12-15-2014

# Methods for Identifying Regions of Brain Activation Using FMRI Meta-Data

Meredith A. Ray
*University of South Carolina - Columbia*

# Methods for Identifying Regions of Brain Activation using FMRI Meta-data

by

Meredith A. Ray

Bachelor of Arts
Piedmont College 2007

Master of Public Health
University of Georgia 2009

---

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Biostatisitics

The Norman J. Arnold School of Public Health

University of South Carolina

2014

Accepted by:

Hongmei Zhang, Major Professor

Bo Cai, Committee Member

Tawanda Greer, Committee Member

Jian Kang, Outside Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

# Dedication

This body of work is dedicated to my family, in particular, my parents and grandparents. Thank you for the unlimited support and love you have always provided me. You all have given me constant encouragement, patience, and guidance and know that I could not have done this without you. I appreciate everything you have done, no matter how small or large; I will always be grateful so such a wonderful blessing.

# Acknowledgments

Throughout this research there have been many people involved that deserve recognition. To begin, I would like to thank my advisor, Dr. Hongmei Zhang. She been an inspiration and a supportive, patient, and exceptionally instructive mentor. My statistical writing skills and implementation of ideas have improved to her credit and hope to further embrace them for future work.

I would also like to thank my other committee members, Dr. Bo Cai, and Dr. Tawanda Greer, and Dr. Jian Kang. I recognize and understand the amount of effort and time it takes to participate in a graduate student's committee and I greatly appreciate it. The constructive input and advice you each have given over the process has it has been very beneficial in the understanding of my data and methods.

In addition to my committee members, I would like to take this opportunity to thank the Department of Epidemiology and Biostatistics of the Arnold School of Public Health at the University of South Carolina for accepting me into their PhD program so that I could have this opportunity to complete this research. I would also like to thank them for their financial support over the past 4 years for tuition assistance, travel support, and graduate assistantships and the excellent education I received. In particular, I would like to recognize Dr. Jim Burch for his encouragement, guidance, instruction, and his infectious excitement about his research. I would also like to thank the Division of Epidemiology, Biostatistics, and Environmental Health of the School of Public Health at the University of Memphis for their financial support in the form of graduate assistantships and travel support. In particular, thank you to Dr. Wilfried Karmaus who made this possible.

Lastly, I would like to thank my family and friends who have been relentlessly supportive and encouraging. Without them, this would not have been possible. Even though I may be states away, please know I appreciate everything you all have done for me.

Meredith Ray

# Abstract

Functional neuroimaging is a relatively young discipline within the neurosciences that has led to significant advances in our understanding of the human brain and progress in neuroscientific research related to public health. Accurately identifying activated regions in the brain showing a strong association with an outcome of interest is crucial in terms of disease prediction and prevention. Functional magnetic resonance imaging (fMRI) is the most widely used method for this type of study as it has the ability to measure and identify the location of changes in tissue perfusion, blood oxygenation, and blood volume. In practice, the three-dimensional brain locations or coordinates of the local maximum of these changes are reported. By nature, fMRIs are noninvasive, slowly becoming more available, have relatively high spatiotemporal resolution, and have the remarkable ability to map the entire network of the brain's function during the thought process. However, due to their high costs, fMRI studies tend to have a very small number of participants, which cause inflated type II error and lack reproducibility. This gives rise to the need for fMRI meta analyzes, which combines studies in order to increase overall sample size and testing power. In this dissertation, two methods are proposed that aim to identify regions of brain activation using fMRI coordinate-based meta analysis; a spatial Cox process and a mixture of Dirichlet processes model.

