POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Project Report

Scalable Bayesian Image Segmentation of Brain Spectroscopy Data

Bayesian Statistics

Authors: Simone Colombara, Alessia Cotroneo, Francesco De Caro, Riccardo Morandi, Chiara Schembri and Alfredo G. Zapiola

Tutors: Dr. Denti Francesco, Dr. Capitoli Giulia

Academic year: 2022-2023

Abstract

Image segmentation aims at grouping similar pixels of an image under a common class label and can be therefore interpreted as a clustering problem. In this project we use the hidden Potts model, which describes the distribution of pixels in an image via a Bayesian mixture model, while accounting for the spatial dependence between adjacent pixels. The spatial dependence is introduced via hidden Markov random fields – in particular, a Gibbs random field.

We implement a Gibbs sampler for the hidden Potts model capable of dealing with images of arbitrary shape and number of channels.

We fit the model to real brain spectroscopy data, a challenging dataset due to the high dimensionality. Our goal is to integrate multiple molecular spectra to extract meaningful insights – and possibly anomalies – from these real biological images.

The source code related to this project is available at our GitHub repository:
`https://github.com/alfredo-g-zapiola/multiPotts`

Contents

1. Dataset

1.1. Mass Spectroscopy Imaging

Mass spectrometry imaging (MSI) is an emerging technology capable of mapping various biomolecules within their native spatial context. This work describes an application to spatial multiomics data obtained via the state-of-the-art MALDI-MSI technology. MALDI-MSI uses a laser energy-absorbing matrix to create ions from large molecules with minimal fragmentation. In other words, this technology visualizes the distribution of molecules such as peptides, lipids, and glycans in a biological sample [Rohner et al., 2005]. Here, we consider the sequential MALDI-MSI imaging of lipids, glycans, and tryptic peptides on a single section of mouse brain tissue. For this work, a slice from the mouse's brain is partitioned into a grid of pixels. Then, MALDI-MSI acquires a mass spectrum for each pixel. In particular, MALDI-MSI creates a grid of pixel-by-pixel spectra. Each spectra measures the mass-to-charge (m/z) values representing analytes of interest along the x-axis and the corresponding abundance along the y-axis [Boggio et al., 2011].
Finally, a three-dimensional MSI dataset is obtained by arranging the mass spectra and the grid coordinates for each pixel to be investigated in further analysis. The full workflow is depicted in Figure 1, from [Aichler, 2015]. Due to experimental limitations, the MALDI-MSI spectra might contain noise affecting the statistical analysis. To mitigate this issue, a well-established biological pipeline is performed.

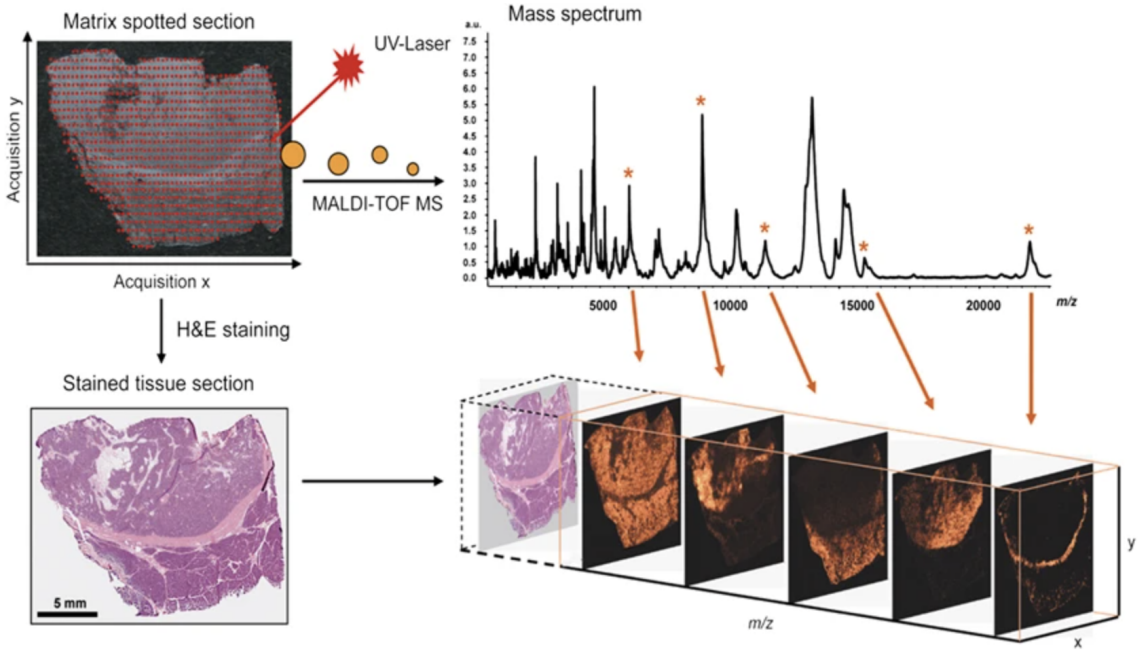

Figure 1: MALDI-MSI imaging workflow and spectrum, from Aichler [2015].

More precisely, the raw dataset produced by the MALDI-MSI technology undergoes some initial preprocessing steps to filter out the noise. The steps are: baseline correction, smoothing, normalization, spectra alignment, peak detection and extraction. In addition, the preprocessing needs to reduce the intra-sample variability and correct for analytical and instrumental variability following sample preparation to ensure accurate m/z localization. The resulting dataset for our statistical analysis has the following characteristics: the image of the biological tissue is analyzed with a raster of 50 μm resulting in a total of about 18,000 laser shots (defining the pixels). In other words, we obtain thousands of spectra, each with its pixel coordinates (x, y).
One of the main challenges we faced working with this dataset was the abundance of missing values. Indeed, whenever the incoming beam of molecules falls below the threshold of the instrument, a missing value is recorded. In our dataset the missing values were around 40% to 50%. Guided by biological information, we interpret the lacking as indicating either absence or presence below detection threshold of a certain molecule. Therefore, we decided to replace all of the missing values with zeros.
We worked on three distinct datasets, all relating to the same tissue section, each focusing on a specific range of m/z values and therefore a specific macro-group of molecules: lipids, glycans, and peptides. We later combined the measurements coming from each of the three datasets to improve the performance of our models.

1.2. Statistical preprocessing: dimensionality reduction

Recall that, for each pixel, the abundance of each molecule is recorded as a function of the molecular masses (m/z) value. This procedure leads to the measurement of potentially hundreds of variables. The correlation between different m/z is quite high. We therefore decided to perform dimensionality reduction, separately on each dataset.

Given the functional nature of the data - a spectrum for each pixel location is recorded - we perform pixel-wise functional principal component analysis (fPCA) [Ramsay and Silverman, 2005]. In order for our dimensionality reduction to be meaningful, we need to take into account the nature of our data (recording intensities). To do so, we constrained each component to be positive, both when dealing with smoothed functions and with the actual fPC curves. Unfortunately, the application of this constraint, combined with the large number of zeros, sensibly complicates the preprocessing task. In our experience, we observed that a large number of zeros pushes the curve below the x-axis when smoothing of the spectra is performed. This issue hinders the standard trick of representing a positive function as the exponential of an unconstrained function.

The best performing solution we found was the one using no smoothing and a number of second order b-spline basis equal to the number of distinct m/z values in the dataset.

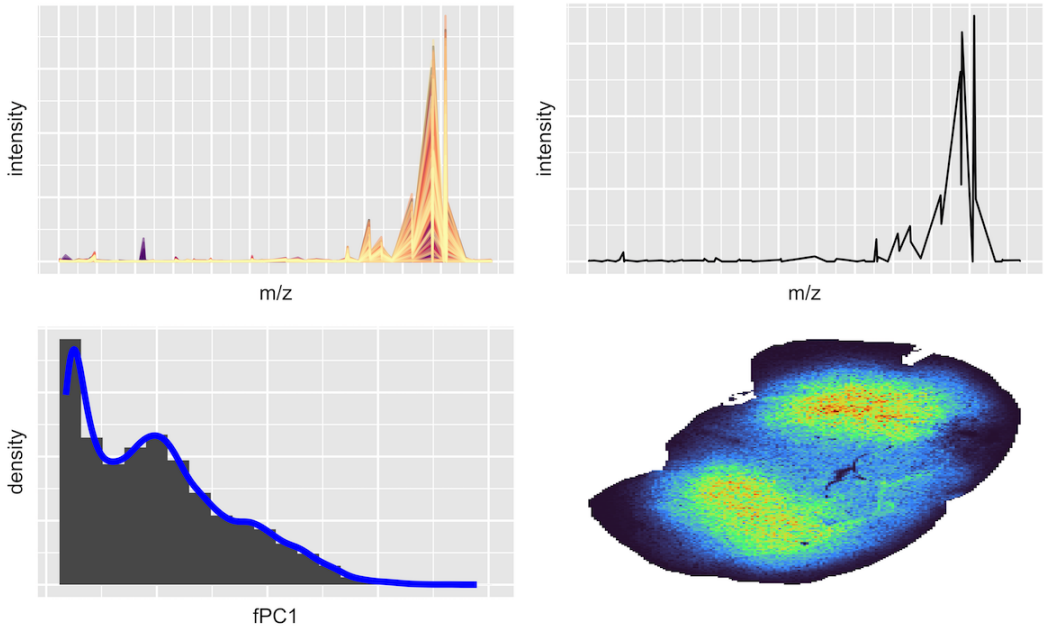

Figure 2: Lipids dataset. Top left panel: example of m/z intensities for a generic subsample of pixels. Top right panel: corresponding fPC curve. Bottom left panel: distribution of the first fPC scores. Bottom right panel: spatial distribution of the first fPC scores.

Figure 2 shows the results of the dimensionality reduction on the lipids dataset. In particular, we can see that the distribution of the fPC scores suggests the use of a Gaussian mixture to segment the picture, and that the principal component found reflects closely the nature of the data. Furthermore, in the spatial distribution of the scores, we can see that the patterns in the data were captured and will be present in the reduced data that we will use for clustering.

Dataset	component 1	component 2	component 3	component 2	component 5
Lipids	0.9441	0.9694	0.9820	0.9907	0.9942
Peptides	0.5894	0.8094	0.9112	0.9612	0.9726
Glycans	0.8385	0.8768	0.9033	0.9196	0.9333

Table 1: Cumulative explained variance using fPCA.

