# T-augmented Gaussian mixture - multiple dataset integration

Stephen Coleman

January 9, 2019

**Abstract**

tagmmdi is a Bayesian method for semi-supervised prediction using paired datasets. It can be considered an extension of multiple dataset integration (MDI) [20], an unsupervised clustering method utilising Dirichlet Processes, to allow semi-supervised clustering using the t-augmented Gaussian mixture (TAGM) model [8]. We applied tagmmdi to protein localisation using two datasets, mass spectrometry data and Gene ontology (GO) terms. The MS data had a TAGM model applied for prediction, using proteins with experimentally verified organelles as labelled data and a fixed number of clusters. The GO terms were treated as simple categorical data (i.e. we assumed no hierarchy of terms for model parsimony) using an overfitted unsupervised Dirichlet mixture model. The joint model is shown to perform as well as the state-of-the-art method TAGM which itself is compared to other methods in Crook et al. [8].

To implement MDI we require that each dataset share common members of the population in the same order, i.e. observation $i$ in dataset 1 corresponds to observation $i$ in dataset 2 for all $i \in (1, \dots, n)$ for $n \in \mathbb{N}$ observations.

TAGM is implemented as part of the pRoloc package for R available from Bioconductor [4].

The code associated with this report is available at https://github.com/stcolema/tagmmdi/ where instructions on running samples and installing the associated R package can also be found.

## A comment on notation

We use the following symbols throughout this piece.

- $n \in \mathbb{N}$: the number of observations in each dataset;
- $x_i$: the $i$th observation for some $i \in \{1, \dots, n\}$;
- $L \in \mathbb{N}$: the number of datasets;
- $K_l \in \mathbb{N}$: the number of components in dataset $l$ for some $l \leq L$. If $L = 1$ we do not include the subscript;
- $c_{il} \in \{1, \dots, K_l\}$: the latent clustering variable for the $i$th observation in the $l$th dataset;
- $Z \in \mathbb{R}$: a relevant normalising constant;
- $\mathbb{R}_+ = \{x \in \mathbb{R} | x > 0\}$;

- $\phi \in \mathbb{R}_+$: the context similarity parameter, a measure of the similarity between datasets 1 and 2;
- $\pi_j \in [0, 1] \subset \mathbb{R}$: the component proportions within the dataset for some $j \leq K$; and
- $\gamma_{jl} \in \mathbb{R}_+$: the component weights for the $j$th component in dataset $l$.

# 1 Theory

We explain some of the concepts upon which the tagmmdi model is built and briefly describe how the datasets used to produce the results are created.

## 1.1 Gibbs sampler

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.

### 1.1.1 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] as a means to solving integrals by use of random number generation [23]. Suppose there is some complex integral we wish to solve on some interval, $(a, b)$:

$$\int_a^b h(x)dx \tag{1}$$

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:

$$\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = \mathbb{E}_{p(x)}\left[f(x)\right] \tag{2}$$

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 (2) becomes:

$$\int_a^b h(x)dx = \mathbb{E}_{p(x)}\left[f(x)\right] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{3}$$

This is *Monte Carlo integration*.

### 1.1.2 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 $s_i, s_j, s_k \in S$:

$$\mathbb{P}(X(t_{n+1}) = s_j | X(t_0) = s_k, \ldots, X(t_n) = s_i) = \mathbb{P}(X(t_{n+1}) = s_j | X(t_n) = s_i) \tag{4}$$

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.

We define the $n$-step transition probability $p_{ij}^n$ as the probability that the process is in state $j$ given it was in state $i$ $n$ steps ago, i.e.:

$$p_{ij}^n = \mathbb{P}(X(t_{i+n} = s_j | X(t_i) = s_i) \tag{5}$$

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

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

$$\mathbb{P}(X(t_{n+1}) = x | X(t_n) = y)\pi_x = \mathbb{P}(X(t_{n+1}) = y | X(t_n) = x)\pi_y \tag{6}$$

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.

### 1.1.3 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 (2). Our goal in the following is to draw samples from some distribution $p(\theta)$ where we have some distribution $f(\theta)$ such that:

$$p(\theta) = \frac{f(\theta)}{K} \tag{7}$$

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

The Metropolis-Hastings algorithm [16] 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 [23] [24] generates a sequence of draws from $p(\theta)$ using the following steps:

1. Initialise with some arbitrary value $\theta_0$ 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 we demand that this be symmetric (i.e. $q(\theta_1|\theta_2) = q(\theta_2|\theta_1)$).
2. For each iteration, $t$:
   (a) Using the current value $\theta_{t-1}$, sample a *candidate point*, $\theta^*$, from $q(\theta^*|\theta_{t-1})$.
   (b) Calculate the *acceptance ratio* for the new value $\theta^*$:
   $$\alpha = \frac{p(\theta^*)}{p(\theta_{t-1})} = \frac{f(\theta^*)}{f(\theta_{t-1})} \tag{8}$$
   Note that as the proportionality constant is the same for all $\theta$ that this is an equivalence rather than proportional relationship.
   (c) Accept the new value $\theta^*$ with probability $\min(\alpha, 1)$. 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 $l$th sample is recorded (called *thinning*) for some small $l$ (normally in the range $[25, 50]$).

Hastings [16] extends this method to allow asymmetric proposal densities, in which case our acceptance ratio changes to:

$$\alpha = \min\left(\frac{f(\theta^*)q(\theta^*|\theta_{t-1})}{f(\theta_{t-1})q(\theta_{t-1}|\theta^*)}, 1\right) \tag{9}$$

Geman and Geman [12] 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 we are 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 (12)). Thus with a sufficient burn-in period we can use MCMC methods to sample directly from the posterior distribution without directly calculating the normalising constant.

## 1.2 Mixture models

Given some data $X = (x_1, \ldots, x_n)$, we assume a number of unobserved processes generate the data, and membership to a process for individual $i$ is represented using the latent variable $c_i$. It is assumed that each of the $K$ processes can be modelled by a parametric distribution, $f(\cdot)$ with associated parameters $\theta$ and that the full model density is then the

weighted sum of these probability density functions where the weights are the component proportions, $\pi_k$:

$$p(x_i) = \sum_{k=1}^{K} \pi_k f(x_i|\theta_k) \tag{10}$$

We carry out Bayesian inference of this model using MCMC methods. Specifically we use a Gibbs sampler. We sample first the component parameters, $\theta_k$, and associated weights, $\pi_k$, from the associated distributions and then sample component membership.