The first method was motivated by the desire to identify significant regions of brain activation using fMRI coordinate-based meta data. To identify these regions we elected to implement a Bayesian spatial Cox process. We considered two levels of clustering, latent foci center and study activation center, utilizing the Dirichlet

process (DP) built into a spatial Cox process used to model the distribution of foci. Commonly used spatial clustering methods model the random variation of the intensity governed by a process such that peaks in these processes would relate to areas of elevated aggregation in the events. However, methods of this type all assume three-dimensional normality, which is inappropriate for fMRI due to the nature of brain functioning and brain structure, and can possibly cause misclassification of foci and increase error in prediction and estimation. We relax this normality assumption and model intensity as a function of distance between the focus and the center of the cluster of foci using Gaussian kernels and the foci center will be identified by the use of a Dirichlet process. Simulation studies were conducted to evaluate the sensitivity and robustness with respect to cluster identification and underlying data distributions. An additional application of the proposed method was applied to an fMRI meta data of emotion foci. Both simulations and real data application produced promising results that highlighted the ability to correctly cluster.

The second method was motivated by the spatial Cox process' inability to statistically distinguish between clusters via a limitation to the Dirichlet process. However, it still aimed to identify significant regions of brain activation. This method modeled the realization of the data as a linear association with the overall mean of the data and adjusts for some study effect. The mean of the data was modeled as a mixture of unknown finite number of components and adjusted for a study effect modeled as a Dirichlet process. Similarly, each component was modeled as a Dirichlet process. Conditional on the mean of the data and some study effect, the distribution of the random error is standard multivariate normal. By modeling the mean as a mixture of Dirichlet processes, this still allows the method flexibility in capturing irregular spatial patterns and relaxes the typical normality assumptions, but can also statistically distinguish between a cluster or a mode within a cluster. The Bayesian framework was again implemented to draw model inferences. Simulation studies were

conducted to explore the sensitivity and robustness of the method, but illustrated a mediocre ability to correctly identify clusters. As an additional application, we applied the proposed method to the same fMRI meta data as was done in the first proposed method. The number of clusters identified were significantly lower and cluster centers identified were not in close proximity to any of those identified in the first proposed method. Both simulation studies and real data applications indicate this second proposed method is not sensitive enough to correctly identify clusters.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Background

Functional neuroimaging is a relatively young discipline within the neurosciences that has led to significant advances in our understanding of the human brain and progress in neuroscientific research related to public health. Accurately identifying activated regions in the brain showing strong association with an outcome of interest is crucial in terms of disease prediction and prevention. Functional magnetic resonance imaging (fMRI) is the most widely used method for this type of study and has the ability to measure changes in tissue perfusion, blood oxygenation, and blood volume [Logothetis, 2008]. The amount and location of these changes are measured when subjects are under some situational environment or experiment that provokes a thought, emotion, or action. Specifically, these changes are measured using the blood oxygen level depend (BOLD) contrast method, which measures the change in blood flow in respect to the amount of energy used by the brain cells [Bandettini et al., 1992, Ogawa et al., 1990]. The resulting fMRI presents an image of the brain with a color-based BOLD signal scale and a set of X,Y,Z coordinates to represent local maximum BOLD contrasts.

Positron emission tomography (PET) scans are sometimes used in addition to fMRI. PET works by exposing the subject via intravenously to a positron-emitting radionuclide (tracer). The PET scanner is then able to measure activation by exposing the subject to low levels of radiation which is reflected by the tracer in the blood flow

of the brain. Similar to fMRI, PET scanners can produce an image and coordinates. Furthermore, these scan also have the ability to measure the brain in the process of thought over a given period of time. However, they are limited by how quickly the tracer is absorbed by the body which limits the stimulation, lower spatial and temporal resolution, and lower availability of machines Devlin et al. [2000], Ojemann et al. [1998]. Studies have been conducted to explicitly compare PET and fMRI using motor, perceptual, and higher-level cognitive activity with successful results while other studies were unable to reproduce results between the two methods Devlin et al. [2000].