A similar procedure was carried out on the other datasets with comparable results. We want to highlight that, for all of the datasets, the vast majority of the variance is explained by just a few principal components, as shown in Table 1.
The patterns of the first principal component, and their relative distribution, can be seen in Figure 3

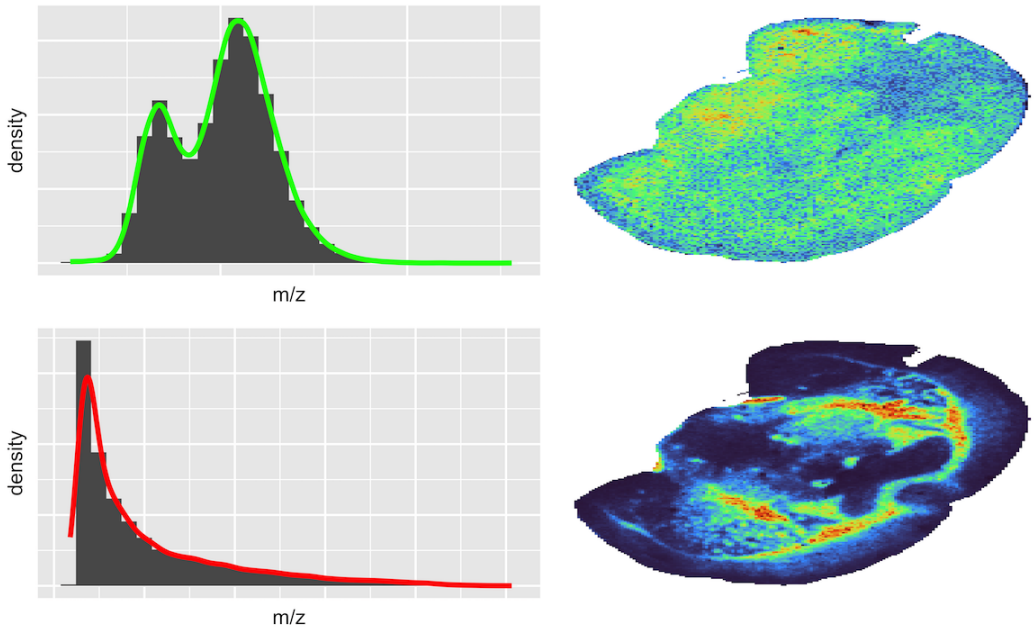

Figure 3: Kernel density estimate and spatial distribution of the first fPC scores for: Top row: glycans data. Bottom row: peptides data.

We would like to underscore that, during the imaging process, the specimen deteriorates, creating two main problems: the first is the fact that there are some pixels for which we do not have measurements for all of the different molecular types, particularly around the edges of the sample, and we therefore had to manually match all of the observation in the different datasets to assemble the composite data; the second is the presence of peculiar patterns for some m/z values, as one can see in Figure 4.

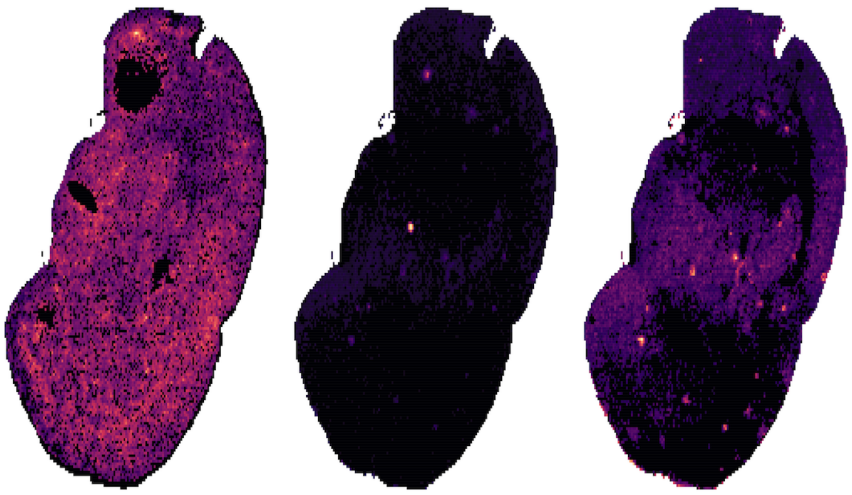

Figure 4: Examples of peculiar patterns found in the data when considering the spatial distribution for a fixed value of m/z equal to, respectively: 129, 1044 and 839.

2. Models

To perform model-based clustering, we adopt two models: the standard Gaussian mixture model (GMM), which does not take into account the spatial correlation between neighbouring pixels, and the Hidden Potts Model (HPM), which uses a hidden Markov random field to account for such spatial dependencies. In both cases, our analysis starts with the one-dimensional approach (taking into account just the first fPC score for the three datasets). Then, the analysis is extended in a multi-dimensional clustering task, taking into account either more than one component for each dataset or combining the three first fPC of each dataset in a single model.
The Gibbs sampler used for the univariate models was already implemented in the `bayesImageS` R-package [Moores et al., 2021]. Mimicking the efficient implementation of `bayesImageS`, we then coded an efficient multivariate Gibbs sampler for both GMM and HPM using `RcppArmadillo` [Eddelbuettel and Sanderson, 2014]. Our customed code is capable of dealing with images of arbitrary shape, as in the case of brain of different morphology, and with an arbitrary number of channels.

All of the methods below were applied to the three datasets and, in the case of multivariate models, to the combined dataset. For the sake of conciseness, we will show the results for a selected subsample of analyses. To see all the numerical results, please refer to the GitHub repository associated with this project.

2.1. Cluster estimation

When performing Bayesian inference, it is necessary to summarize the posterior distribution in order to convey meaningful results. This is particularly important when we want to summarize the posterior of parameters with complicated structures. With clustering, we are interested in estimating a partition ρ of $\{1, \dots, n\}$. A partition can be represented using cluster labels: we say that items i and j belong to the same cluster if and only if their cluster labels c_i and c_j are equal. In all the models in this project, we represented the partition using cluster allocation labels.
The canonical approach is to introduce a loss function and to choose as estimator the one minimizing such loss function:

$$\boldsymbol{c}^* = \underset{\hat{c}}{\operatorname{argmin}}\, \mathbb{E}[\ell(c, \hat{c}) \mid \mathcal{D}], \tag{1}$$

where $\mathcal{D}$ represents the data. With the exception of trivial cases, the expectation in (1) must be computed using the posterior MCMC samples $\{c^{(h)}\}_{h=1}^{H}$:

$$\mathbb{E}[\ell(c, \hat{c}) \mid \mathcal{D}] \approx \frac{1}{H} \sum_{h=1}^{H} \ell(c^{(h)}, \hat{c}). \tag{2}$$

One of the most commonly used loss functions in this setting is Binder's loss [Binder, 1978]:

$$\ell(c, \hat{c}) = \sum_{i<j} (a \mathbb{1}_{\{c_i = c_j\}} \mathbb{1}_{\{\hat{c}_i \neq \hat{c}_j\}} + b \mathbb{1}_{\{c_i \neq c_j\}} \mathbb{1}_{\{\hat{c}_i = \hat{c}_j\}}), \tag{3}$$

where $a > 0$ and $b > 0$ give the unit costs for pairwise misclassification. Specifically, a represents the cost of failing to cluster together two items which should be clustered together, whereas b represents the cost of clustering together two items which should be separate. In this project we assumed $a = b = 1$. Lau and Green [2007] noted that minimizing the posterior expectation of Binder's loss is equivalent to maximizing the following:

$$f(\hat{c}) = \sum_{i<j} \mathbb{1}_{\{\hat{c}_i = \hat{c}_j\}} \left(\pi_{i,j} - \frac{b}{a+b} \right), \tag{4}$$

where π is the posterior similarity matrix, an n-by-n matrix with elements:

$$\pi_{i,j} = \mathbb{P}(c_i = c_j \mid \mathcal{D}) \approx \hat{\pi}_{i,j} = \frac{1}{H} \sum_{h=1}^{H} \mathbb{1}_{\{c_i^{(h)} = c_j^{(h)}\}}. \tag{5}$$

This optimization problem can be addressed in multiple ways. For example, it can be viewed as an agglomerative hierarchical clustering problem [Fritsch and Ickstadt, 2009] or as a binary integer programming [Lau and Green, 2007]. Both these methods scale quite poorly with the number of observations n. More recent and efficient methods are based on greedy search algorithms. These approaches take small, locally-optimal updates at each step in an attempt to find the globally optimal solution. One downside to this method is its dependence on

the initial partition, especially since the algorithm can get stuck in a local minimum. In our project, given the high dimensionality of the dataset, we opt for this method. In particular we rely on the `R` package `SALSO` [Dahl et al., 2022] to minimize Binder's loss.

2.2. Univariate Gaussian Mixture Model

As our first Bayesian model, we fit a GMM using the first fPC score to obtain a benchmark result.

$$\begin{aligned} y_1, \dots, y_n \mid \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 &\overset{\text{iid}}{\sim} \sum_{k=1}^{K} w_k \mathcal{N}(\mu_k, \sigma_k^2), \\ \boldsymbol{w} &\sim \mathit{Dirichlet}(\boldsymbol{\lambda}), \\ \mu_k &\sim \mathcal{N}(m_{k,0}, s_{k,0}^2), \qquad k = 1, \dots, K, \\ \sigma_k^2 &\sim \mathit{Inv-Gamma}\left(\frac{n_k}{2}, \frac{n_k v_k}{2}\right) \quad k = 1, \dots, K. \end{aligned} \tag{6}$$

Since the goal is to apply the mixture model in order to find clusters, we make this connection to clustering more explicit by introducing the "cluster allocation" variables z_i, for $i = 1, \dots, n$, indicating the cluster (and, therefore, the mixture component pixel i belongs to).
Introducing this auxiliary variables, Model (6) becomes:

$$\begin{aligned} y_i \mid z_i = k, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 &\overset{\text{iid}}{\sim} \mathcal{N}(\mu_k, \sigma_k^2), \\ z_i \mid \boldsymbol{w} &\sim \mathit{Categorical}(\boldsymbol{w}), \\ \mu_k &\sim \mathcal{N}(m_{k,0}, s_{k,0}^2), \\ \sigma_k^2 &\sim \mathit{Inv-Gamma}\left(\frac{n_k}{2}, \frac{n_k v_k}{2}\right), \\ \boldsymbol{w} &\sim \mathit{Dirichlet}(\boldsymbol{\lambda}), \end{aligned} \tag{7}$$

In order to sample from the posterior distributions we used a Gibbs sampler, whose full conditionals are:

$$\begin{aligned} \mu_k \mid \boldsymbol{\sigma}^2, \boldsymbol{y}, \boldsymbol{z} &\sim \mathcal{N}(m_{k,p}, s_{k,p}^2), \\ \sigma_k^2 \mid \boldsymbol{\mu}, \boldsymbol{y}, \boldsymbol{z} &\sim \mathit{Inv-Gamma}\left(\frac{n_k}{2} + \frac{N_k}{2}, \frac{n_k v_k + \sum_{i:z_i=k}(y_i - \mu_k)^2}{2}\right), \\ \boldsymbol{w} \mid \boldsymbol{z} &\sim \mathit{Dirichlet}(\lambda_1 + N_1, \dots, \lambda_K + N_K), \\ z_i \mid \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{y} &\sim \mathit{Categorical}\left(\frac{w_1 \phi(y_i \mid \mu_1, \sigma_1^2)}{\sum_{k=1}^{K} w_k \phi(y_i \mid \mu_k, \sigma_k^2)}, \dots, \frac{w_K \phi(y_i \mid \mu_K, \sigma_K^2)}{\sum_{k=1}^{K} w_k \phi(y_i \mid \mu_k, \sigma_k^2)}\right), \\ m_{k,p} &= \frac{N_k s_k^2 \overline{y}_{n,k} + m_k \sigma_k^2}{N_k s_k^2 + m_j \sigma_k^2} \qquad s_{k,p}^2 = \frac{n_k v_k + \sum_{i:z_i=k}(y_i - \mu_k)^2}{2}, \\ N_k &= \sum_{i=k}^{N} \mathbb{1}_{\{z_i = k\}}, \qquad \overline{y}_{n,k} = \frac{\sum_{i:z_i=k} y_i}{N_k}. \end{aligned} \tag{8}$$

In equation (8), $\phi(x \mid \mu, \sigma^2)$ represents the probability density function of a normal distribution with mean μ and variance σ^2 evaluated in x.