Basically:

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

Each individual's membership probabilities are conditionally independent of the other memberships given the cluster parameters:

$$p(c_i|c_{-i}, \theta_1, \ldots, \theta_K, \pi_1, \ldots, \pi_K) = p(c_i|\theta_1, \ldots, \theta_K, \pi_1, \ldots, \pi_K) \tag{11}$$

Where $c_{-i} = (c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n)$. Thus our problem is *embarrassingly parallel*. This is part of the reason we use this method rather than a *collapsed Gibbs sampler*. Instead of sampling the parameters each iteration a collapsed Gibbs sampler marginalises them (i.e. integrates over them) and updates them as each individual's allocation is updated. This method tends to reduce the number of iterations required before stationarity is reached [21], but each iteration is slower and the method is more difficult to implement.

The distribution we sample from for each parameter, $\theta$, is updated after observing data $X$ using Bayes' theorem:

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int_\Theta p(X|\theta')p(\theta')d\theta'} \tag{12}$$

Here $\Theta$ is the entire sample space for $\theta$.

- We refer to $p(\theta|X)$ as the *posterior* distribution of $\theta$ as it is the distribution associated with $\theta$ *after* observing $X$.
- $p(\theta)$ is the *prior* distribution of $\theta$ and captures our beliefs about $\theta$ before we observe $X$.
- $p(X|\theta)$ is the *likelihood* of $X$ given $\theta$, the probability of data $X$ being generated given our model is true. It is the criterion we focus on in our model if we would use a frequentist approach to the inference; maximising this quantity in our model generates the curve that best describes the observed data.
- $\int_\Theta p(X|\theta')p(\theta')d\theta'$ is the *normalising constant*. This quantity is also referred to as the *evidence* [22] or *marginal likelihood* and is normally represented by $Z$. It is referred to as the marginal likelihood as we marginalise the parameter $\theta$ by integrating over its entire sample space.

In terms of sampling the prior is very useful as it allows us to ensure that the posterior is always solvable, that we do not encounter singularities in our distribution.

Our implementation uses distributions on the priors that enforce conjugacy. This allows us to sample directly from the correct distribution for each posterior distribution.

## 1.3 Conjugate priors

We use *conjugate priors* to make sampling easier. Conjugate priors are a family of distributions such that for a likelihood of a given distribution the posterior is of the same family as the prior. As a consequence we obtain a closed, tractable integral for the posterior.

For the multivariate normal distribution of unknown mean and variance the conjugate prior is a normal-inverse-Wishart (NIW) with 4 associated prior parameters:

- $\mu_0$ - prior on the mean, a $p$-vector where $p$ is the number of features in the data;
- $\lambda_0$ - the shrinkage on the variance, a positive real number;
- $\nu_0$ - the degrees of freedom in the inverse-Wishart, a positive integer; and
- $\Psi_0$ - the inverse scale matrix for the inverse-Wishart distribution, a positive definite $p \times p$ matrix.

For the Categorical distribution this is a Dirichlet with prior concentration parameter $\alpha_0$.

### 1.3.1 Continuous

For the continuous dataset we use a mixture of multivariate Gaussian models with each mixture defined by parameters $\mu$ adn $\Sigma$, the mean and covariance respectively. After observing a sample of $n$ individuals allocated to mixture component $k$ we update the associated parameters, $\mu$ and $\Sigma$, by drawing from the NIW. That is:

$$(\mu, \Sigma) \sim \text{NIW}(\mu_n, \lambda_n, \Psi_n, \nu_n) \tag{13}$$

The pdf of this is:

$$f(\mu, \Sigma | \mu_n, \lambda_n, \Psi_n, \nu_n) = \mathcal{N}\left(\mu | \mu_n, \frac{1}{\lambda_n}\Sigma\right) \mathcal{W}^{-1}\left(\Sigma | \Psi_n, \nu_n\right) \tag{14}$$

To calculate the updated hyperparameters for the sample, first let:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{15}$$

$$S = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T \tag{16}$$

Then update the hyperparameters using:

$$\mu_n = \frac{\lambda_0 \mu_0 + n\bar{x}}{\lambda_0 + n} \tag{17}$$

$$\lambda_n = \lambda_0 + n \tag{18}$$

$$\nu_n = \nu_0 + n \tag{19}$$

$$\Psi_n = \Psi_0 + S + \frac{\lambda_0 n}{\lambda_0 + n}(\bar{x} - \mu_0)(\bar{x} - \mu_0)^T \tag{20}$$

### 1.3.2 Categorical

To obtain the posterior distribution in the categorical data we update our Dirichlet prior for each mixture by adding the count of the individuals allocated to said mixture. That is for a prior concentration parameter $\alpha_0$, for a mixture $k$ we update as so:

$$\alpha_{nk} = \alpha_0 + \sum_{i=1}^{n} \mathbb{I}(c_i = k) \tag{21}$$

## 1.4 Multiple dataset integration

If we have 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 $i$th row of each dataset represent information about the same individual. We would like to cluster using information common to both datasets. One could concatenate the datasets, adding additional covariates for each individual. However, if the two datasets have different clustering structures this would reduce the signal of both clusterings and probably have one 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. [20] 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:

$$p(c_{i,1}, c_{i,2}|\phi) \propto \pi_{i,1}\pi_{i,2}(1 + \phi\mathbb{I}(c_{i,1} = c_{i,2})) \tag{22}$$

Here $\phi \in \mathbb{R}_+$ controls the strength of association between datasets. $\mathbb{I}(\cdot)$ is the indicator function. (22) 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 $\phi$ if the individual has the same labelling in each dataset. Thus as $\phi$ 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. Note that Kirk et al. [20] include the generalised case for $L$ datasets for any $L \in \mathbb{N}$.

MDI has been applied to precision medicine, specifically glioblastoma sub-typing [27], in the past showing its potential as a tool.

