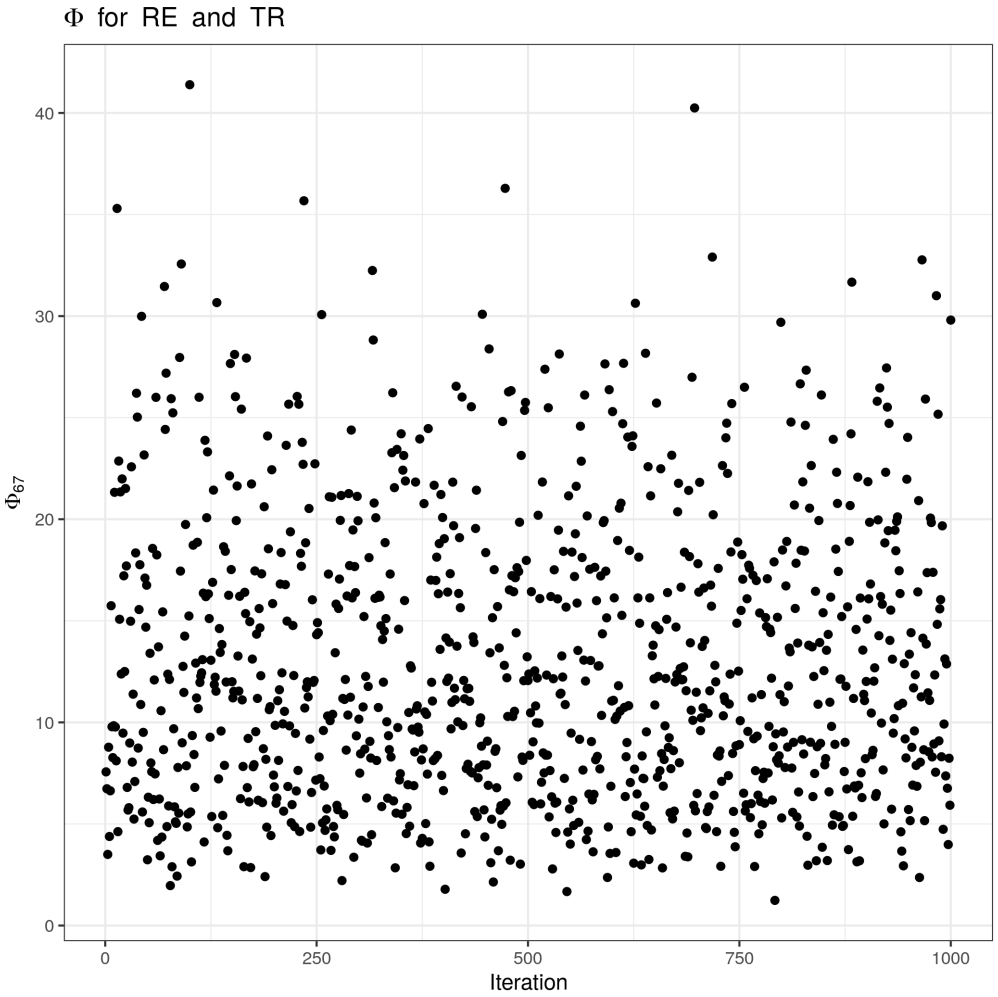

Figure 33: Plot of the ϕ_{67} values across all seeds, between the RE and TR datasets (note that the high values indicate a high clustering correlation).