2.2.1. Gaussian Mixture Model results on peptides

Here, we report an example of the GMM applied to the first fPC of the peptides dataset, in which case the best clustering is obtained with $K = 3$. In Figure 5 we can clearly see that even with the simplest model we are able to perform a satisfactory clustering. On the one hand, we can capture the complex pattern pattern of the fPC and clearly distinguish different clusters. On the other hand we aim to detect sharper cluster edges to reduce the number of isolated pixels. In order to improve the clustering we have multiple options:

- incorporating more than one fPC to introduce more information in the model (Section 2.3);
- introducing a spatial relation using the Potts model(Section 2.4);
- combining both options in a multidimensional Potts model (Section 2.5)

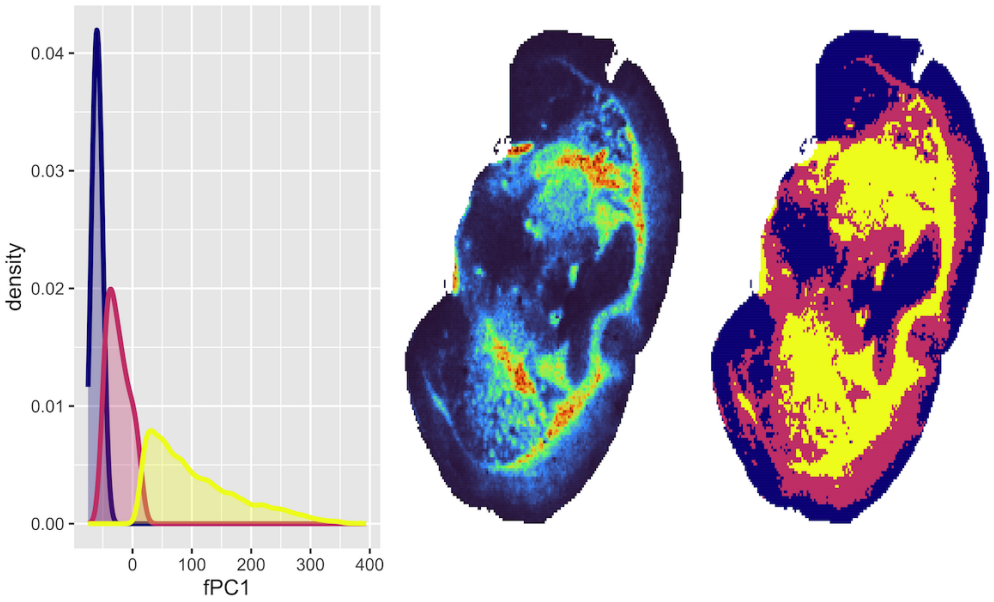

Figure 5: Unidimensional GMM on the peptides dataset. Left panel: kernel density estimates of the first fPC of each cluster. Center panel: fPC distribution in the brain. Right panel: obtained clustering.

2.3. Multivariate Gaussian Mixture Model

To investigate if the clustering can be improved by taking into account more than one principal component, we want to focus on multivariate likelihoods. As the first step, we extended the previous model to a multidimensional GMM:

$$\begin{aligned} \boldsymbol{y_i} \mid z_i = k, \boldsymbol{\mu}, \boldsymbol{\Sigma} &\overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu_k}, \Sigma_k), \\ z_i \mid \boldsymbol{w} &\sim Categorical(\boldsymbol{w}), \\ \boldsymbol{\mu}_k &\sim \mathcal{N}(\boldsymbol{b}_{k,0}, B_{k,0}), \\ \Sigma_k &\sim Inv-Wishart(V_{k,0}, n_{k,0}), \\ \boldsymbol{w} &\sim Dirichlet(\boldsymbol{\lambda}). \end{aligned} \tag{9}$$

Once again, in order to sample from the posterior distributions we used a Gibbs sampler. The full conditionals of the model are reported in equation (10), our implementation of this Gibbs sampler is available on GitHub in the `GibbsGMM` function.

$$\begin{aligned} \boldsymbol{\mu_k} \mid \boldsymbol{\Sigma}, \boldsymbol{y}, \boldsymbol{z} &\sim \mathcal{N}(\boldsymbol{b_{k,p}}, B_{k,p}), \\ \Sigma_k \mid \boldsymbol{\mu}, \boldsymbol{y}, \boldsymbol{z} &\sim Inv-Wishart(V_{k,p}, n_{k,p}), \\ \boldsymbol{w} \mid \boldsymbol{z} &\sim Dirichlet(\lambda_1 + N_1, \dots, \lambda_K + N_K), \\ z_i \mid \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{y}, &\sim Cat\left(\frac{w_1 \phi(\boldsymbol{y_i} \mid \boldsymbol{\mu_1}, \Sigma_1)}{\sum_{k=1}^K w_k \phi(\boldsymbol{y_i} \mid \boldsymbol{\mu_k}, \Sigma_k)}, \dots, \frac{w_K \phi(\boldsymbol{y_i} \mid \boldsymbol{\mu_K}, \Sigma_K)}{\sum_{k=1}^K w_k \phi(\boldsymbol{y_i} \mid \boldsymbol{\mu_k}, \Sigma_k)}\right), \\ B_{k,p} &= (N_k \Sigma_k^{-1} + B_{k,0}^{-1})^{-1}, \\ \boldsymbol{b}_{k,p} &= B_{k,p}(N_k \Sigma_k^{-1} \overline{\boldsymbol{y}}_{\boldsymbol{k}} + B_{k,0}^{-1} \boldsymbol{b}_{k,0}), \\ N_k = \sum_{i=k}^{N} \mathbb{1}_{\{z_i = k\}}, &\qquad \overline{\boldsymbol{y}}_{\boldsymbol{k}} = \frac{\sum_{i:z_i=k} \boldsymbol{y_i}}{N_k}, \\ V_{k,p} &= \left(V_{k,0}^{-1} + \sum_{i:z_i=k} (\boldsymbol{y}_i - \boldsymbol{\mu}_k)(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T\right)^{-1}, \\ n_{k,p} &= n_{k,0} + N_k. \end{aligned} \tag{10}$$

In equation (10), $\phi(\boldsymbol{x} \mid \boldsymbol{\mu}, \Sigma)$ represents the probability density function of a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ evaluated in $\boldsymbol{x}$.

2.3.1. Multivariate Gaussian Mixture Model results on combined datasets

When dealing with multidimensional models, we can follow two different routs: either we can (i) jointly model more than one fPC for each dataset or we can (ii) jointly consider one component for each of the three different datasets. In our experiments we use 3 components for the peptides, 4 for the glyacans and 2 for the lipids; for the combined dataset we use a single component per molecular type. As you can see in the left-most panel of Figure 6 a multidimensional model allows to provide a clustering using all (or at least, more) available data. The results in this case are quite satisfactory, even though the edges of the clusters are still noisy and we have some pixels that are completely surrounded by pixels of a different cluster, which is something that we would like to fix with the Potts model. The second, third and fourth panels of Figure 6 show the distribution of the components and how they were classified.

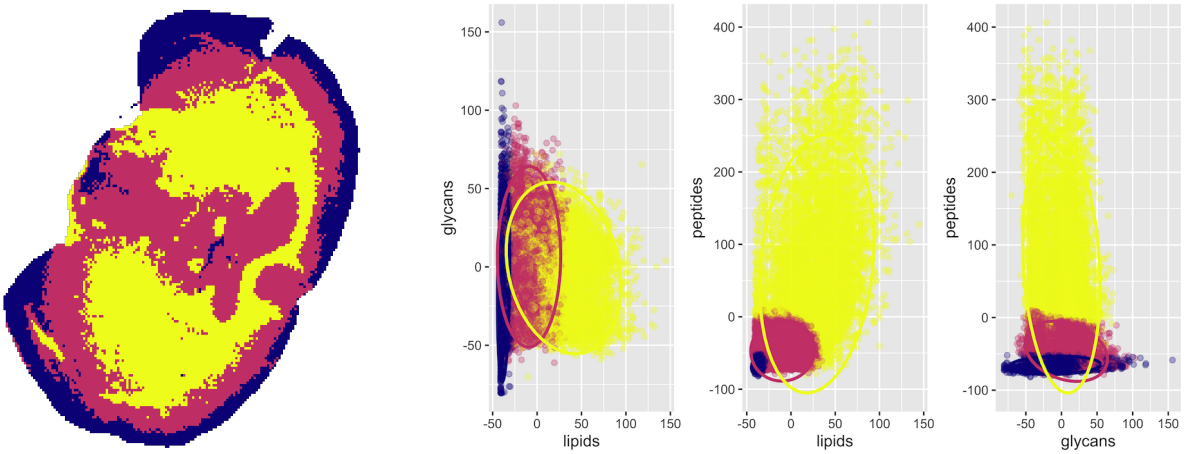

Figure 6: Combined dataset. Left-most panel: clustering obtained via GMM. Second, third and fourth panel: distribution of the fPC for each couple of molecular type and respective clustering allocation.

We highlight that, when trying to improve the performance of the clustering on a specific dataset by incorporating more than one principal component, we run into a problem. The different components in a multivariate GMM all share the same "importance" when performing the clustering, while usually the first fPC is extremely more representative (in terms of explained variance) than the second, the second more representative than the third, and so on.