## 1.5 Protein localisation

Protein localisation is a fundamental question as localisation to the correct location is required for interaction with its binding partners and to carry out its function [14]. These cellular components also provide the ideal biochemical environment for the proteins to function. Aberrant localisation is associated with a multitude of diseases including many subtypes of cancer, obesity and cardiovascular disease [28][18][19]. A thorough understanding of the biology behind protein localisation is important to a better understanding of these diseases. Possible translational implications of this understanding are diagnostic tools such as blood or urine tests based on protein localisation or drugs to correct mislocalisation as part of treatment [19][17].

The protein localisation data is produced using synchronous precursor selection (SPS)-based $MS^3$ technology using the LOPIT and *hyper*LOPIT pipelines [11][9]. In summary:

1. The cells undergo lysis in such a way as to maintain the integrity of their organelles.
2. The cell content is then separated along a density gradient. Thus, organelles and macro-molecular complexes are described by density-specific profiles along the gradient.
3. Discrete fractions along the density gradient are collected.
4. Within these fractions, quantitative protein profiles are measured using high accuracy mass spectrometry.

Thus for each fraction we have a description of the proteins present. The normalised incidence of the protein across fractions is found to follow a specific pattern unique to each organelle's associated proteins. From pre-exiting microscopy experiments we have some proteins with known associations to certain organelles. This allows use of supervised and semi-supervised methods to predict the localisation of the unlabelled data.

## 1.6 T-augmented Gaussian mixture models

T-augmented Gaussian mixture (TAGM) models are a semi-supervised prediction method using Gaussian mixture models (i.e. models as described in 1.2 where the distribution $f$ is restricted to the Normal distribution) with a t-distribution as a "junk" or outlier distribution. The model is defined:

$$p(x_i|\theta, \pi, \kappa, \epsilon, M, V) = \sum_{k=1}^{K} \pi_k((1-\epsilon)f(x_i|\mu_k, \Sigma_k) + \epsilon g(x_i|\kappa, M, V)) \tag{23}$$

Where:

- $\theta$ is the component specific parameters for the component Gaussian distribution (here $\mu_k$, $\Sigma_k$);
- $\pi_k$ is the mixture proportion;
- $\kappa$ is the degrees of freedom for the global outlier distribution;
- $\epsilon$ is the outlier component weight;

- $M$ is the global mean used as the mean in the outlier distribution;
- $V$ is half the global covariance, used in the outlier distribution;
- $f(\cdot)$ is the probability density function for the multivariate normal distribution; and
- $g(\cdot)$ is the probability density function for a t-distribution.

Similar to the method outlined in 1.2, this model iterates over these steps:

1. Sample component parameters based on current allocation;
2. Sample component weights based on current allocation;
3. Sample outlier distribution weight based on current allocation;
4. Sample component allocation for each individual;
5. Given the above allocation, sample membership of the outlier distribution; and
6. Repeat.

The proteins allocated as outliers contribute to the mixing proportions but not the component parameters. One of the advantages associated with tagm models is that they quantify the uncertainty of the allocation. This distribution of allocation probability across the $K$ organelles allows greater interpretation of the allocation and as Crook et al. [8] show can lead to the concept of multiple membership.

## 1.7 GO terms

Gene ontology terms describe characteristics of gene-products in three non-overlapping areas of molecular biology [13].

- Molecular function: biochemical activities at the molecular level such as catalytic or binding activity. Examples of broad functional terms are 'enzyme', 'transporter' or 'ligand'. Examples of narrower functional terms are 'adenylate cyclase' or 'Toll receptor ligand' [1].
- Biological processes: a biological objective to which the gene or gene product contributes. A process is accomplished via one or more ordered assemblies of molecular functions [1]. Examples of broad (high level) biological process terms are 'cell growth and maintenance' or 'signal transduction'. Examples of more specific (lower level) process terms are 'translation', 'pyrimidine metabolism' or 'cAMP biosynthesis' [1].
- Cellular component: the location within the cell where a gene product is active, i.e. the component it localises to.

As one can see in the examples provided by Ashburner et al. [1], there is a hierarchy within the terms - the terms may be described using a directed acyclic graph. In our implementation we ignore this hierarchy and treat the terms as a matrix of binary variables. This simplifying assumption is made for the sake of computational complexity. We choose GO terms as an additional dataset as they contain the same proteins as our MS data (i.e. can be used within the MDI framework), are freely available and are expected to provide some additional information regarding the prediction of localisation based on the results of Breckels et al. [3]. The GO terms are a summary dataset in some way - they are not experiment

specific and not cell-type specific, and hence are expected to contain a lower signal-to-noise ratio for protein localisation then the MS data.

### 1.8 tagmmdi

Within each dataset we assume a common prior for each mixture. For the MS data with some known labels we fit a TAGM model where $K$ is equal to the number of organelles present in the training data. These points are held fixed in their allocation across all iterations and the associated component is considered to represent allocation to the associated organelle. We overfit a mixture of Dirichlet models on the GO term dataset (i.e. we set the number of cluster to an arbitrarily high number). MDI is applied and the output of interest for prediction is the posterior similarity matrix. As each component represents a specific organelle, proteins with a greater frequency of allocation to the same component are predicted to localise to the associated organelle.