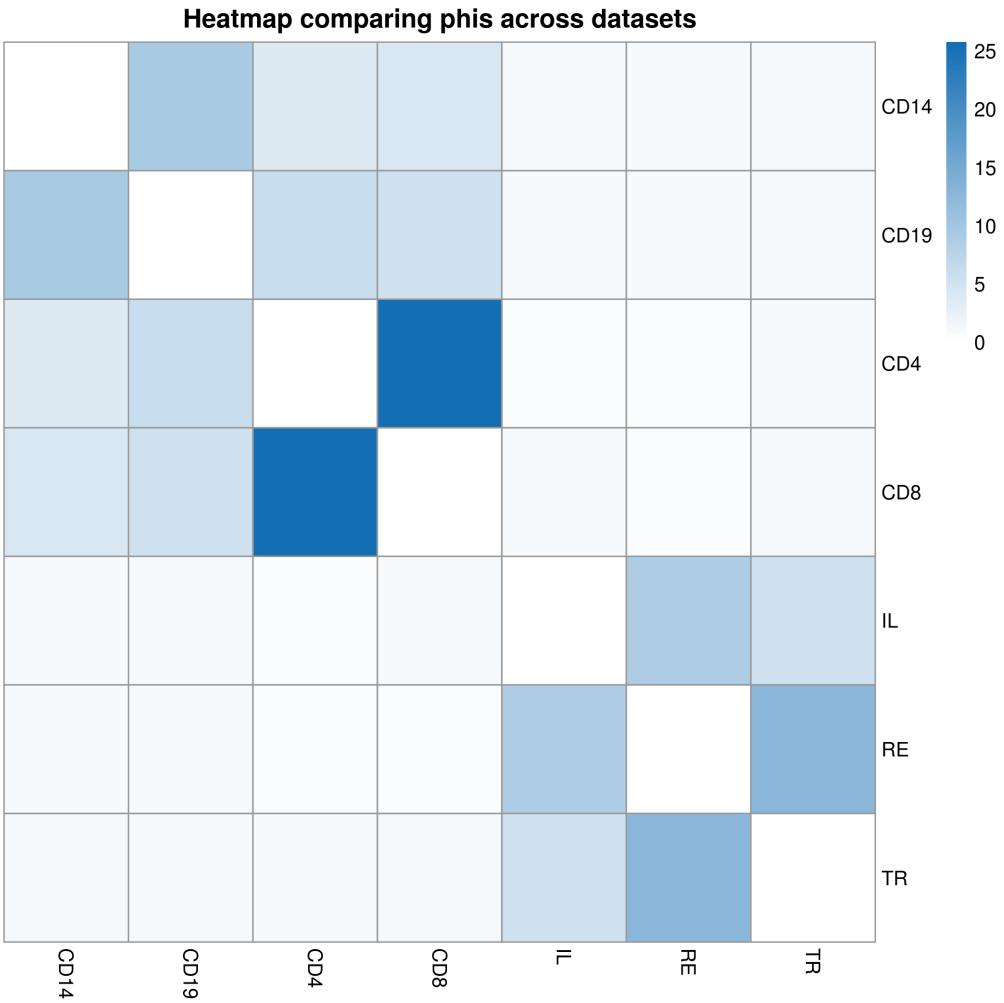

Figure 34: Plot of the mean ϕ_{ij} values across all seeds, between the all datasets.