Depending on the fMRI software, these results have been standardized to one of two space scales or brain atlases, Talairach or the Montreal Neurological Institute (MNI) templates [Laird et al., 2010]. The Talairach space, was originally developed by Jean Talairach and P. Tournoux under the assumption that all distances within the brain are proportional to the overall brain size. Based on a postmortem dissection of a single human brain, they defined their brain space with the overall dimensions (x=136 mm, y= 172 mm, z=118 mm) with the primary axis lying between the anterior commissure and posterior commissure and the origin being the anterior commissure [Talairach and Tournoux, 1988, Laird et al., 2010]. The MNI template was originally defined in two stages. In the first stage, anatomical landmarks were manually identified in 250 brains from healthy right-handed adults using MRI scans. This allowed the edges and orientation or primary axis to be defined for each brain. Next, all 250 brains were scaled to those equivalent landmarks in the Talairach space and thus resulting in the original 250 MNI brain atlas [Evans et al., 1992]. However, immediate advancements were made when an additional 55 subjects were mapped to the 250 MNI brain atlas by least-squares linear regression. These 55 and 250 were then averaged together to create the more commonly used 305 MNI brain atlas [Laird et al., 2010, Lancaster et al., 2007, Evans et al., 1993, Collins et al., 1994].

The Talairach and MNI spaces mentioned above are not directly comparable due to the different brain sizes and orientations [Laird et al., 2010, 2005]. Thus, the issue of transforming these coordinates to fit the other or to an overall standard scale is a necessity for meta-analyzes. Currently, the two most widely used transformations are the Brett transformation [Brett et al., 2002] and the Lancaster transformation [Lancaster et al., 2007]. The Brett transformation applies several non-linear transformations to different regions of the brain to match the MNI with the Talariach space [Brett et al., 2002]. The Lancaster transformation is a scaled transformation whose parameters were derived from 100 brains using least-squared error methods [Lancaster et al., 2007]. Studies have shown that the Lancaster transformation provides the least amount of disparity between MNI and Talairach coordinates [Laird et al., 2010, Lancaster et al., 2007]. These normalizations, or the scaling of an individual's coordinates to the respective space of MNI or Talairach, are most often done within the fMRI software.

By nature, fMRI's are noninvasive, slowly becoming more available, have relatively high spatiotempral resolution, and have the remarkable ability to map the entire network of the brain's function during the thought process [Logothetis, 2008]. However, it is limited in that it measures surrogate signal whose spatial specificity and temporal response are subject to both physical and biological constraintsİ and whose signal reflect neuronal mass activity [Logothetis, 2008]. In other words, brain activation is recorded before and after the expected stimulation and thus gives rise to noise and therefore must decipher which areas of activation, statistically greater than the noise [Smith, 2004]. Furthermore, due to the high cost, fMRI studies tend to have very small number of participants (less than 20) which cause inflated type II error (low power) and lack reproducibility [Thirion et al., 2007]. This gives rise to the need for fMRI meta analyzes to combine these studies to increase sample size and thus testing power [Fox et al., 1997, Kober et al., 2008, Eickhoff et al., 2009,

Salimi-Khorshidi et al., 2009, Kang et al., 2011].

In practice, peak activation coordinates and not full fMRI image data are often given. Thus, coordinate-based meta analyzes (CBMA) are configured by merging coordinates from groups of fMRI studies, given they are on the same atlas system [Salimi-Khorshidi et al., 2009] and measure the same type of stimulant response such as an emotional response. The analysis of CBMA is specific to this format and aims to identify areas within the brain that are statistically activated during a given stimulant. The identification of these areas can be performed using various methods but we aim to achieve this by exercising two forms of spatial clustering analysis, Cox process and finite mixture model, while using the Bayesian framework to estimate the appropriate parameters.

## 1.2 Existing Methods for Coordinate-based Meta-Analysis

The most common statistical methods for CBMA include: activation likelihood estimation (ALE), kernel density analysis (KDA), and spatial point process modeling. Additional research has been done with ALE and KDA to include the methods modified activation likelihood estimation (modALE) and multi-level kernel density analysis (MKDA).