## 2 Results

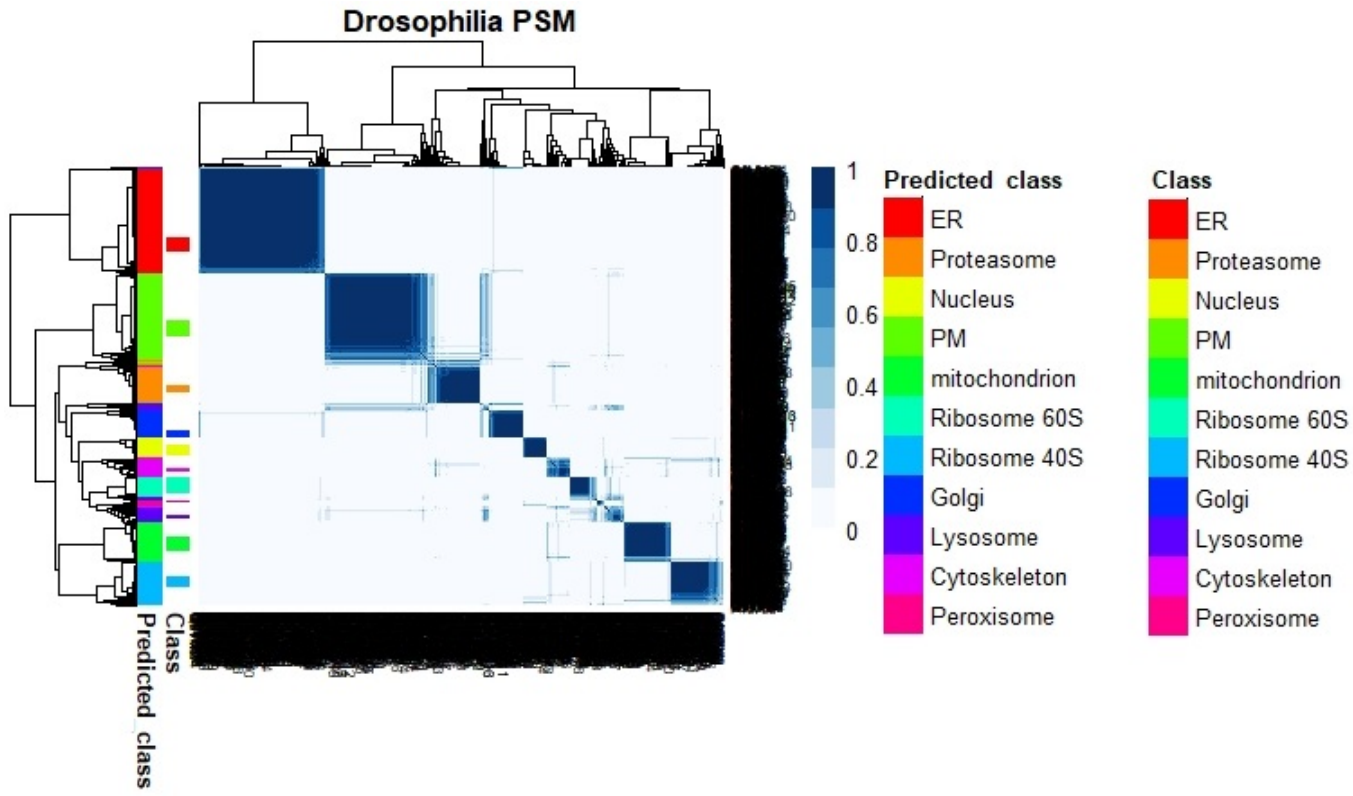


Figure 1: Posterior similarity matrix of data from Tan et al. [29].

To test the robustness of the tagmmdi model, we compare the improvement in prediction compared to the standalone TAGM model under the quadratic loss function. We compare the improvement in prediction from including the GO terms on data from *Arabidopsis*

[15], *Drosophila*[29] and mouse stem cells [6]. Within the known proteins we carry out cross validation, holding out 20% of the data as a validation set for 10 separate folds of 15,000 iterations with a burn-in of 5,000. We test for a difference in the variance and mean of the vector of resulting loss values for TAGM and tagmmdi and show them in figure 2. The results of the tests are shown in table 1.

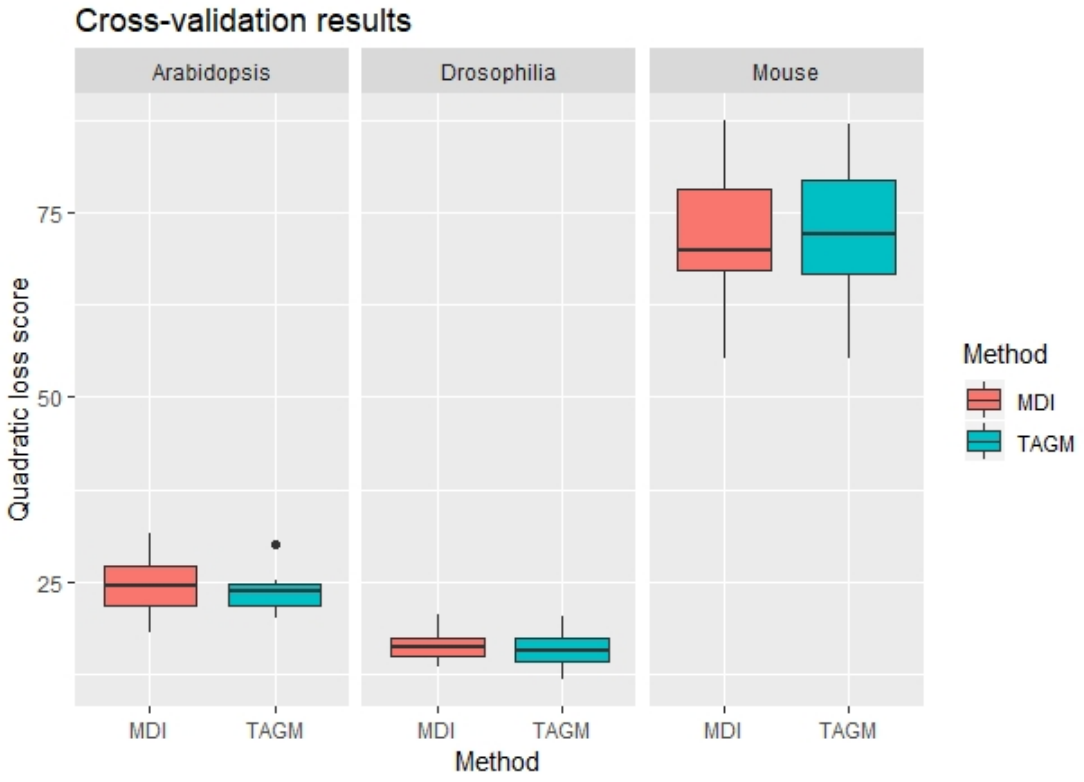


Figure 2: Boxplot of results of cross-validation.

| Organism | F-test | t-test |
|---|---|---|
| *Arabidopsis* | 0.378 | 0.572 |
| *Drosophila* | 0.761 | 0.573 |
| Mouse | 0.861 | 0.947 |

Table 1: Results of cross-validation showing $p$-values for tests for difference in mean and variance of loss value for folds for TAGM compared to tagmmdi.