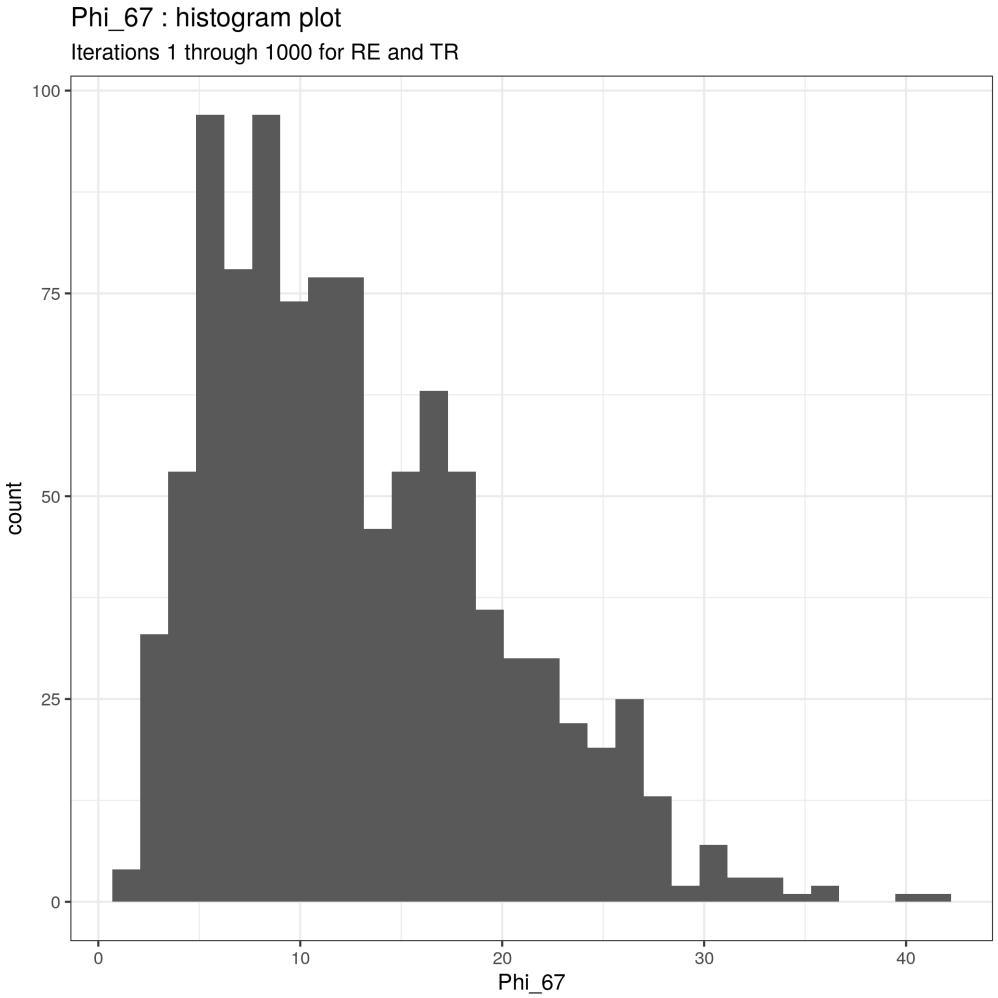

Figure 35: Histogram of the distribution of ϕ_{67} values across seeds(between the RE and TR datasets).

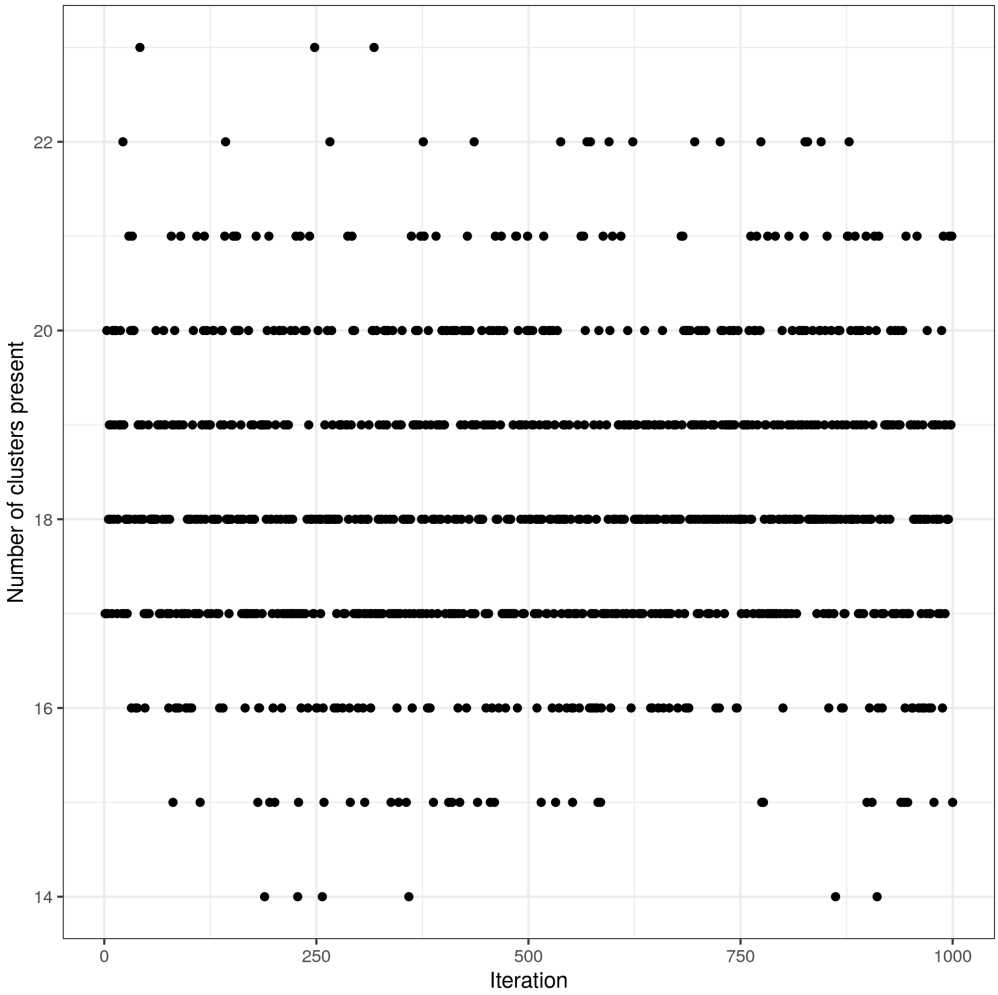

Figure 36: Plot of the number of clusters present in each seed for the CD4 dataset.