### 1.2.1 Kernel Density Analysis

Kernel density analysis (KDA) was first implemented with fMRI data in Wager et al. [2003] and seeks to identify regions of significant brain activation. This method applies a smoothing kernel to each brain voxel. A voxel is a pre-specified block of the brain, i.e. 2 $mm^3$, that collectively make up the entire brain space. This smoothing kernel, with a pre-specified radius is applied to each voxel within the brain producing a smoothed histogram. This smoothed histogram reflects the estimated density of nearby reported activation locations within that voxel [Wager et al., 2003, 2004,

2007]. These estimated densities are then compared to a threshold for an indication of significance; testing against a null hypothesis. This threshold level is derived using Monte Carlo procedures by generating a number of permutated datasets of randomly drawn foci to create a set of statistical maps under the null hypothesis that the number of nearby peaks is equal or less than what is expected by chance. KDA searches for densities that can reject this and controls the chances of false positives at a rate of 5% [Wager et al., 2003, 2004, 2007]. This is done by recording the maximum KDA statistic for each Monte Carlo dataset and setting the threshold to the value that exceeds the maximum KDA statistic in 95% of the permutated datasets.

Wager et al. [2007] made improvements upon this original KDA method, termed multilevel kernel density analysis (MKDA), by letting the proportion of studies be the test statistic, and therefore the peaks are nested within the studies. MKDA creates an indicator map for each study where these maps indicate which voxels contain one or more foci within a radius. These maps are then averaged to provide a proportion of studies with foci within a given radius in a particular voxel [Salimi-Khorshidi et al., 2009, Wager et al., 2007]. This averaged map identifies statistical activations using a certain number of permutated datasets as described in the KDA except that these datasets are generated uniformly over the randomly selected indicated areas from the study indicator maps. Familywise error rate is applied as it was in KDA but by using the maximum proportion of activated contrasts [Wager et al., 2007].

The dominant drawbacks to these methods are the pre-specified voxel and radius bandwidth sizes and lack of any spatial model or component [Wager et al., 2007, Kang et al., 2011]. In the original KDA model, the threshold calculation was based upon the null distribution equally across the entire brain; treating each focus as spatially independent within and across studies. This drawback was sanctified in the revised MKDA method by incorporating the proportion of studies into the test statistic. Thus, the threshold and null distributions were based upon areas where activation

occurred and not uniformly across the entire brain [Wager et al., 2007].

### 1.2.2 Activation Likelihood Estimate

Similar to KDA, the activation likelihood estimate (ALE) also aims to identify areas of the brain that are significantly activated during given stimulants. ALE identifies these areas by again dividing the brain onto voxels but instead of implementing a smoothing kernel as in the previous method, implements a three-dimensional Gaussian function to estimate the density of reported activation locations within that voxel. The summation of these densities (union of probabilities) gives the resulting ALE value and can be interpreted as the probability of at least one activated foci laying within a given voxel. As in KDA, these probabilities are compared to a threshold to determine statistical significance or which union of probabilities exceeds random chance [Turkeltaub et al., 2002]. This threshold is derived in much of the same way as in KDA: that is, based on permutated datasets [Laird et al., 2005].

Eickhoff et al. [2009] and Eickhoff et al. [2012] pointed out limitations to the current ALE method such as the inability to address variances within the fMRI data itself and the calculation of the threshold. They proposed adjustments to the ALE calculation, termed modALE, that incorporated empirical estimations for between-subject and between-template variances from fMRI studies into the probability distribution and adjusted the Monte Carlo threshold to incorporate further familywise error rate and clustering inference (Eickhoff et al. [2009],Eickhoff et al. [2012]).