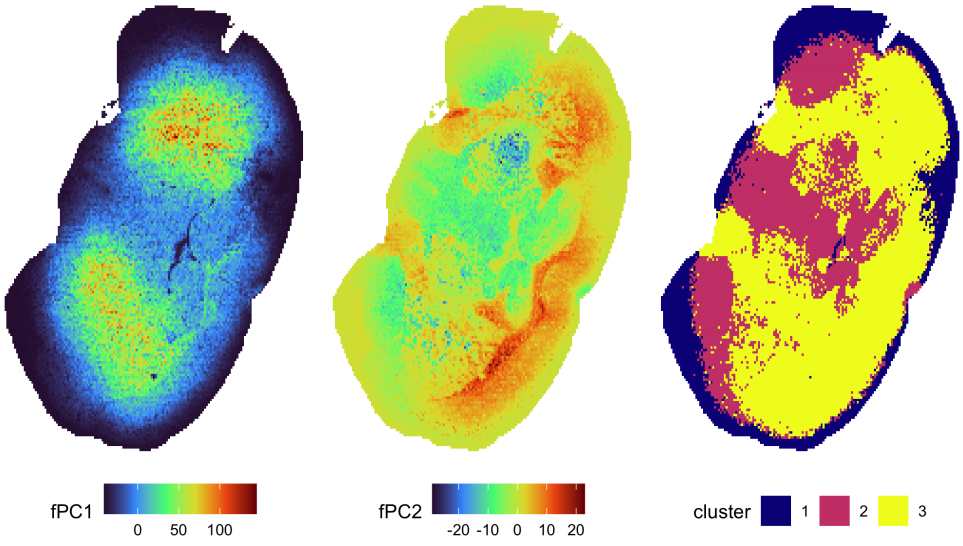

Figure 7: Lipids dataset: Left panel: first fPC distribution. Center panel: second fPC distribution. Right panel: the obtained clustering.

An extreme example of this behavior is the lipids dataset, where the first component explains 95% of the variance. If we use a 2d GMM, we estimate a clustering solution that mainly resembles the structure of the second principal component, as Figure 7 displays. A brute force solution would be to run the algorithm on the full dataset rather than the fPCs, which would massively increase computational cost.

2.4. Spatial information in the mixture: univariate Hidden Potts Model

The GMM specified in Sections 2.2 and 2.3 does not take into account any spatial information. We expect neighbouring pixels to share information and to have a higher probability of being clustered together. Following [Besag, 1974, 1986; Potts, 1952], we can specify a hidden Markov random field as the distribution of the membership labels, introducing a dependence of each pixel i on the set of neighbouring pixels $\mathcal{N}_i$.
The Hammersley-Clifford Theorem identifies Markov random fields with Gibbs distributions [Besag, 1975]. This is a valuable result because it provides a straightforward way of specifying the distribution through the representation as Gibbs distribution, which is specified in terms of its conditional probabilities, for $i = 1, \dots, n$:

$$p(z_i|\mathbf{z}_{\setminus \mathbf{i}}, \beta) = \frac{exp(\beta \sum_{i\sim l} \delta(z_i, z_l))}{\sum_{j=1}^{k} exp(\beta \sum_{i\sim l} \delta(j, z_l)))}, \tag{11}$$

where β governs the degree of dependence in the model and is known as the *inverse temperature* due to its origin in statistical physics, and $\mathbf{z}_{\setminus \mathbf{i}}$ represents all the labels except z_i. Moreover, writing $i \sim l$ indicates all the neighbouring pixels of i i.e., $l \in \mathcal{N}_i$. Finally, $\delta(i, j)$ is the Kronecker delta function. Thus $\sum_{i\sim l} \delta(z_i, z_l)$ is a count of the neighbours that share the same label as pixel i.
In this project, we focus on first order neighbours, so each pixel has a neighbourhood composed of 4 pixels: the ones directly above, below, left, and right. Pixels situated at the boundary of the image domain have less than four neighbours. These neighbourhood relationships are reciprocal, so $j \in \mathcal{N}_i$ implies $i \in \mathcal{N}_j$. If $\mathcal{E}$ is the set of all unique neighbour pairs, or edges in the image lattice, then $|\mathcal{E}| = 2(n - \sqrt{n})$ for a square lattice. This choice was due to the resolution of the images obtained via the MALDI-MSI method, where each pixel represented an area of around $50\mu m$ which contains a few cells, and therefore a bigger neighbourhood would have captured too many cells, compromising the results. We want to stress that our implementation, in the function `GibbsPotts`, can handle an arbitrary order neighbourhood. The hidden Potts model can thus be viewed as a spatially-correlated generalisation of the finite mixture model [Rydén and Titterington, 1998]. We follow Geman and Geman [1984] in assuming that the pixels with label k share a common mean μ_k corrupted by additive Gaussian noise with variance σ_k^2. The model is the following:

$$\begin{aligned} y_i \mid z_i = k, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 &\overset{\text{iid}}{\sim} \mathcal{N}(\mu_k, \sigma_k^2), \\ \mu_k &\sim \mathcal{N}(m_{k,0}, s_{k,0}^2), \\ \sigma_k^2 &\sim Inv-Gamma\left(\frac{n_k}{2}, \frac{n_k v_k}{2}\right), \\ z_i \mid \mathbf{z}_{\setminus \mathbf{i}} &\sim Gibbs(\beta). \end{aligned} \tag{12}$$

Note that the joint distribution of all of the pixel labels can be expressed in the form of an exponential family [Grelaud et al., 2009]:

$$p(\boldsymbol{z} \mid \beta) = exp\{\beta S(\mathbf{z}) - log\mathcal{C}(\beta)\}. \tag{13}$$

The sufficient statistic $S(\boldsymbol{z}) = \sum_{i\sim l \in \mathcal{E}} \delta(z_i, z_l)$ represents the total number of like neighbour pairs in the image, while $\mathcal{C}(\beta)$ is a normalizing constant.
The augmented likelihood $p(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \beta)$ can therefore be factorised into $p(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}) p(\boldsymbol{z} \mid \beta)$, where the second factor does not depend on the observed data, but only on the sufficient statistics.
The parameter β plays a key role in the model since it governs the degree of interaction between the neighbouring pixels. It is difficult to set this parameter by trial and error, particularly for noisy images. The Potts model undergoes a phase transition at a critical value of β, switching from a disordered to an ordered state. Potts [Potts, 1952] showed that the critical value for a regular 2D lattice is defined as:

$$\beta_{critic} = log(1 + \sqrt{K}). \tag{14}$$

This is the value of the critical inverse temperature for a lattice with infinite columns, so in principle the critical value for our images would be different. However, the error introduced by a finite boundary diminishes as n increases.
We first start by fixing the inverse temperature a priori, choosing its best value by trial and error. Later on, we

introduce it as part of the parameters to be estimated from the data. See Sections 2.6 for more details.

In order to sample from the posterior we used a Gibbs sampler, following the full conditionals reported in equation (15) and (16). Similarly to the GMM case, our implementation of this Gibbs sampler is available on GitHub in the `GibbsPotts` function.

$$
\begin{aligned}
&\mu_k \mid \boldsymbol{\sigma}^2, \boldsymbol{y}, \boldsymbol{z} \sim \mathcal{N}(m_{k,p}, s^2_{k,p}),\\
&\sigma^2_k \mid \boldsymbol{\mu}, \boldsymbol{y}, \boldsymbol{z} \sim Inv-Gamma\left(\frac{n_k}{2}+\frac{N_k}{2}, \frac{n_k v_k + \sum_{i:z_i=k}(y_i-\mu_k)^2}{2}\right),\\
&m_{k,p} = \frac{N_k s^2_k \overline{y}_{n,k} + m_k \sigma^2_k}{N_k s^2_k + m_j \sigma^2_k} \qquad s^2_{k,p} = \frac{n_k v_k + \sum_{i:z_i=k}(y_i-\mu_k)^2}{2},\\
&N_k = \sum_{i=k}^{N} \mathbb{1}_{\{z_i=k\}} \qquad \overline{y}_{n,k} = \frac{\sum_{i:z_i=k} y_i}{N_k}.
\end{aligned}
\tag{15}
$$

Up to here we have the same full conditionals of the GMM (8). The clustering allocation variables, on the other hand, have a different full conditional distribution:

$$
\begin{aligned}
&z_i \mid \boldsymbol{z}_{\backslash \boldsymbol{i}}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{y} \sim Categorical(\boldsymbol{p_i}),\\
&(\boldsymbol{p_i})_k = \frac{\phi(y_i \mid \mu_k, \sigma^2_k) exp\{\beta \sum_{i\sim l} \delta(k, z_l)\}}{\sum_{k=1}^{K} \phi(y_i \mid \mu_k, \sigma^2_k) exp\{\beta \sum_{i\sim l} \delta(k, z_l)\}}.
\end{aligned}
\tag{16}
$$

In equation (16), $\phi(x \mid \mu, \sigma^2)$ represents the probability density function of a normal distribution with mean μ and variance σ^2 evaluated in x.
For the complete calculation of the full conditionals refer to Appendix A.

2.4.1. Hidden Potts Model results on lipids: the importance of β

Figure 8 shows that choosing an appropriate value for β is key to our problem: when $\beta = 0$ we have just a GMM. For a low value of beta the cluster boundaries are not well defined and we have a noisy clusters. However, as we increase the beta we get more and more homogeneous clusters but we loose detail about their shapes and we tend to have a dominant cluster in the image.

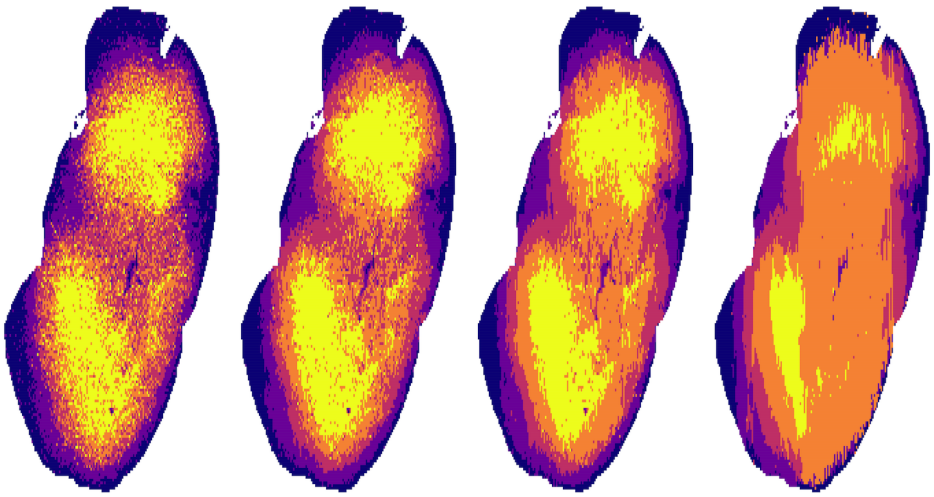

Figure 8: Potts model on the lipids dataset for different values of the inverse temperature. From left to right $\beta = 0$, $0.5\beta_{critic}$, β_{critic}, $1.5\beta_{critic}$.

These difficulties in fixing a correct value for beta lead us to incorporate the beta in the model as a parameter to be estimated (Section 2.6).