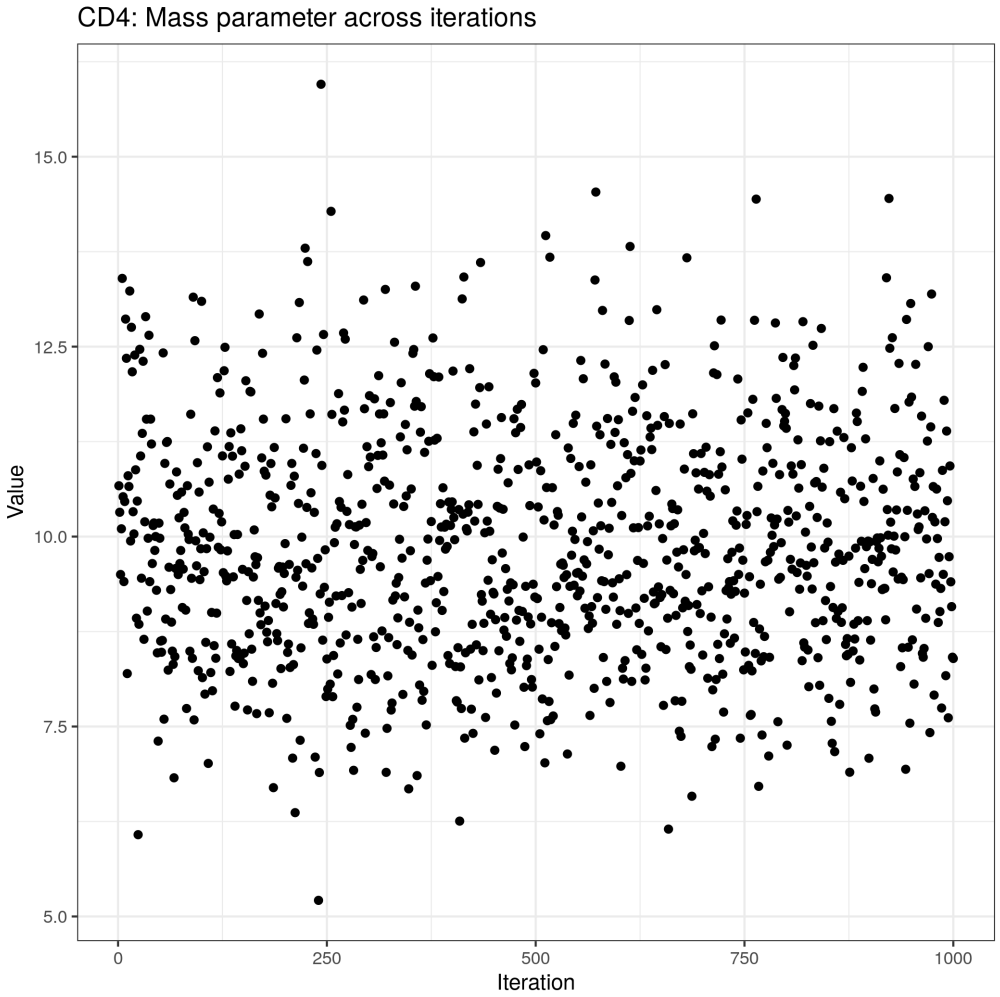

Figure 37: Plot of the mass parameter (α) for the Dirichlet process for the CD4 dataset of MDI.