Based on the lack of significant results we did not apply any multiple test correction [2] as it was deemed unnecessary.

All results were produced using the versions of software and hardware as shown in table 2.

| Object | Details |
|---|---|
| Model | LENOVO ideapad 110-17ACL |
| Processor | AMD A8-7410 APU with AMD Radeon R5 Graphics 2.20 GHz |
| Available RAM | 6.91GB |
| OS | Windows 10 Home v 1803 |
| R version | 3.5.1 |
| Rstudio version | 1.1.456 |
| Rcpp version | 0.12.18 |
| ARMA version | 9.100.5 |

Table 2: Description of relevant computer hardware and software.

# 3 Conclusions and further work

Compare to the vanilla TAGM model we see that tagmmdi does significantly differ in table 1. It is worth commenting that there is no disimprovement either, so if the computational cost is not a limit use of tagmmdi models will not offer any risk. We suggest that with a carefully chosen secondary dataset we could acquire information about our subclusters as categorical data is normally more interpretable than continuous data. We state this as the categories are normally human defined whereas continuous data, as a matrix of quantitative measurements, can be slightly dense. This applies less so in the supervised case investigated here, but certainly holds for many unsupervised cases and could reveal information about the subclusters even in the supervised case. We recommend testing the method on different auxiliary datasets such as Protein-Protein Interactions (PPI) to further test its capabilities.

The functions in *tagmmdi* also allow use of a second continuous dataset. This could be used to test the strength of association between replicates or to investigate for new organelles (perhaps by running the second dataset with an unsupervised mixture of Gaussians). The possibility for this later concept is investigated briefly in figure 3.

One can see that different organelles show strong sub-clusters. This suggests investigating this question and developing extensions to *tagmmdi* for this purpose.

Currently the R package *tagmmdi* is being updated by O. Crook to be integrated into the *Bioconductor* repository as part of the *pRoloc* suite.

Different MCMC methods such as a collapsed Gibbs sampler or Hamiltonian MCMC might lower the computational cost.

# A MDI model

For MDI in the case of $n$ observations in 2 datasets (also referred to as *contexts*):

$$p(\{c_{i1}, c_{i2}\}_{i=1}^{n}, v) \propto \gamma_{c_{i1}1}\gamma_{c_{i2}2}\left(1 + \phi\mathbb{I}(c_{i1} = c_{i2})\right) \tag{24}$$

We assume priors of $\gamma_{1l}, \ldots, \gamma_{nl} \overset{i.i.d}{\sim} Gamma(\alpha_l/K_l, 1)\ \forall\, l \in \{1, 2\}$ where $K_l$ is the number of clusters in the $l$th dataset. Similarly $\phi \sim Gamma(a, b)$.

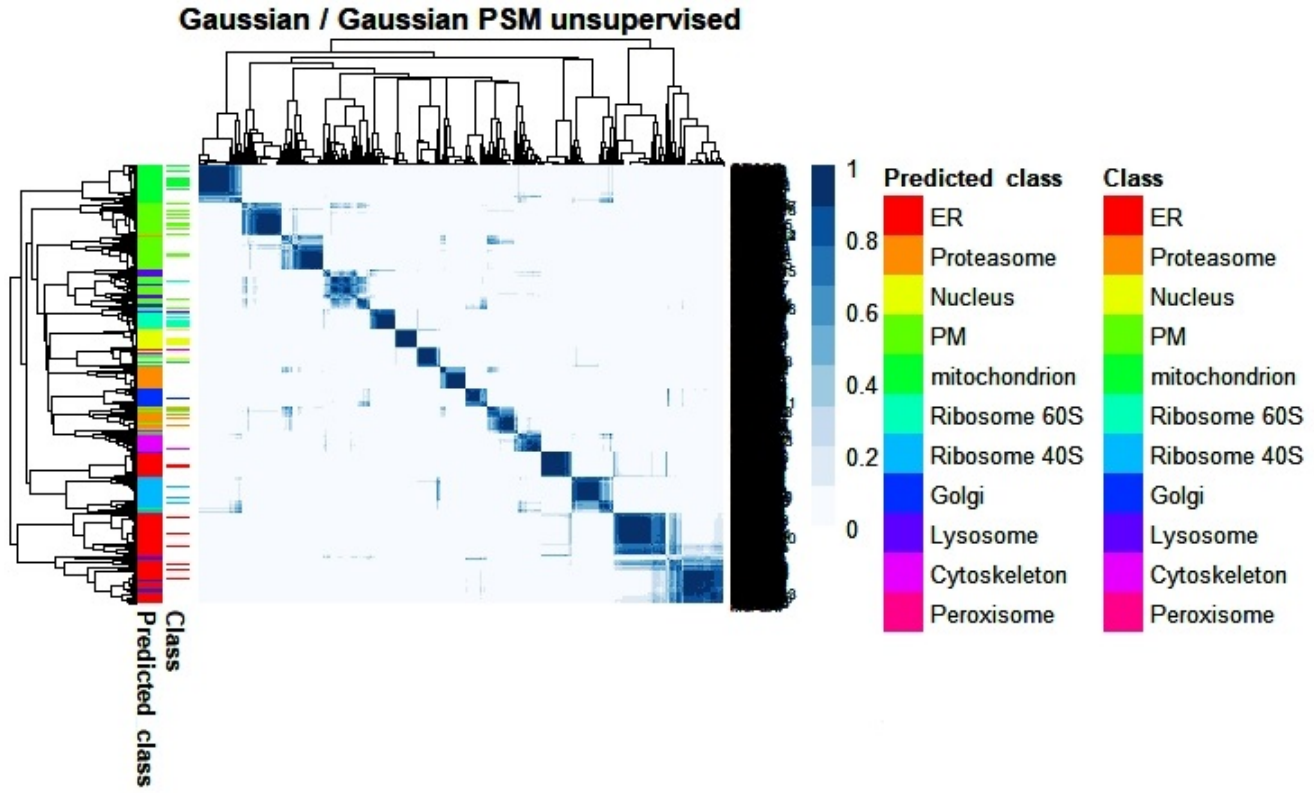


Figure 3: Posterior similarity matrix of data from Tan et al. [29] for one dataset with a TAGM model as in figure 1 and the second with an over-fitted Gaussian mixture model for which $K = 17$. The PSM is for the latter. Class predictions are from the TAGM model.