2.5. Multidimensional Hidden Potts Model

As in the GMM case, also here we want to take into account more than one fPC or data from more than one dataset: therefore, we extend the model to a multidimensional formulation. The model is as follows:

$$\begin{aligned} \boldsymbol{y_i} \mid z_i = k, \boldsymbol{\mu}, \boldsymbol{\Sigma} &\overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu_k}, \Sigma_k), \\ \boldsymbol{\mu}_k &\sim \mathcal{N}(\boldsymbol{b}_{k,0}, B_{k,0}), \\ \Sigma_k &\sim Inv - Wishart(V_{k,0}, n_{k,0}), \\ z_i \mid \mathbf{z_{\setminus i}} &\sim Gibbs(\beta). \end{aligned} \tag{17}$$

In order to sample from the posterior we used a Gibbs sampler; the full conditionals of this model are reported in equation (18). Our implementation of this Gibbs sampler is available on GitHub in the `GibbsPotts` function.

$$\begin{aligned} \boldsymbol{\mu_k} \mid \boldsymbol{\Sigma}, \boldsymbol{y}, \boldsymbol{z} &\sim \mathcal{N}(\boldsymbol{b_{k,p}}, B_{k,p}), \\ \Sigma_k \mid \boldsymbol{\mu}, \boldsymbol{y}, \boldsymbol{z} &\sim Inv - Wishart(V_{k,p}, n_{k,p}), \\ B_{k,p} &= (N_k \Sigma_k^{-1} + B_{k,0}^{-1})^{-1}, \\ \boldsymbol{b}_{k,p} &= B_{k,p}(N_k \Sigma_k^{-1} \overline{\boldsymbol{y}}_{\boldsymbol{k}} + B_{k,0}^{-1} \boldsymbol{b}_{k,0}), \\ N_k = \sum_{i=k}^{N} \mathbb{1}_{\{z_i = k\}}, &\qquad \overline{\boldsymbol{y}}_{\boldsymbol{k}} = \frac{\sum_{i:z_i=k} \boldsymbol{y_i}}{N_k}, \\ V_{k,p} &= \left(V_{k,0}^{-1} + \sum_{i:z_i=k} (\boldsymbol{y}_i - \boldsymbol{\mu}_k)(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T \right)^{-1}, \\ n_{k,p} &= n_{k,0} + N_k, \\ z_i \mid \boldsymbol{z_{\setminus i}}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{y} &\sim Categorical(\boldsymbol{p_i}), \\ (\boldsymbol{p_i})_k &= \frac{\phi(\boldsymbol{y_i} \mid \boldsymbol{\mu_k}, \Sigma_k) exp\{\beta \sum_{i \sim l} \delta(k, z_l)\}}{\sum_{k=1}^{K} \phi(\boldsymbol{y_i} \mid \boldsymbol{\mu_k}, \Sigma_k) exp\{\beta \sum_{i \sim l} \delta(k, z_l)\}}. \end{aligned} \tag{18}$$

In equation (18), $\phi(\boldsymbol{x} \mid \boldsymbol{\mu}, \Sigma)$ represents the probability density function of a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ evaluated in $\boldsymbol{x}$. For the complete calculation of the full conditionals refer to Appendix B.

2.5.1. Results of the MHPM on combined datasets

The multivariate Potts model allowed us to improve our clustering on the combined dataset, as can be seen in Figure 9. For the dataset in question we found the best value of β to be 0.6, which is about half the value of β_{critic} for $K = 6$.

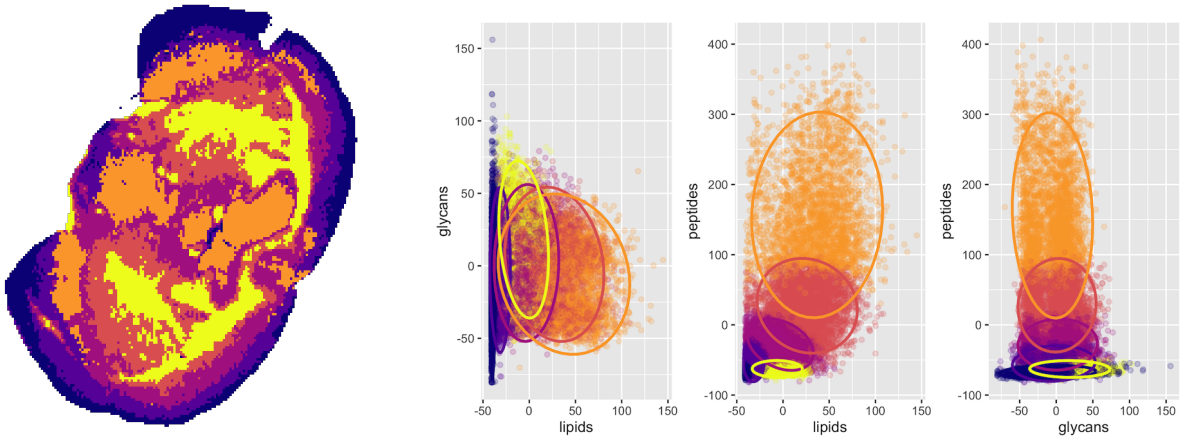

Figure 9: Combined dataset: Left-most panel: clustering obtained via multivariate HPM. Second, third and fourth panel: distribution of the fPC for each couple of molecular type an respective clustering.

Figure 10 shows the comparison between the clustering via multidimensional GMM and via HPM on the combined dataset, using the same priors for the parameters. The differences are remarkable: it is clearly noticeable that the Potts model gives us a much more uniform clustering, as the isolated points are significantly reduced; moreover, the edges of the cluster are sharper. In this case around 25% of the pixels have differing allocations under the two models.

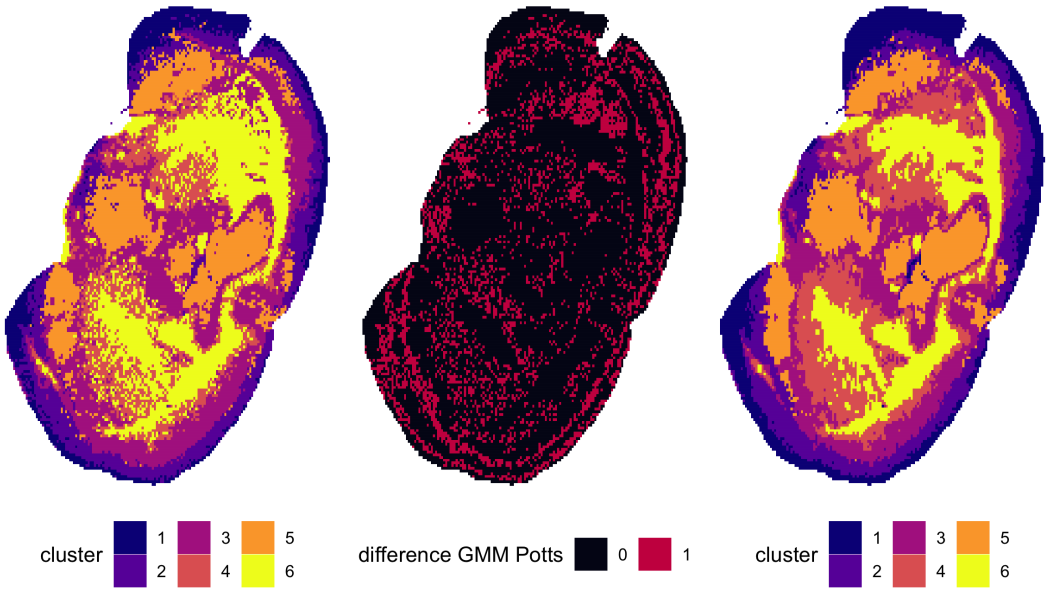

Figure 10: Clustering comparison on combined dataset. Left panel: clustering via GMM. Right panel: clustering via HPM. Center Panel: the pixels for which the cluster allocation differs are highlighted in red.

2.6. Potts model including β as parameter

The parameter β governs the degree of spatial dependence in the model and it is quite difficult to estimate by trial and error. Rather than using a fixed value, we now estimate it as part of the model fitting.
Including β in the model, the augmented joint distribution becomes:

$$p(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta) = \left[\prod_{i=1}^{n} p(\boldsymbol{y_i} \mid z_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})\right] p(\boldsymbol{z} \mid \beta)\pi(\boldsymbol{\mu})\pi(\boldsymbol{\Sigma})\pi(\boldsymbol{\beta}). \tag{19}$$

We then need to introduce a prior for the inverse temperature. Since β takes real values and we know the value of the phase transition β_{critic}, we chose as prior

$$\beta \sim \mathcal{U}([0, \beta_{max}]), \qquad \beta_{max} > \beta_{critic}. \tag{20}$$

We opt for the uniform distribution since we wanted to be non-informative. In particular we choose β to be larger than zero since we want to make neighbouring pixels more likely to be clustered together, aiming at reducing the number of isolated pixels and sharpening cluster edges. The value of β_{max} was chosen to be higher than the critical temperature, not to exclude, a priori, the presence of a dominating class; moreover, since the phase transition is quite abrupt, we are able to cover even cases with a highly dominant class with a value of β_{max} of the order of 2 times β_{critic}.
We can sample from the posterior distribution of this model using a Gibbs sampler, in particular $\pi(\boldsymbol{\mu} \mid \boldsymbol{\Sigma}, \boldsymbol{y}, \boldsymbol{z}, \beta)$, $\pi(\boldsymbol{\Sigma} \mid \boldsymbol{\mu}, \boldsymbol{y}, \boldsymbol{z}, \beta)$ and $\pi(\boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{y}, \beta)$ follow the same full conditional of equation (18). On the other hand, $p(\beta \mid \boldsymbol{z})$ involves an intractable normalizing constant since

$$p(\beta \mid \boldsymbol{z}) = \frac{p(\boldsymbol{z} \mid \beta)\pi(\beta)}{\int p(\boldsymbol{z} \mid \beta)\pi(\beta)\, d\beta} \propto \frac{exp\{\beta S(\boldsymbol{z})\}}{\mathcal{C}(\beta)}\pi(\beta). \tag{21}$$

The normalising constant $\mathcal{C}(\beta)$, also known as *partition function* in statistical physics, has a computational complexity of $\mathcal{O}(nK^n)$, because it involves a sum over all possible configurations of the labels $\boldsymbol{z} \in \mathcal{Z}$:

$$\mathcal{C}(\beta) = \sum_{\boldsymbol{z} \in \mathcal{Z}} exp\{\beta S(\boldsymbol{z})\}. \tag{22}$$

Since it is unfeasible to calculate this value exactly for large images, a computational approximation is required. There are various ways to address this problem. One of them consists in replacing $p(\boldsymbol{z} \mid \beta)$ with the *pseudolikelihood* [Heikkinen and Hogmander, 1994]:

$$p(\boldsymbol{z} \mid \beta) \approx \prod_{i=1}^{n} p(z_i \mid \boldsymbol{z_{\setminus i}}, \boldsymbol{\beta}) = \prod_{i=1}^{n} p(z_i \mid \boldsymbol{z_{\delta_i}}, \boldsymbol{\beta}), \tag{23}$$

where z_{δ_i} indicates the z_j such that $j \in \mathcal{N}_i$. The normalising constants for the factors of the pseudolikelihood are now computable, and so the unobtainable quantity has been replaced by an obtainable approximation. *Pseudolikelihood* is exact when $\beta = 0$ and provides a reasonable approximation for small values of the inverse temperature. However, the approximation error increases rapidly for $\beta \geq \beta_{critic}$, due to long-range dependence between the labels, which is inadequately modelled by the local approximation.
This approximation enables updates for the inverse temperature at a certain iteration of the Gibbs sampler to be simulated using a Metropolis-Hastings step. For theoretical guarantees of the asymptotic behaviour of the hybrid Gibbs-Hastings algorithm refer to Rydén and Titterington [1998].
The proposal density $q(\beta' \mid \beta_{t-1})$ can, in theory, be any probability distribution, however, there is a trade-off between adequately exploring the parameter space and making sure that the acceptance probability is relatively high. In order to find the correct balance between exploration and acceptance rate we used an adaptive random walk Metropolis Hastings algorithm with Gaussian proposal [Garthwaite et al., 2016], which automatically log-linearly tunes the bandwidth, i.e., the covariance of the proposal distribution, to target a specific Metropolis Hastings acceptance rate, set at 0.44 following [Roberts and Rosenthal, 2001].
Our implementation of this Gibbs sampler is available on GitHub in the `MCMCPotts` function.