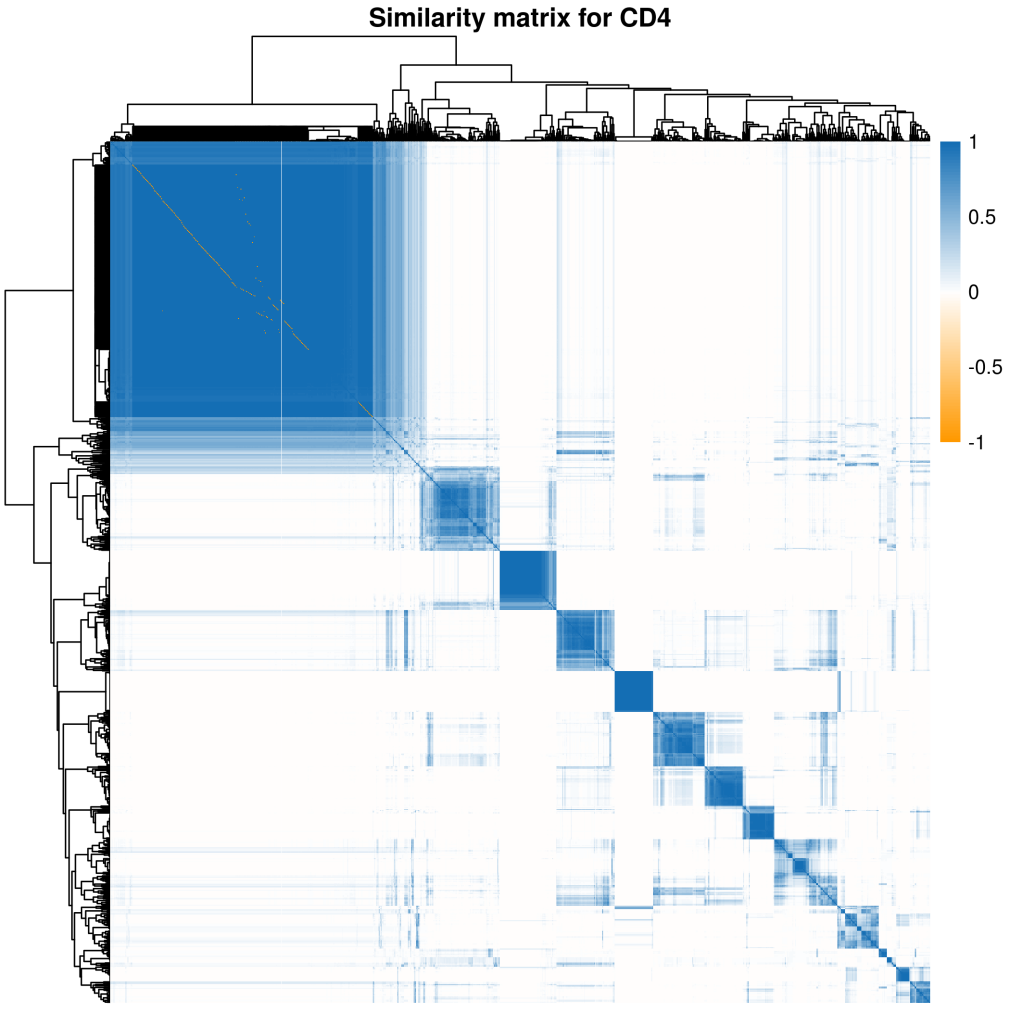

Figure 38: Heatmap of the PSM for the CD4 dataset from the consensus clustering of MDI.

Figure 39: Heatmap of the PSM for the CD4 dataset from the consensus clustering of MDI.