This new threshold proposed in Eickhoff et al. [2012], refutes the general null hypothesis that foci are uniformly throughout the brain. This is performed by first calculating the modelled activation (MA) map for each study or experiment in the fMRI data. This is calculated for each voxel by summing the Gaussian probability of each individual foci using in that voxel. The ALE value is then calculated by summing all of these MA maps. Intuitively, each brain voxel has an MA value thus produces

a map of MA values and therefore the name MA maps. All voxels expressing the same MA value are combined to a single histogram-bin. Once all voxels have been combined, this produces an entire histogram representing groups of voxels with the same MA values. This process is repeated for all studies or experiments and the histograms are continually merged to finally produce the null-distribution of ALE values under spatial independence. The p-value for each study or experimental ALE is the probability of observing that ALE value or one more extreme. The threshold for a particular ALE value is dependent upon the histogram-bin the value is located and all other bins more extreme to calculate the chance of observing this ALE value or a more extreme one [Eickhoff et al., 2012].

Even though this newly proposed method for correcting the family-wise error rate and cluster-level significance greatly improved the original ALE method, there are still two main drawbacks, specifically with the cluster-based thresholding. The first is that it produces low spatial specificity when clusters are large and the decision of the primary threshold can greatly affect the the significance and robustness of clusters making them appear larger [Woo et al., 2014].

### 1.2.3 Hierarchical Spatial Point Process

The other method, hierarchical spatial point process modeling, was introduced in Kang et al. [2011]. Unlike the previous two methods, it does not estimate the density distribution by voxels using kernel estimation. This article presents a hierarchical spatial point process model using Bayesian methods to estimate parameters that will identify regions of highly dense foci, which in turn will suggests regions of high activation across studies [Kang et al., 2011]. The model consists of three layers. The first layer models the individual foci, assumed independent, by clustering with an independent cluster process controlled by a random intensity function. At this level, two types of foci are considered: singly and multiply reported, which were assigned a

mark or indicator as to identify the specific types. The second level models the latent study activation center, again assumed independent and clustered by an independent cluster process and driven by a random intensity function. These two layers are both assumed conditional on the realisation of latent study activation centers or latent population center, respectively, and normally distributed. Also, both layers allow for individual foci and study centers to be singleton clusters and model this by an independent homogeneous Cox process, hence the random intensity functions. The third and final level models the latent population centers with a homogeneous Cox process controlled by a homogeneous random intensity. Posterior distributions are estimated using a spatial birth death processes nested within a MCMC simulation algorithm [Kang et al., 2011].

Kang noted that the number of population centers were sensitive to priors and that the concept of a spatial model could take practitioners time to adjust to from voxel-wise assessments such as those from KDA, MKDA, ALE, and the modALE. It can also be conceded that a potential limitation is the assumption of normality. This assumption forces the identified clusters to be spherical in nature.

This dissertation is built upon the model in Kang et al. [2011]. We consider two methods aiming to improve the flexibility of the methods by relaxing the normality assumption and identifying cluster of reaction regions with study effects adjusted. Both methods are in the Bayesian framework. In the next section, we discuss the general construction of Bayesian models.

## 1.3 Bayesian Methods

Bayesian methods arise from the need of inferences on observed data, $y$, conditional on some unknown parameter(s), $\theta$. Although these parameters are unknown, there is some prior information available [Congdon, 2007]. This prior information is presented as a density, $P(\theta)$ and therefore the likelihood or probability of the presented

data and its conditional parameters is presented as $P(y|\theta)$ or some specified model. Therefore, in order to make inferences on the posterior, $P(\theta|y)$, Bayes theorem can be implemented. Bayes theorem was defined by Thomas Bayes as

$$P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A|B)P(B)}{P(A)}.$$

Following the Bayes theorem, we have

$$P(\theta|y) = \frac{P(\theta \cap y)}{P(y)} = \frac{P(y|\theta)P(\theta)}{\int_\theta P(y|\theta)P(\theta)d\theta} \propto P(y|\theta)P(\theta). \tag{1.1}$$