2.6.1. Numerical Results

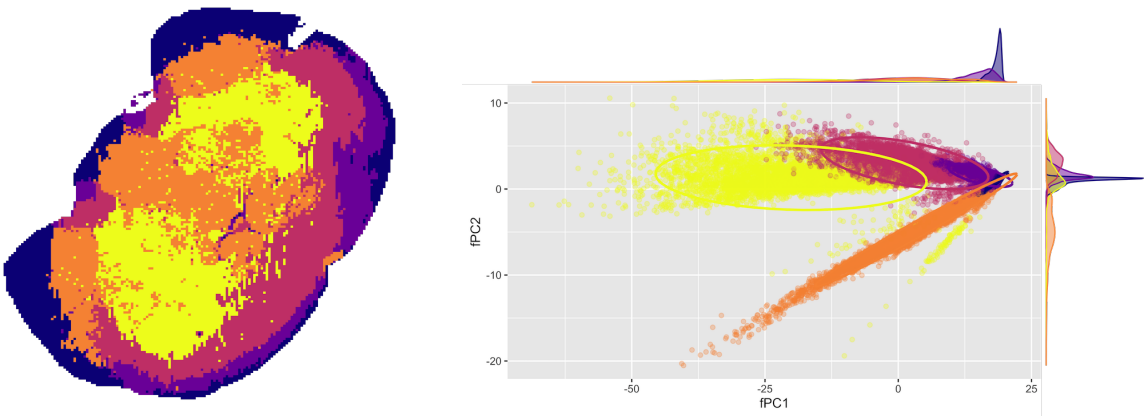

Figure 11: Lipids dataset. Left panel: clustering obtained via HPM with β as parameter. Right panel: distribution of the fPC and respective clustering allocation.

In Figure 11 we see that we are able to estimate a satisfactory clustering even when estimating β as a parameter. In this case the estimated value for the inverse temperature was around 1.4, which is higher than the critical value. This tendency of the model to estimate β larger than the critical value was quite common in our experiments. In some cases, the estimate of beta was way off, as it went considerably above the critical value and produced a clustering having a dominating class, as in Figure 12 left.
To partially fix this problem, we noticed that considering a square image with an added background class helped the algorithm, and resulted in more suitable clusters, as in Figure 12 right, but it did not reduce the tendency of the estimate of β from being much higher than the critical value.

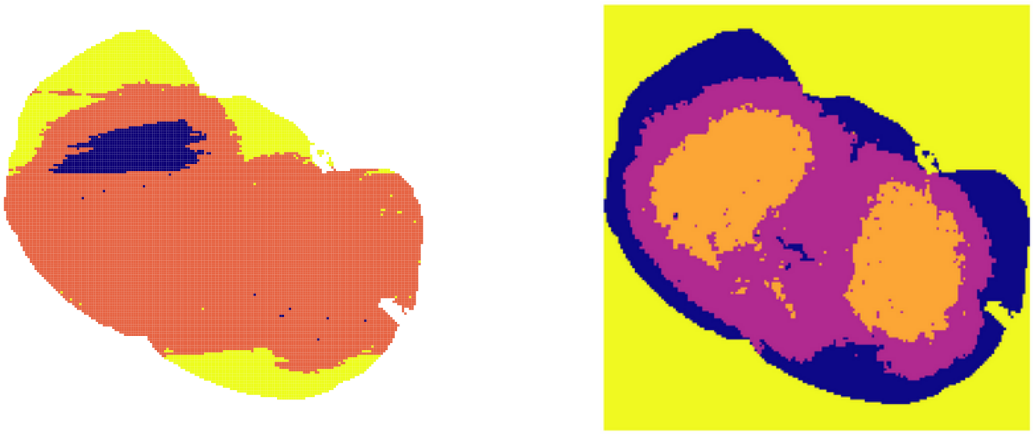

Figure 12: Lipids dataset. Left panel: an extreme case of overestimating β, resulting in a meaningless clustering. Right panel: clustering the square image using an auxiliary background class and the same prior parameters.

3. Conclusions and Further Developments

In this project, we addressed the problem of image segmentation, starting from the standard GMM. Such model, however, does not take into account the spatial dependence between different pixels. We therefore moved to the HPM, introducing such spatial relation via a random Markov field on the cluster allocation variables. This model, which can be seen as a spatially correlated generalization of a GMM, introduces a new parameter, the hidden temperature, which is quite difficult to estimate a priori. Therefore, we incorporated it as part of the parameters to be estimated. This choice introduced some complications since, in order to compute the posterior of the inverse temperature, we had to compute an intractable constant, which lead us to the approximate procedure of *pseudolikelihood* to obtain a computable approximation of such constant.
We implemented in `Rcpp` an efficient Gibbs Sampler for all of the methods above, capable of handling images of arbitrary shape and number of channels. There's still room for improvement, addressing all the challenges that arose. Some possible future developments stem from our work, and include:

- Finding a way of weighting the fPCs, in order to give more importance to the ones explaining most of the variance, to obtain more meaningful clusters;
- Reducing the computational cost: Variational Inference could be used as a different model estimation technique, as opposed to Gibbs sampling;
- Exploring a non-parametric framework, in order to avoid fixing the number of clusters a priori.

References

Walch Aichler. MALDI Imaging mass spectrometry: current frontiers and perspectives in pathology research and practice. *Laboratory Investigation*, 95(4):422 – 431, 2015. doi: 10.1038/labinvest.2014.156. URL https://doi.org/10.1038/labinvest.2014.156.

Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974. ISSN 00359246. URL http://www.jstor.org/stable/2984812.

Julian Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24(3):179–195, 1975. ISSN 00390526, 14679884. URL http://www.jstor.org/stable/2987782.

Julian Besag. On the statistical analysis of dirty pictures. *Journal of the royal statistical society series b-methodological*, 48:259–279, 1986.

D. A. Binder. Bayesian cluster analysis. *Biometrika*, 65(1):31–38, 1978. ISSN 00063444. URL http://www.jstor.org/stable/2335273.

Kristin J Boggio, Emmanuel Obasuyi, Ken Sugino, Sacha B Nelson, Nathalie YR Agar, and Jeffrey N Agar. Recent advances in single-cell maldi mass spectrometry imaging and potential clinical impact. *Expert Review of Proteomics*, 8(5):591–604, 2011. doi: 10.1586/epr.11.53. URL https://doi.org/10.1586/epr.11.53. PMID: 21999830.

David B. Dahl, Devin J. Johnson, and Peter Müller. Search algorithms and loss functions for bayesian clustering. *Journal of Computational and Graphical Statistics*, 31(4):1189–1201, 2022. doi: 10.1080/10618600.2022.2069779. URL https://doi.org/10.1080/10618600.2022.2069779.

Dirk Eddelbuettel and Conrad Sanderson. Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis*, 71:1054–1063, March 2014. URL http://dx.doi.org/10.1016/j.csda.2013.02.005.

Arno Fritsch and Katja Ickstadt. Improved criteria for clustering based on the posterior similarity matrix. *Bayesian Analysis*, 4(2):367 – 391, 2009. doi: 10.1214/09-BA414. URL https://doi.org/10.1214/09-BA414.

P. H. Garthwaite, Y. Fan, and S. A. Sisson. Adaptive optimal scaling of metropolis–hastings algorithms using the robbins–monro process. *Communications in Statistics - Theory and Methods*, 45(17):5098–5111, 2016. doi: 10.1080/03610926.2014.936562. URL https://doi.org/10.1080/03610926.2014.936562.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, 1984. doi: 10.1109/TPAMI.1984.4767596.

Aude Grelaud, Jean-Michel Marin, Christian P. Robert, François Rodolphe, and Jean-François Taly. ABC likelihood-free methods for model choice in Gibbs random fields. *Bayesian Analysis*, 4(2):317 – 335, 2009. doi: 10.1214/09-BA412. URL https://doi.org/10.1214/09-BA412.

Juha Heikkinen and Harri Hogmander. Fully bayesian approach to image restoration with an application in biogeography. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 43(4):569–582, 1994. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2986258.

John W. Lau and Peter J. Green. Bayesian model-based clustering procedures. *Journal of Computational and Graphical Statistics*, 16(3):526–558, 2007. ISSN 10618600. URL http://www.jstor.org/stable/27594259.

Matthew T. Moores, Dai Feng, and Kerrie Mengersen. *bayesImageS: Bayesian Methods for Image Segmentation using a Potts Model*, 2021. URL https://CRAN.R-project.org/package=bayesImageS.

R. B. Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48(1):106–109, 1952. doi: 10.1017/S0305004100027419.

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, 2005. ISBN 9780387400808. URL http://www.worldcat.org/isbn/9780387400808.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351 – 367, 2001. doi: 10.1214/ss/1015346320. URL https://doi.org/10.1214/ss/1015346320.

Tatiana C. Rohner, Dieter Staab, and Markus Stoeckli. Maldi mass spectrometric imaging of biological tissue sections. *Mechanisms of Ageing and Development*, 126(1):177–185, 2005. ISSN 0047-6374. doi: https://doi.org/10.1016/j.mad.2004.09.032. URL https://www.sciencedirect.com/science/article/pii/S0047637404002349. Functional Genomics of Ageing II.

Tobias Rydén and D. M. Titterington. Computational bayesian analysis of hidden markov models. *Journal of Computational and Graphical Statistics*, 7(2):194–211, 1998. doi: 10.1080/10618600.1998.10474770. URL https://www.tandfonline.com/doi/abs/10.1080/10618600.1998.10474770.