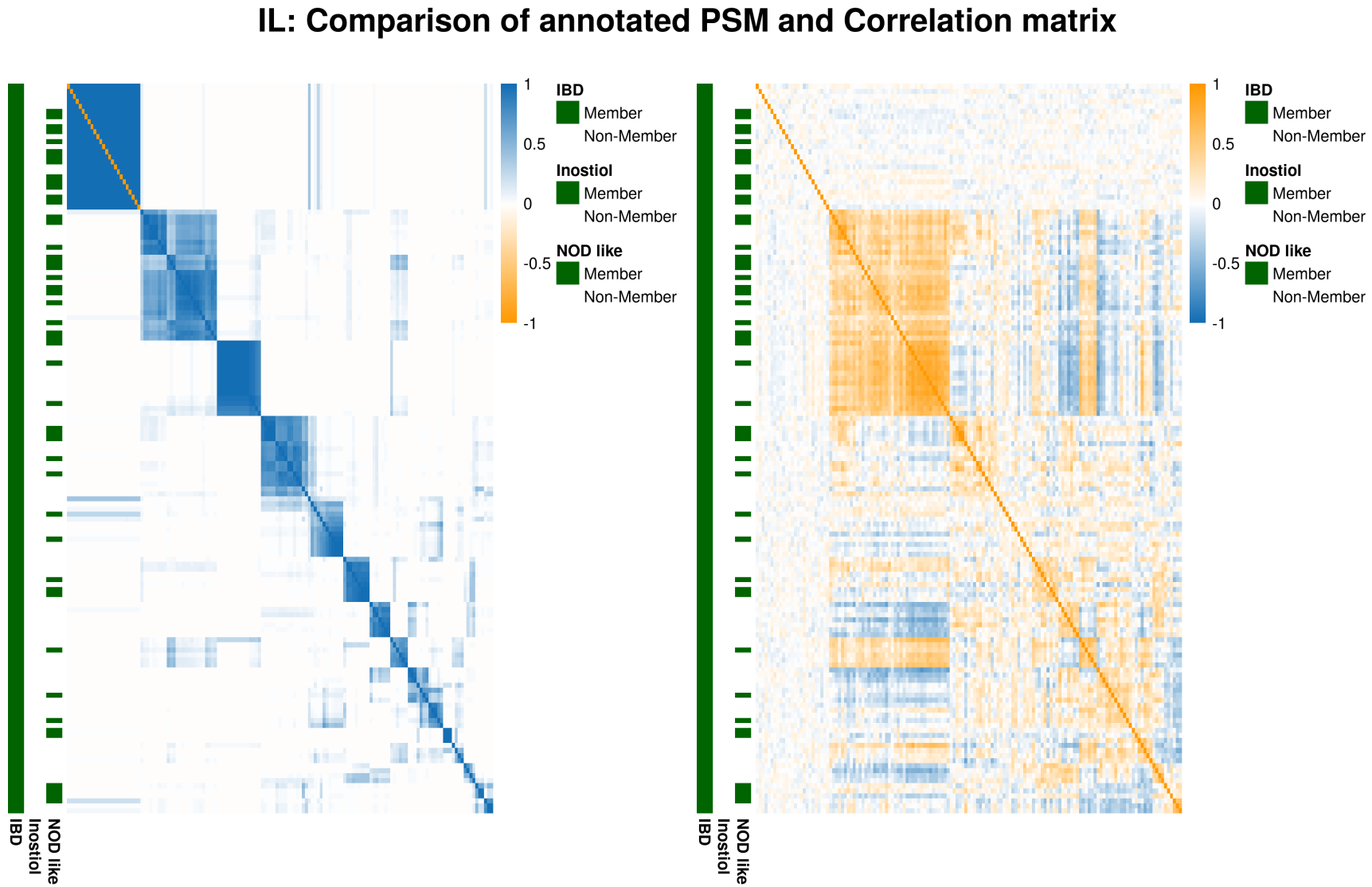

Figure 40: Heatmap of the PSM and expression data for the IBD probes for the IL dataset from the consensus clustering of MDI.