From (24) we calculate the normalising constant $Z$, and find:

$$Z = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \gamma_{c_{i1}1} \gamma_{c_{i2}2} \left(1 + \phi \mathbb{I}(k_1 = k_2)\right) \tag{25}$$

The joint density is hence:

$$p(\{c_{i1}, c_{i2}\}_{i=1}^{n} | \phi) = \frac{1}{Z} \prod_{i=1}^{n} \gamma_{c_{i1}1} \gamma_{c_{i2}2} \left(1 + \phi \mathbb{I}(c_{i1} = c_{i2})\right) \tag{26}$$

As in Nieto-Barajas et al. [25], we introduce a strategic latent variable $v$ such that the form is:

$$p(\{c_{i1}, c_{i2}\}_{i=1}^{n}, v) = \frac{v^{n-1} \exp(-vZ)}{(n-1)!} \prod_{i=1}^{n} \left( \left(1 + \phi \mathbb{I}(c_{i1} = c_{i2})\right) \prod_{l=1}^{2} \gamma_{c_{il}l} \right) \tag{27}$$

Where $Z$ remains as in (25).

## A.1 $v$

From (27) we can see the conditional for the strategic latent variable $v$ is:

$$p(v|\{c_{i1}, c_{i2}\}_{i=1}^{n}) \propto \frac{v^{n-1}\exp(-vZ)}{(n-1)!} \tag{28}$$

We recognise this as the pdf for a Gamma distribution with shape $n$ and rate $Z$, i.e. $v \sim Gamma(n, Z)$.

## A.2 $\gamma_{kl}$

If we expand $Z$ in (27) we can derive the conditional for the component weights, $\gamma_{kl}$:

$$p(\gamma_{kl}|\{c_{i1}, c_{i2}\}_{i=1}^{n}) \propto \exp\left(-v\gamma_{kl}\sum_{k_{l'}=1}^{K_{l'}}\gamma_{c_{il'}l'}\left(1+\phi\mathbb{I}(k_l=k_{l'})\right)\right)\prod_{i=1}^{n}\left(\left(1+\phi\mathbb{I}(c_{i1}=c_{i2})\right)\prod_{l=1}^{2}\gamma_{c_{il}l}\right)\mathbb{I}(c_{il}=k) \tag{29}$$

$$\propto \exp\left(-v\gamma_{kl}\sum_{k_{l'}=1}^{K_{l'}}\gamma_{c_{il'}l'}\left(1+\phi\mathbb{I}(k_l=k_{l'})\right)\right)\prod_{i=1}^{n}\left(1+\phi\mathbb{I}(c_{i1}=c_{i2})\right)\mathbb{I}(c_{il}=k)\gamma_{kl} \tag{30}$$

$$\propto \exp\left(-v\gamma_{kl}\sum_{k_{l'}=1}^{K_{l'}}\gamma_{c_{il'}l'}\left(1+\phi\mathbb{I}(k_l=k_{l'})\right)\right)\gamma_{kl}^{\sum_{i=1}^{n}\mathbb{I}(c_{il}=k)} \tag{31}$$

Where $l' \in \{1,2\} \wedge l' \neq l$. Thus $\gamma_{kl} \sim Gamma(a_\gamma, b_\gamma)$ where:

$$a_\gamma = 1 + \sum_{i=1}^{n}\mathbb{I}(c_{il}=k) \tag{32}$$

$$b_\gamma = v\sum_{k_{l'}=1}^{K_{l'}}\gamma_{c_{il'}l'}\left(1+\phi\mathbb{I}(k_l=k_{l'})\right) \tag{33}$$

## A.3 $\phi$ and associated derivations

### A.3.1 Conditional probability

Consider the conditional probability of $\phi$, then from (27) and expanding $Z$:

$$p(\phi|\{c_{i1}, c_{i2}\}_{i=1}^{n}, v) \propto \exp\left(-v\sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}\left(\left(1+\phi\mathbb{I}(j_1=j_2)\right)\prod_{l=1}^{2}\gamma_{k_l l}\right)\right)\prod_{i=1}^{n}\left(\left(1+\phi\mathbb{I}(c_{i1}=c_{i2})\right)\prod_{l=1}^{2}\gamma_{c_{il}l}\right) \tag{34}$$

We would like to transform this quantity into a density function we recognise. Consider the two occurrences of $\phi$:

$$a = \prod_{i=1}^{n} \left( (1 + \phi \mathbb{I}(c_{i1} = c_{i2})) \prod_{l=1}^{2} \gamma_{c_{il} l} \right) \tag{35}$$

$$b = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \left( (1 + \phi \mathbb{I}(k_1 = k_2)) \prod_{l=1}^{2} \gamma_{k_l l} \right) \tag{36}$$

Beginning with (35):

$$a = \prod_{i=1}^{n} \gamma_{c_{i1}1} \gamma_{c_{i2}2} \left(1 + \phi \mathbb{I}(c_{i1} = c_{i2})\right) \tag{37}$$

$$= \left(1 + \phi \mathbb{I}(c_{i1} = c_{i2})\right)^n \prod_{i=1}^{n} \gamma_{c_{i1}1} \gamma_{c_{i2}2} \tag{38}$$

$$\propto \left(1 + \phi \mathbb{I}(c_{i1} = c_{i2})\right)^n \tag{39}$$

$$= (1 + \phi))^{\sum_{i=1}^{n} \mathbb{I}(c_{i1} = c_{i2})} \tag{40}$$

$$= \sum_{r=0}^{\sum_{i=1}^{n} \mathbb{I}(c_{i1} = c_{i2})} \binom{\sum_{i=1}^{n} \mathbb{I}(c_{i1} = c_{i2})}{r} \phi^r \qquad \text{(from the binomial theorem)} \tag{41}$$

Here $\sum_{i=1}^{n} \mathbb{I}(c_{i1} = c_{i2})$ is the count of observations assigned to the same cluster in both contexts and will be called $c$.