A. Unidimensional Potts model - full conditionals computations

$$p(z_i|\mathbf{z}_{\setminus \mathbf{i}}, \beta) = \frac{exp(\beta \sum_{i\sim l} \delta(z_i, z_l))}{\sum_{j=1}^{k} exp(\beta \sum_{i\sim l} \delta(j, z_l)))}$$

$$y_i|z_i = k, \mu_k, \sigma_k^2 \sim \mathcal{N}(\mu_k, \sigma_k^2)$$

$$\mu_k \sim \mathcal{N}(m_{k,0}, s_{k,0}^2)$$

$$\sigma_k^2 \sim Inverse - Gamma(\frac{n_k}{2}, \frac{n_k v_k}{2})$$

Note the Gibbs distribution has a sufficient statistic:

$$S(\mathbf{z}) = \sum_{i\sim l \in \mathcal{E}} \delta(z_i, z_l)$$

The posterior

$$p(\boldsymbol{\mu}, \boldsymbol{\sigma^2}, \boldsymbol{z}|\boldsymbol{y}, \beta) \propto p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}) \pi(\boldsymbol{\mu}, \boldsymbol{\sigma^2}) p(\boldsymbol{z}|\beta)$$

$$\propto \left(\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{z_i}^2}} e^{-\frac{(y_i - \mu_{z_i})^2}{2\sigma_{z_i}^2}} \right) exp\{\beta S(\boldsymbol{z})\} \left(\prod_{k=1}^{K} \frac{1}{\sqrt{2\pi s_{k,0}^2}} e^{-\frac{(\mu_k - m_{k,0})^2}{2s_{k,0}^2}} \right) \left(\prod_{k=1}^{K} (\sigma_k^2)^{-\frac{n_k}{2}-1} e^{-\frac{n_k v_k}{2\sigma_k^2}} \right)$$

A.1. Means

$$\boldsymbol{\mu}|\boldsymbol{\sigma^2}, \boldsymbol{z}, \boldsymbol{y} \propto \left[\prod_{k=1}^{K} \left(\prod_{i|z_i=k} \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(y_i-\mu_k)^2}{2\sigma_k^2}} \right) \right] \left[\prod_{k=1}^{K} \frac{1}{\sqrt{2\pi s_k^2}} e^{-\frac{(\mu_k - m_{k,0})^2}{2s_{k,0}^2}} \right]$$

$$= \prod_{k=1}^{K} \left(\frac{1}{2\pi\sigma_k^2} \left(\prod_{i|z_i=k} e^{-\frac{(y_i-\mu_k)^2}{2\sigma_k^2}} \right) e^{-\frac{(\mu_j - m_{k,0})^2}{2s_{k,0}^2}} \right)$$

$$\propto \prod_{k=1}^{K} \left(\frac{1}{2\pi\sigma_k^2} e^{-\frac{1}{2\sigma_k^2}\sum_{i:z_i=k}(y_i-\mu_k)^2} e^{-\frac{(\mu_k - m_{k,0})^2}{2s_{k,0}^2}} \right)$$

focusing on the exponent:

$$\frac{1}{\sigma_k^2} \sum_{i:z_i=k} (\mu_k - \overline{y}_{n,k} + \overline{y}_{n,k} - y_i)^2 = \frac{N_k}{\sigma_k^2}(\mu_k - \overline{y}_{n,k})^2 + \underbrace{\frac{1}{\sigma_k^2} \sum_{i:z_i=k} (y_i - \overline{y}_{n,k})^2}_{\text{constant w.r.t. } \mu_k}$$

where:

$$N_k = \sum_{i=k}^{N} \mathbb{1}_{\{z_i=k\}} \qquad \overline{y}_{n,k} = \frac{\sum_{i:z_i=k} y_i}{N_k} \qquad k = 1, \ldots, K$$

$$= \frac{N_k}{\sigma_k^2}(\mu_k - \overline{y}_{n,k})^2 + \frac{1}{s_{k,0}^2}(\mu_k - m_{k,0})^2 = \left(\frac{N_k}{\sigma_k^2} + \frac{1}{s_{k,0}^2} \right) \left(\mu_k - \underbrace{\frac{\frac{N_k}{\sigma_k^2}\overline{y}_{n,k} + \frac{m_{k,0}}{s_{k,0}^2}}{\frac{N_k}{\sigma_k^2} + \frac{m_{k,0}}{s_{k,0}^2}}}_{m_{k,p}} \right)^2 + \underbrace{\frac{\frac{N_k}{\sigma_k^2}\frac{m_{k,0}}{s_{k,0}^2}}{\frac{N_k}{\sigma_k^2} + \frac{m_{k,0}}{s_{k,0}^2}}(\overline{y}_{n,k} - m_{k,0})^2}_{\text{constant w.r.t.} \mu_k}$$

$$\propto \prod_{k=1}^{K} exp\left\{ -\frac{1}{2} \left(\frac{N_k}{\sigma_k^2} + \frac{1}{s_{k,0}^2} \right) (\mu_k - m_{k,p})^2 \right\}$$

$$\mu_k|\sigma_k, \mathbf{z}, \mathbf{y} \sim \mathcal{N}\left(\frac{N_k s_{k,0}^2 \overline{y}_{n,k} + m_{k,0}\sigma_k^2}{N_k s_{k,0}^2 + m_{0,k}\sigma_k^2}, \frac{\sigma_k^2 s_{0,k}^2}{N_k s_{0,k}^2 + \sigma_k^2} \right) \qquad k = 1, ..., K$$

A.2. Variances

$$\boldsymbol{\sigma^2}|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{y} \propto \prod_{k=1}^{K}\left[\left(\prod_{i|z_i=k}\frac{1}{\sqrt{2\pi\sigma_k^2}}e^{-\frac{y_i-\mu_k}{2\sigma_k^2}}\right)(\sigma_k^2)^{-\frac{n_k}{2}-1}e^{-\frac{n_k v_k}{2\sigma_k^2}}\right]$$

$$\propto \prod_{k=1}^{K}\left((\sigma_j^2)^{-\frac{n_k}{2}-1-\frac{N_k}{2}}exp\left\{-\frac{1}{2\sigma_k^2}\left(\sum_{i|z_i=k}(y_i-\mu_k)^2-n_k v_k\right)\right\}\right)$$

$$\sigma_k|\boldsymbol{z},\mu_k,\boldsymbol{y} \sim Inverse-Gamma\left(\frac{n_k}{2}+\frac{N_k}{2},\frac{n_k v_k+\sum_{i:z_i=k}(y_i-\mu_k)^2}{2}\right) \qquad k=1,...,K$$

A.3. Labels

$$\boldsymbol{z}|\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\sigma} \propto \left(\prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma_{z_i}^2}}e^{-\frac{(y_i-\mu_{z_i})^2}{2\sigma_{z_i}^2}}\right)e^{\beta S(\boldsymbol{z})}$$

let's focus on $z_i|\boldsymbol{z_{\setminus i}},\beta,y_i,\mu_{z_i},\sigma_{z_i}^2$

$$\mathbb{P}(z_i=k|\boldsymbol{z_{\setminus i}},y_i,\mu_k,\sigma_k^2)=\frac{\frac{1}{\sqrt{2\pi\sigma_k^2}}e^{-\frac{(y_i-\mu_k)^2}{2\sigma_k^2}}\frac{exp\{\beta\sum_{i\sim l}\delta(j,z_l)\}}{\sum_{k=1}^{K}exp\{\beta\sum_{i\sim l}\delta(k,z_l)\}}}{\sum_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_k^2}}e^{-\frac{(y_i-\mu_k)^2}{2\sigma_k^2}}\frac{exp\{\beta\sum_{i\sim l}\delta(j,z_l)\}}{\sum_{k=1}^{K}exp\{\beta\sum_{i\sim l}\delta(k,z_l)\}}}=$$

$$=\underbrace{\frac{\frac{1}{\sqrt{2\pi\sigma_k^2}}e^{-\frac{(y_i-\mu_k)^2}{2\sigma_k^2}}exp\{\beta\sum_{i\sim l}\delta(j,z_l)\}}{\sum_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_k^2}}e^{-\frac{(y_i-\mu_k)^2}{2\sigma_k^2}}exp\{\beta\sum_{i\sim l}\delta(j,z_l)\}}}_{\boldsymbol{p_{i,k}}} \qquad i=1,...,N \qquad k=1,...,K$$

$$z_i|\boldsymbol{\mu},\boldsymbol{\sigma}^2,\boldsymbol{y},\boldsymbol{z_{\setminus i}} \sim Categorical(\mathbf{p}_i) \qquad i=1,...,N$$

B. Multivariate Potts model - full conditionals computations

$$\begin{aligned} \boldsymbol{y_i} \mid z_i = k, \boldsymbol{\mu}, \boldsymbol{\Sigma} &\overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu_k}, \Sigma_k) && i = 1, \dots, n \\ \boldsymbol{\mu}_k &\sim \mathcal{N}(\boldsymbol{b}_{k,0}, B_{k,0}) && k = 1, \dots, K \\ \Sigma_k &\sim Inverse - Wishart(V_{k,0}, n_{k,0}) && k = 1, \dots, K \\ z_i \mid \mathbf{z}_{\setminus \mathbf{i}} &\sim Gibbs(\beta) && i = 1, \dots, n \end{aligned}$$

we indicate $\Lambda_k = \Sigma_k^{-1} \sim Wishart(V_{k,0} n_{k,0}), \quad k = 1, ..., K$ The posterior

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{z} | \boldsymbol{y}) \propto p(\boldsymbol{y} | \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \pi(\boldsymbol{\mu}) \pi(\boldsymbol{\Sigma}) p(\boldsymbol{z} | \beta)$$

$$\propto \left(\prod_{i=1}^{N} \frac{\exp\{-\frac{1}{2}(\boldsymbol{y_i} - \boldsymbol{\mu_{z_i}})^T \Lambda_{z_i} (\boldsymbol{y_i} - \boldsymbol{\mu_{z_i}})\}}{(2\pi)^{\frac{d}{2}} \det(\Lambda_{z_i})^{-\frac{1}{2}}} \right) \left(\prod_{k=1}^{K} \frac{\exp\{-\frac{1}{2}(\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}})^T B_{k,0}^{-1} (\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}})\}}{(2\pi)^{\frac{d}{2}} \det(B_{k,0})^{\frac{1}{2}}} \right)$$
$$\left(\prod_{k=1}^{K} \frac{\det(\Lambda_k)^{\frac{n_{k,0}-d-1}{2}} \exp\{-\frac{1}{2} Trace(\Lambda_k V_{k,0}^{-1})\}}{2^{n_{k,0}\frac{d}{2}} \det(V_{k,0})^{\frac{1}{2}} \Gamma_d(\frac{n_{k,0}}{2})} \right) exp\{\beta S(\boldsymbol{z})\}$$

B.1. Means

$$\boldsymbol{\mu_k} | \Sigma_k, \boldsymbol{z}, \boldsymbol{y} \propto \left(\prod_{i:z_i=k} \frac{exp\{-\frac{1}{2}(\boldsymbol{y_i} - \boldsymbol{\mu_k})^T \Lambda_k (\boldsymbol{y_i} - \boldsymbol{\mu_k})\}}{(2\pi)^{\frac{d}{2}} det(\Lambda_k)^{-\frac{1}{2}}} \right) \frac{exp\{-\frac{1}{2}(\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}})^T B_{k,0}^{-1} (\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}})\}}{(2\pi)^{\frac{d}{2}} det(B_{k,0})^{\frac{1}{2}}}$$
$$\propto exp\left\{ -\frac{1}{2} \left[\sum_{i:z_i=k} (\boldsymbol{y_i} - \boldsymbol{\mu_k})^T \Lambda_k (\boldsymbol{y_i} - \boldsymbol{\mu_k}) + (\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}})^T B_{k,0}^{-1} (\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}}) \right] \right\}$$