The proportionality in (1.1) is due to the conditionality on $y$. $P(y|\theta)$ and $P(\theta)$ as the prior distribution, hence it's assumed prior density or distribution. Inferences on the posterior distribution can take several forms, but most often are in summary form such as means, variances, or medians [Congdon, 2007]. The derivation of these summaries statistics follow from their generic definitions, i.e.

$$E(\theta) = \int \theta P(\theta|y)d\theta \tag{1.2}$$

$$\begin{aligned} Var(\theta) &= \int \theta^2 P(\theta|y)d\theta - [E(\theta|y)]^2 \\ &= [E(\theta^2|y)] - [E(\theta|y)]^2. \end{aligned} \tag{1.3}$$

These summary statistics offer useful insights into the nature of posterior distribution; the posterior mean indicates the central tendency of the distribution, posterior variance offers the overall spread of the distribution, and posterior median can also indicate central tendency [Congdon, 2007]. Credible intervals, 100(1- $\alpha$)% can also be calculated for $\theta$ which is interpreted that there is a 1-$\alpha$% probability that $\theta$ lies within the interval $[lower, upper]$ [Congdon, 2007]. *Lower* and *upper* are calculated by $\frac{\alpha}{2}$ and $100 - \frac{\alpha}{2}$ quantiles of the posterior density [Congdon, 2007]. Another credible interval is the 100(1-$\alpha$)% highest probability density (HPD) interval, thus the density for each point within the interval exceeds that for every point outside of the interval. It immediately follows that this is also the shortest possible 100(1-$\alpha$)% credible interval [Congdon, 2007].

If the posterior distribution takes an analytical form, the summary statistics are rather straightforwardly calculated from their pre-derived formulas. However, if the posterior is not a standard distribution, the integrals can be difficult to calculate [Congdon, 2007]. This directly illustrates the link between Bayesian inference and sampling-based estimation methods. More general, Markov chain Monte Carlo (MCMC) methods are a set of algorithms that simulate the stochastic process of posterior densities [Geyer, 1992]. These methods propose various ways of sampling from the posterior distribution. From the theory of Strong Law of Large Numbers, repeated sampling will eventually converge and allow for an empirical estimate of these summary statistics.

The most basic MCMC algorithm used to simulate a Markov chain with a stationary posterior distribution is referred to as the Metropolis-Hastings (MH) algorithm [Hastings, 1970, Metropolis et al., 1953]. The sampling is conducted at each iteration of the chain over a pre-specified total number of iterations, $T$. The current value is denoted as $\theta^{(t)}$ with $P(\theta|y)$ as its stationary distribution [Congdon, 2007]. The chain is updated from its current value, $\theta^{(t)}$, to $\theta^*$ with the probability:

$$\alpha(\theta^*|\theta^{(t)}) = min(1, \frac{P(\theta^*|y)f(\theta^{(t)}|\theta^*)}{P(\theta^{(t)}|y)f(\theta^*|\theta^{(t)})}), \tag{1.4}$$

where $f()$ is a jumping density [Chib and Greenberg, 1995]. The jumping density is the probability of moving back and forth between $\theta^*$ and $\theta^{(t)}$; specifically, $f(\theta^*|\theta^{(t)})$ is the probability of moving to $\theta^*$ from the center, $\theta^{(t)}$ and $f(\theta^{(t)}|\theta^*)$ is the probability of moving back to $\theta^{(t)}$ from $\theta^*$. If $\theta^*$ is accepted then $\theta^{(t+1)} = \theta^*$ else $\theta^{(t+1)} = \theta^{(t)}$. If the jumping distribution is symmetric (i.e. normal), the acceptance probability reduces to:

$$\alpha(\theta^*|\theta^{(t)}) = min(1, \frac{P(\theta^*|y)}{P(\theta^{(t)}|y)}), \tag{1.5}$$

which is known as random walk MH [Gelman et al.]. Another scheme for sampling allows the proposal or jumping distribution to be independent of $\theta^{(t)}$, $f(\theta^*)$, which

is known as independence sampler [Congdon, 2007]. The rate at which the proposal value, $\theta^*$, is accepted indicates the proximity of the proposal and current values. If acceptance is too high or too low, the variance of the proposal density may need to be adjusted [Congdon, 2007].