Now consider (36):

$$b = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \left( (1 + \phi \mathbb{I}(k_1 = k_2)) \prod_{l=1}^{2} \gamma_{k_l l} \right) \tag{42}$$

We see that for our conditional we can ignore all cases when $k_1 \neq k_2$ as $\phi$ is not present in these. Letting $K_{min} = \min(K_1, K_2)$, (42) simplifies to:

$$b \propto \sum_{k=1}^{K_{min}} \gamma_{k1} \gamma_{k2} \left(1 + \phi\right) \tag{43}$$

$$\propto \sum_{k=1}^{K_{min}} \gamma_{k1} \gamma_{k2} \phi \tag{44}$$

Thus updating (34) accordingly gives us:

$$p(\phi | \{c_{i1}, c_{i2}\}_{i=1}^{n}, v) \propto \exp\left( -v \sum_{k=1}^{K_{min}} \gamma_{k1} \gamma_{k2} \phi \right) \sum_{r=0}^{c} \binom{c}{r} \phi^r \tag{45}$$

We notice this has the structure similar to a mixture of Gamma distributions with each distribution having a shape of $r - 1$ and a rate of $-v \sum_{k=1}^{K_{min}} \gamma_{k1} \gamma_{k2}$. We thus multiply the equation by 1 to have:

$$p(\{c_{i1}, c_{i2}\}_{i=1}^{n}, v|\phi) \propto \sum_{r=0}^{c} \binom{c}{r} \frac{r!}{\left(v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\right)^{r+1}} \frac{\left(v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\right)^{r+1}}{r!} \phi^r \exp\left(-v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\phi\right) \tag{46}$$

$$= \sum_{r=0}^{c} \binom{c}{r} \frac{r!}{\left(v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\right)^{r+1}} Gamma\left(r+1, v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\right) \tag{47}$$

As we know that $p(\phi|\{c_{i1}, c_{i2}\}_{i=1}^{n}, v)$ must integrate over $\phi$ to 1, we know the normalising constant must be the sum of the integrals of the Gamma distributions, i.e.:

$$Z_\phi = \sum_{r=0}^{c} \binom{c}{r} \frac{r!}{\left(v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\right)^{r+1}} \int \frac{\left(v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\right)^{r+1}}{r!} \phi^r \exp\left(-v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\phi\right) d\phi \tag{48}$$

$$= \sum_{r=0}^{c} \binom{c}{r} \frac{r!}{\left(v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\right)^{r+1}} \tag{49}$$

Combining these gives:

$$p(\phi|\{c_{i1}, c_{i2}\}_{i=1}^{n}, v) = \frac{1}{Z_\phi} \sum_{r=0}^{c} \binom{c}{r} \frac{r!}{\left(v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\right)^{r+1}} Gamma\left(r+1, v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\right) \tag{50}$$

### A.3.2 Posterior distribution

Now if we consider a prior of $Gamma(a_0, b_0)$ on the $\phi$, we have a prior probability of:

$$p(\phi) = \frac{b_0^{a_0}}{(a_0-1)!} \phi^{a_0-1} \exp(-b_0\phi) \tag{51}$$

Thus our posterior conditional is:

$$p(\phi|\cdot) \propto p(\phi) p(\{c_{i1}, c_{i2}\}_{i=1}^{n}, v|\phi) \tag{52}$$

$$\propto \frac{b_0^{a_0}}{(a_0-1)!} \phi^{a_0-1} \exp(-b_0\phi) \sum_{r=0}^{c} \binom{c}{r} \phi^r \exp\left(-v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}\phi\right) \tag{53}$$

$$\propto \sum_{r=0}^{c} \binom{c}{r} \phi^{r+a_0-1} \exp\left(\left(-v \sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2} - b_0\right)\phi\right) \tag{54}$$

We now use the same trick as with the likelihood to move towards the shape of a Gamma density function, and multiply by 1:

$$p(\phi|\cdot) \propto \sum_{r=0}^{c} \binom{c}{r} \frac{(r+a_0-1)!}{\left(v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right)^{r+a_0}} \frac{\left(v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right)^{r+a_0}}{(r+a_0-1)!} \phi^{r+a_0-1} \exp\left(-\left(v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right)\phi\right) \tag{55}$$

$$= \sum_{r=0}^{c} \binom{c}{r} \frac{(r+a_0-1)!}{\left(v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right)^{r+a_0}} Gamma\left(r+a_0, v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right) \tag{56}$$

Again we have a mixture of Gamma densities. For the normalising constant, we have, similarly to (49):

$$Z'_\phi = \sum_{r=0}^{c} \binom{c}{r} \frac{(r+a_0-1)!}{\left(v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right)^{r+a_0}} \int \frac{\left(v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right)^{r+a_0}}{(r+a_0-1)!} \phi^{r+a_0-1} \exp\left(-\left(v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right)\phi\right) d$$

(57)

$$= \sum_{r=0}^{c} \binom{c}{r} \frac{(r+a_0-1)!}{\left(v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right)^{r+a_0}} \tag{58}$$

Thus the posterior distribution of the context similarity parameter $\phi$ is:

$$p(\phi|\{c_{i1}, c_{i2}\}_{i=1}^{n}, v) = \frac{1}{Z'_\phi} \sum_{r=0}^{c} \binom{c}{r} \frac{(r+a_0-1)!}{\left(v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right)^{r+a_0}} Gamma\left(r+a_0, v\sum_{k=1}^{K_{min}} \gamma_{k1}\gamma_{k2}+b_0\right) \tag{59}$$

## Acknowledgments

Thanks are due to my supervisors Paul Kirk (primary) and Laurent Gatto (secondary) as well as their PhD student Oliver M Crook who was very free with his time in explaining concepts to me. The project was designed and overseen by Paul who also took the time to run through my code on a few occasions which is an intimidating task for the unwary. All the people at the MRC Biostatistics Unit, Cambridge University made my stay there very pleasant as well as educational.

Thanks also to Aalt-Jan van Dijk for agreeing to supervise me from Wagenignen thereby enabling this project.

With regards to software this project was massively enabled by RStudio [30] and GitHub. Eddelbuettel and François [10] deserve special mention for the development of Rcpp which made integrating the Gibbs sampler from C++ far easier. The C++ is almost entirely coded using objects and functions from Armadillo [26].