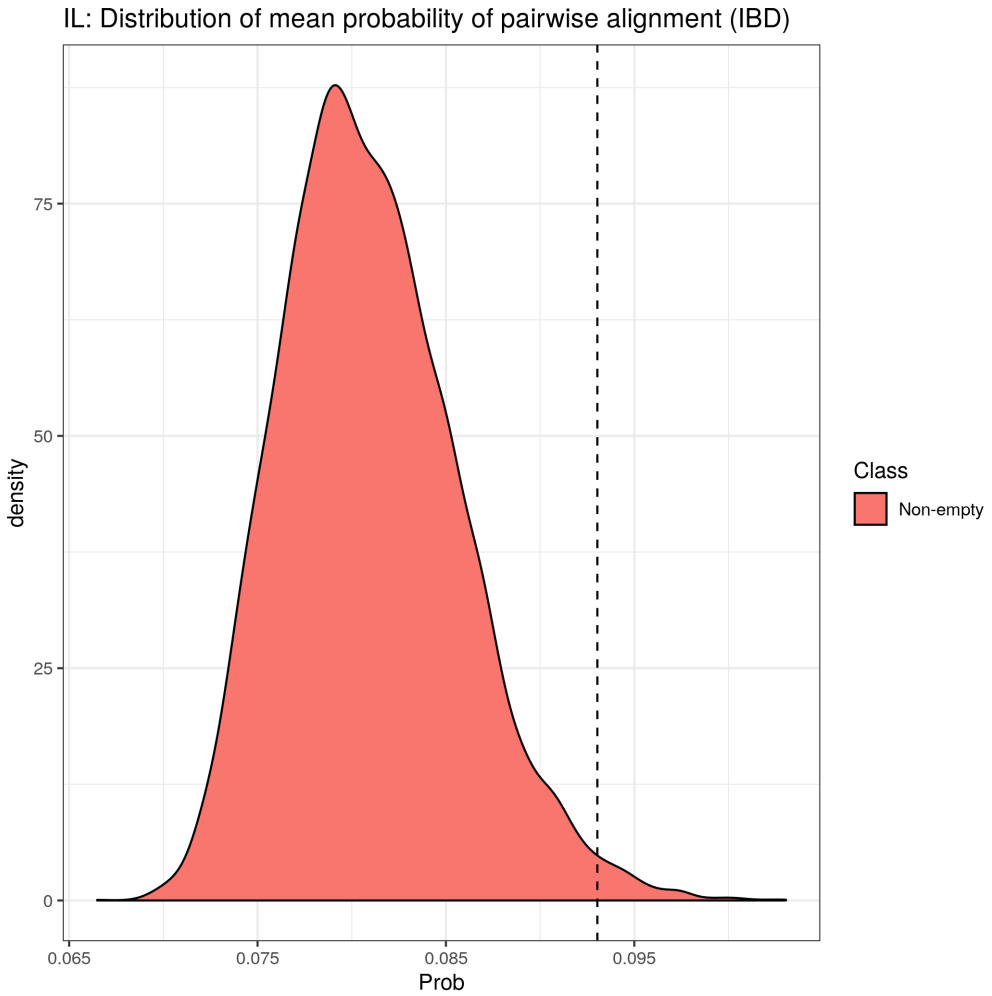

Figure 41: Plot of the distribution of the mean probability of pairwise alignment for a random sample of probes genes with a dashed line indicating the mean probability of pairwise alignment for the IBD associated probes in the IL dataset.

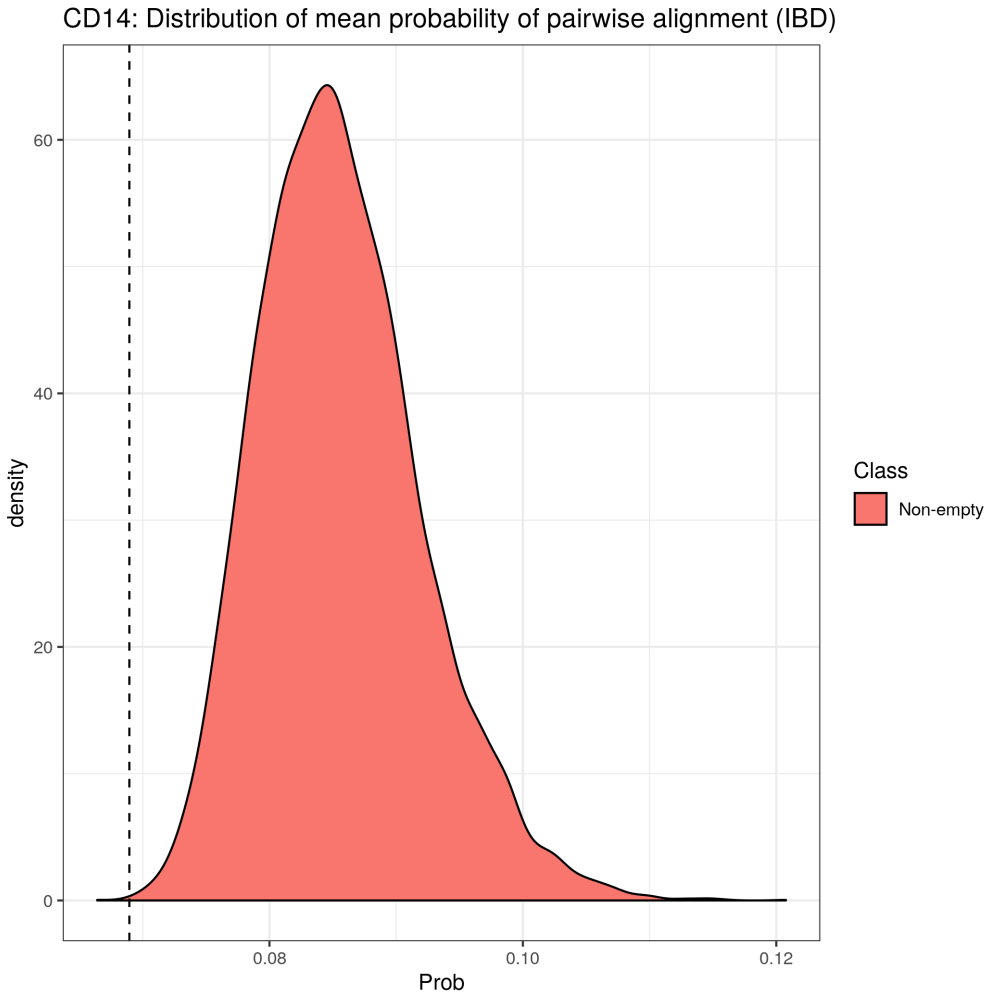

Figure 42: Plot of the distribution of the mean probability of pairwise alignment for a random sample of probes genes with a dashed line indicating the mean probability of pairwise alignment for the IBD associated probes in the CD14 dataset.