Define:

$$N_k = \sum_{i=k}^{N} \mathbb{1}_{\{z_i=k\}} \qquad \overline{\boldsymbol{y}}_{\boldsymbol{k}} = \frac{\sum_{i:z_i=k} \boldsymbol{y_i}}{N_k} \qquad k = 1, \dots, K$$

focusing on the square brackets:

$$\sum_{i:z_i=k} (\boldsymbol{y_i} - \overline{\boldsymbol{y}}_{\boldsymbol{k}} + \overline{\boldsymbol{y}}_{\boldsymbol{k}} - \boldsymbol{\mu_k})^T \Lambda_k (\boldsymbol{y_i} - \overline{\boldsymbol{y}}_{\boldsymbol{k}} + \overline{\boldsymbol{y}}_{\boldsymbol{k}} - \boldsymbol{\mu_k})$$
$$= \underbrace{\sum_{i:z_i=k} (\boldsymbol{y_i} - \overline{\boldsymbol{y}}_{\boldsymbol{k}})^T \Lambda_k (\boldsymbol{y_i} - \overline{\boldsymbol{y}}_{\boldsymbol{k}})}_{\text{constant w.r.t. } \boldsymbol{\mu_k}} + \underbrace{2N_k \sum_{i:z_i=k} (\boldsymbol{y_i} - \overline{\boldsymbol{y}}_{\boldsymbol{k}})^T \Lambda_k (\overline{\boldsymbol{y}}_{\boldsymbol{k}} - \boldsymbol{\mu_k})}_{=0} + N_k \sum_{i:z_i=k} (\boldsymbol{\mu_k} - \overline{\boldsymbol{y}}_{\boldsymbol{k}})^T \Lambda_k (\boldsymbol{\mu_k} - \overline{\boldsymbol{y}}_{\boldsymbol{k}})$$

So the full conditional is proportional to:

$$\exp\left\{ -\frac{1}{2} \underbrace{\left[N_k (\boldsymbol{\mu_k} - \overline{\boldsymbol{y}}_{\boldsymbol{k}})^T \Lambda_k (\boldsymbol{\mu_k} - \overline{\boldsymbol{y}}_{\boldsymbol{k}}) + (\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}})^T B_{k,0}^{-1} (\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}}) \right]}_{} \right\}$$
$$N_j (\boldsymbol{\mu_k} - \overline{\boldsymbol{y}}_{\boldsymbol{k}})^T \Lambda_k (\boldsymbol{\mu_k} - \overline{\boldsymbol{y}}_{\boldsymbol{k}}) + (\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}})^T B_{k,0}^{-1} (\boldsymbol{\mu_k} - \boldsymbol{b_{k,0}})$$
$$= N_j (\boldsymbol{\mu_k}^T \Lambda_k \boldsymbol{\mu_k} - 2\boldsymbol{\mu_k}^T \Lambda_k \overline{\boldsymbol{y}}_{\boldsymbol{k}} + \overline{\boldsymbol{y}}_{\boldsymbol{k}}^T \Lambda_k \overline{\boldsymbol{y}}_{\boldsymbol{k}}) + \boldsymbol{\mu_k}^T B_{k,0}^{-1} \boldsymbol{\mu_k} - 2\boldsymbol{\mu_k} B_{k,j0}^{-1} \boldsymbol{b_{k,0}} + \boldsymbol{b_{k,0}} B_{k,0}^{-1} \boldsymbol{b_{k,0}}$$
$$= \boldsymbol{\mu_k}^T (N_k \Lambda_j + B_{k,0}^{-1}) \boldsymbol{\mu_k}^T - 2\boldsymbol{\mu_k}^T (N_k \Lambda_j \overline{\boldsymbol{y}}_{\boldsymbol{k}} + B_{k,0}^{-1} \boldsymbol{b_{k,0}}) + const$$
$$= \boldsymbol{\mu_k}^T \underbrace{(N_k \Lambda_k + B_{k,0}^{-1})}_{B_{k,p}^{-1}} \boldsymbol{\mu_k}^T - 2\boldsymbol{\mu_k}^T \underbrace{(N_k \Lambda_k + B_{k,0}^{-1})}_{B_{k,p}^{-1}} \underbrace{\left[\underbrace{(N_k \Lambda_k + B_{k,0}^{-1})^{-1}}_{B_{k,p}} (N_j \Lambda_k \overline{\boldsymbol{y}}_{\boldsymbol{k}} + B_{k,p}^{-1} \boldsymbol{b_{k,p}}) \right]}_{\boldsymbol{b_{k,p}}} + const$$
$$= (\boldsymbol{\mu_k} - \boldsymbol{b_{k,p}})^T B_{k,p}^{-1} (\boldsymbol{\mu_k} - \boldsymbol{b_{k,p}})$$
$$\boldsymbol{\mu_k} | \Sigma_k, \boldsymbol{z}, \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{b_{k,p}}, B_{k,p}) \qquad k = 1, ..., K$$

B.2. Covariance matrices

$$\Sigma_k^{-1} = \Lambda_k|\boldsymbol{\mu_k}, \boldsymbol{z}, \boldsymbol{y_i} \propto \left(\prod_{i|z_i=k} \frac{exp\{\frac{1}{2}(\boldsymbol{y_i}-\boldsymbol{\mu_k})^T\Lambda_l(\boldsymbol{y_i}-\boldsymbol{\mu_k})\}}{(2\pi)^{\frac{d}{2}} det(\Lambda_k)^{-\frac{1}{2}}} \right) \frac{det(\Lambda_k)^{\frac{n_{k,0}-d-1}{2}} exp\{-\frac{1}{2}trace(\Lambda_k V_{k,0}^{-1})\}}{2^{\frac{n_{k,0}d}{2}} det(V_{k,0})^{\frac{n_{k,0}}{2}} \Gamma_d(\frac{n_{k,0}}{2})}$$

$$\propto \frac{exp\{-\frac{1}{2}\sum_{i|z_i=k}(\boldsymbol{y_i}-\boldsymbol{\mu_k})^T\Lambda_k(\boldsymbol{y_i}-\boldsymbol{\mu_k})\}}{det(\Lambda_k)^{-\frac{N_k}{2}}} det(\Lambda_k)^{\frac{n_{k,0}-d-1}{2}} exp\left\{-\frac{1}{2}trace(\Lambda_k V_{k,0}^{-1})\right\}$$

$$\propto exp\left\{-\frac{1}{2}\left[\underbrace{\sum_{i|z_i=k}(\boldsymbol{y_i}-\boldsymbol{\mu_k})^T\Lambda_k(\boldsymbol{y_i}-\boldsymbol{\mu_k})}_{trace(\Lambda_k Y),\ \text{where} Y=\sum_{i|z_i=k}(\boldsymbol{y_i}-\boldsymbol{\mu_k})(\boldsymbol{y_i}-\boldsymbol{\mu_k})^T} + trace(\Lambda_k V_{k,0}^{-1})\right]\right\} det(\Lambda_k)^{\frac{n_{k,0}+N_k-d-1}{2}}$$

$$\propto exp\left\{-\frac{1}{2}trace(\Lambda_j(V_{k,0}^{-}1+Y))\right\} det(\Lambda_k)^{\frac{n_{k,0}+N_k-d-1}{2}}$$

$$\Sigma_k^{-1} = \Lambda_k|\boldsymbol{\mu_k}, \boldsymbol{z}, \boldsymbol{y_i} \sim Wishart\left(\left(V_{k,0}^{-1} + \sum_{i:z_i=k}(\boldsymbol{y_i}-\boldsymbol{\mu_k})(\boldsymbol{y_i}-\boldsymbol{\mu_k})^T\right)^{-1}, n_{k,0}+N_k\right) \qquad j=1,\dots,K$$

B.3. Labels

$$\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\Sigma} \propto \left(\prod_{i=1}^{N} \frac{\exp\{-\frac{1}{2}(\boldsymbol{y_i}-\boldsymbol{\mu_{z_i}})^T\Lambda_{z_i}(\boldsymbol{y_i}-\boldsymbol{\mu_{z_i}})\}}{(2\pi)^{\frac{d}{2}}\det(\Lambda_{z_i})^{-\frac{1}{2}}}\right) e^{\beta S(\boldsymbol{z})}$$

let's focus on $z_i|\boldsymbol{z_{\setminus i}}, \beta, \boldsymbol{y_i}, \boldsymbol{\mu_{z_i}}, \Sigma_{z_i}$

$$\mathbb{P}(z_i=k|\boldsymbol{z_{\setminus i}}, \boldsymbol{\mu_k}, \Sigma_k, \boldsymbol{y_i}, \beta) = \frac{\frac{exp\{-\frac{1}{2}(\boldsymbol{y_i}-\boldsymbol{\mu_k})^T\Lambda_k(\boldsymbol{y_i}-\boldsymbol{\mu_k})\}}{(2\pi)^{\frac{d}{2}}det(\Lambda_k)^{-\frac{1}{2}}} \frac{exp\{\beta\sum_{i\sim l}\delta(j,z_l)\}}{\sum_{k=1}^{K} exp\{\beta\sum_{i\sim l}\delta(k,z_l)\}}}{\sum_{k=1}^{K} \frac{exp\{-\frac{1}{2}(\boldsymbol{y_i}-\boldsymbol{\mu_k})^T\Lambda_k(\boldsymbol{y_i}-\boldsymbol{\mu_k})\}}{(2\pi)^{\frac{d}{2}}det(\Lambda_k)^{-\frac{1}{2}}} \frac{exp\{\beta\sum_{i\sim l}\delta(j,z_l)\}}{\sum_{k=1}^{K} exp\{\beta\sum_{i\sim l}\delta(k,z_l)\}}}$$

$$= \underbrace{\frac{\exp\{-\frac{1}{2}(\boldsymbol{y_i}-\boldsymbol{\mu_k})^T\Lambda_k(\boldsymbol{y_i}-\boldsymbol{\mu_k}) + \beta\sum_{i\sim j}\delta(j,z_l)\}\det(\Lambda_k)^{\frac{1}{2}}}{\sum_{k=1}^{K}\exp\{-\frac{1}{2}(\boldsymbol{y_i}-\boldsymbol{\mu_k})^T\Lambda_k(\boldsymbol{y_i}-\boldsymbol{\mu_k}) + \beta\sum_{i\sim j}\delta(k,z_l)\}\det(\Lambda_k)^{\frac{1}{2}}}}_{\boldsymbol{p_{i,k}}}$$

$$z_i|\boldsymbol{z_{\setminus i}}, \boldsymbol{\mu_k}, \Sigma_k, \boldsymbol{y_i}, \beta \sim Categorical(\boldsymbol{p_i}) \qquad i=1,...,N \qquad k=1,...,K$$