The advice from Bryan [5] was much appreciated and saved an awful lot of time.

## 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. doi: 10.1038/75556.

[2] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57:289–300, 1995.

[3] Lisa M. Breckels, Sean B. Holden, David Wojnar, Claire M. Mulvey, Andy Christoforou, Arnoud Groen, Matthew W. B. Trotter, Oliver Kohlbacher, Kathryn S. Lilley, and Laurent Gatto. Learning from Heterogeneous Data Sources: An Application in Spatial Proteomics. *PLOS Computational Biology*, 12(5):e1004920, May 2016. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004920.

[4] Lisa Marie Breckels, Claire Mairead Mulvey, Kathryn Susan Lilley, and Laurent Gatto. A bioconductor workflow for processing and analysing spatial proteomics data, 2016.

[5] Jennifer Bryan. Excuse me, do you have a moment to talk about version control? doi: 10.7287/peerj.preprints.3159v2.

[6] Andy Christoforou, Claire M. Mulvey, Lisa M. Breckels, Aikaterini Geladaki, Tracey Hurrell, Penelope C. Hayward, Thomas Naake, Laurent Gatto, Rosa Viner, Alfonso Martinez Arias, and Kathryn S. Lilley. A draft map of the mouse pluripotent stem cell spatial proteome. *Nature Communications*, 7(1), December 2016. ISSN 2041-1723. doi: 10.1038/ncomms9992.

[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] Oliver M Crook, Claire M Mulvey, Paul D. W. Kirk, Kathryn S Lilley, and Laurent Gatto. A Bayesian Mixture Modelling Approach For Spatial Proteomics. *PLOS Computational Biology*, 14(11), November 2018. doi: 10.1101/282269.

[9] T. P. J. Dunkley, R. Watson, J. L. Griffin, P. Dupree, and K. S. Lilley. Localization of Organelle Proteins by Isotope Tagging (LOPIT). *Molecular & Cellular Proteomics*, 3(11):1128–1134, November 2004. ISSN 1535-9476, 1535-9484. doi: 10.1074/mcp.T400009-MCP200.

[10] Dirk Eddelbuettel and Romain François. **Rcpp** : Seamless *R* and *C++* Integration. *Journal of Statistical Software*, 40(8), 2011. ISSN 1548-7660. doi: 10.18637/jss.v040.i08.

[11] Aikaterini Geladaki, Nina Kocevar Britovsek, Lisa M. Breckels, Tom S. Smith, Claire M. Mulvey, Oliver M. Crook, Laurent Gatto, and Kathryn S. Lilley. LOPIT-DC: A simpler approach to high-resolution spatial proteomics. July 2018. doi: 10.1101/378364.

[12] 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.

[13] Gene Ontology Consortium. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32(90001):258D–261, January 2004. ISSN 1362-4962. doi: 10.1093/nar/gkh036.

[14] Toby J. Gibson. Cell regulation: Determined to signal discrete cooperation. *Trends in Biochemical Sciences*, 34(10):471–482, October 2009. ISSN 09680004. doi: 10.1016/j.tibs.2009.06.007.

[15] Arnoud J. Groen, Gloria Sancho-Andrés, Lisa M. Breckels, Laurent Gatto, Fernando Aniento, and Kathryn S. Lilley. Identification of Trans-Golgi Network Proteins in *Arabidopsis thaliana* Root Tissue. *Journal of Proteome Research*, 13(2):763–776, February 2014. ISSN 1535-3893, 1535-3907. doi: 10.1021/pr4008464.

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

[17] Richard P Horgan and Louise C Kenny. 'Omic' technologies: Genomics, transcriptomics, proteomics and metabolomics: The Obstetrician & Gynaecologist. *The Obstetrician & Gynaecologist*, 13(3):189–195, July 2011. ISSN 14672561. doi: 10.1576/toag.13.3.189.27672.

[18] M.-C. Hung and W. Link. Protein localization in disease and therapy. *Journal of Cell Science*, 124(20):3381–3392, October 2011. ISSN 0021-9533, 1477-9137. doi: 10.1242/jcs.089110.

[19] Tweeny R. Kau, Jeffrey C. Way, and Pamela A. Silver. Nuclear transport and cancer: From mechanism to intervention. *Nature Reviews Cancer*, 4:106, February 2004.

[20] 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.

[21] Jun S Liu. The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem. *Journal of the American Statistical Association*, 89(427):958–966, September 1994.

[22] David J C MacKay. *Information Theory, Inference, and Learning Algorithms*. September 2003.

[23] 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.

[24] 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.

[25] Luis E. Nieto-Barajas, Igor Prünster, and Stephen G. Walker. Normalized random measures driven by increasing additive processes. *The Annals of Statistics*, 32(6):2343–2360, December 2004. ISSN 0090-5364. doi: 10.1214/009053604000000625.

[26] Conrad Sanderson and Ryan Curtin. Armadillo: A template-based C++ library for linear algebra. *The Journal of Open Source Software*, 1(2):26, June 2016. ISSN 2475-9066. doi: 10.21105/joss.00026.

[27] 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.

[28] Jacqueline E. Siljee, Yi Wang, Adelaide A. Bernard, Baran A. Ersoy, Sumei Zhang, Aaron Marley, Mark Von Zastrow, Jeremy F. Reiter, and Christian Vaisse. Subcellular localization of MC4R with ADCY3 at neuronal primary cilia underlies a common pathway for genetic predisposition to obesity. *Nature Genetics*, 50(2):180–185, February 2018. ISSN 1061-4036, 1546-1718. doi: 10.1038/s41588-017-0020-9.

[29] Denise J. L. Tan, Heidi Dvinge, Andrew Christoforou, Paul Bertone, Alfonso Martinez Arias, and Kathryn S. Lilley. Mapping Organelle Proteins and Protein Complexes in *Drosophila melanogaster*. *Journal of Proteome Research*, 8(6):2667–2678, June 2009. ISSN 1535-3893, 1535-3907. doi: 10.1021/pr800866n.

[30] RStudio Team. RStudio: Integrated Development Environment for R. RStudio, Inc., 2016.