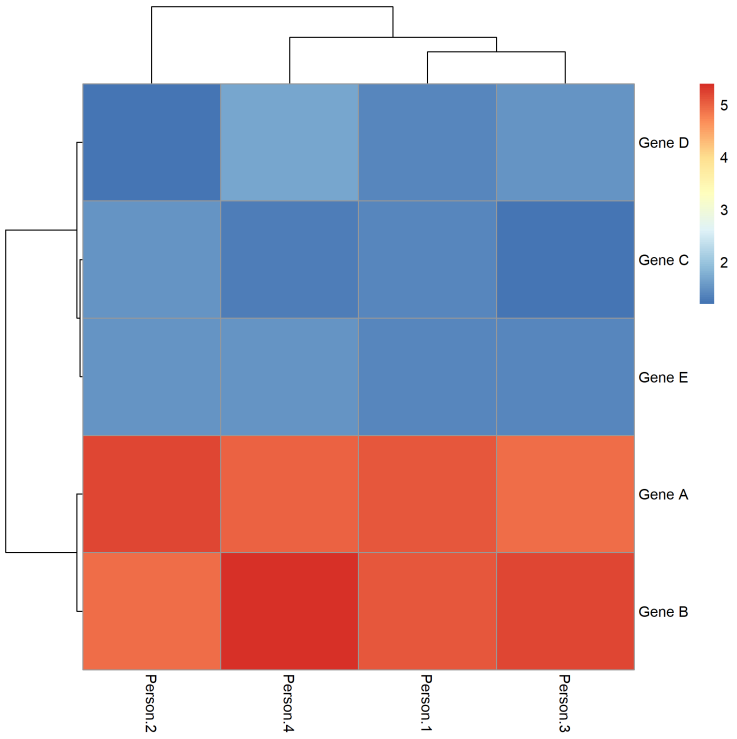

Figure 43: Heatmap of expression data in table 5 showing the clusters based upon magnitude of expression.

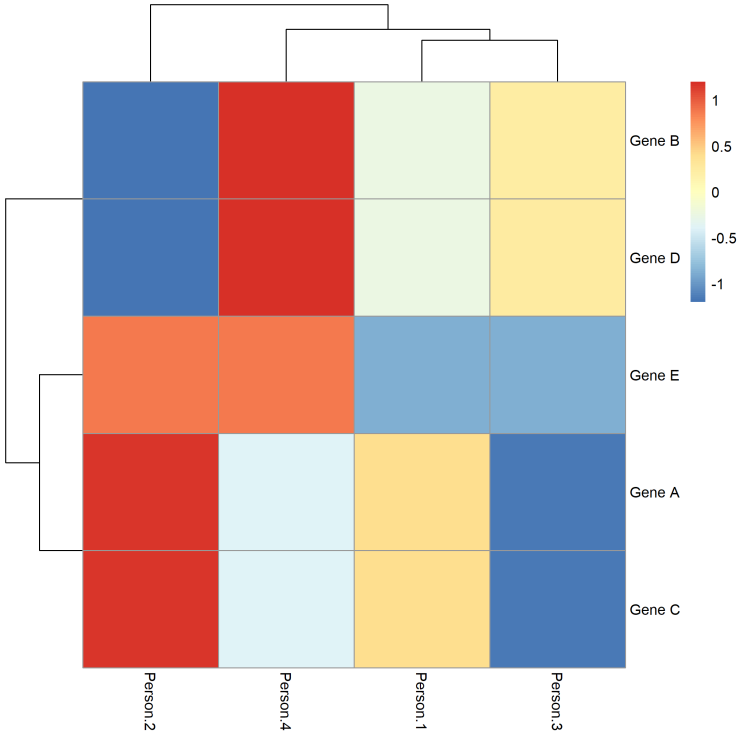

Figure 44: Heatmap of expression data in table 6 showing the clusters based upon variation of expression across people.