This sampling scheme can be extended to more than one parameter when $\theta$ is multidimensional. Although all parameters can be updated simultaneously, it is much easier and simpler to update one at a time while holding all other parameters constant at their current values. For example, let $\theta_{[j]} = (\theta_1, \theta_2, ..., \theta_{j-1}, \theta_{j+1}, ..., \theta_D)$ all $D$ number of parameters excluding the $j^{th}$ parameter and $\theta_j^{(t)}$ denote the value of $\theta_j$ at the $t$ iteration [Congdon, 2007]. Since each parameter is updated individually at the $t$ iteration, when updating the $j^{th}$ parameter then all preceding parameters, $j-1$ are already updated and thus denoted $\theta_{[j]}^{(t,t+1)} = (\theta_1^{(t+1)}, \theta_2^{(t+1)}, ..., \theta_{j-1}^{(t+1)}, \theta_{j+1}^{(t)}, ..., \theta_D^{(t)})$. The acceptance probability of selecting the candidate value, $\theta_j^{(t)}$, is defined as:

$$\alpha(\theta_j^*, \theta_j^{(t)}, \theta_{[j]}^{(t,t+1)}) = min[1, \frac{P(\theta_j^*|\theta_{[j]}^{(t,t+1)}) f(\theta_j^{(t)}|\theta_j^*, \theta_{[j]}^{(t,t+1)})}{P(\theta_j^{(t)}|\theta_{[j]}^{(t,t+1)}) f(\theta_j^*|\theta_j^{(t)}, \theta_{[j]}^{(t,t+1)})}]. \tag{1.6}$$

A second algorithm, a special case of the MH algorithm specifically for a multidimensional parameter space, is called Gibbs sampler [Gelfand and Smith, 1990, Congdon, 2007] originally developed by Geman and Geman [1984]. In the Gibbs sampler, the proposal density equals the full conditional, $P(\theta_j^*|\theta_{[j]})$, and therefore the acceptance probably reduces to 1 and thus the candidate values are always accepted [Congdon, 2007]. Each parameter is again updated one at a time, conditional on all other parameters. Letting $\theta = (\theta_1, \theta_2, ..., \theta_D)$, the sampling distribution for each

parameter is:

$$\begin{aligned}
\theta_1^{(t+1)} &\sim f(\theta_1|\theta_2^{(t)}, \theta_3^{(t)}, \ldots, \theta_D^{(t)}) \\
\theta_2^{(t+1)} &\sim f(\theta_2|\theta_1^{(t+1)}, \theta_3^{(t)}, \ldots, \theta_D^{(t)}) \\
&\vdots \\
\theta_D^{(t+1)} &\sim f(\theta_D|\theta_1^{(t+1)}, \theta_2^{(t+1)}, \ldots, \theta_{D-1}^{(t+1)}).
\end{aligned} \tag{1.7}$$

By repeatedly sampling in this fashion, the initial values for $\theta$ are irrelevant which gives a "memoryless" property in addition to converging to a stationary sampling distribution [Congdon, 2007]. These full conditional sampling distributions are obtainable by implementing Bayes theorem from the full joint distribution (full model distribution equivalently proportional to the likelihood time prior density) while handling all other parameters other than $\theta_j$ as constant. In order for the Gibbs sampler to be convenient and efficient, these full conditional distribution generally produce a known or standard distribution making sampling effortless [Congdon, 2007]. If these produce non-standard distributions, MH is generally used.

One practice that is used across all MCMC algorithms to assist with calculations of summary statistics is the removal of the first $B$ iterations, or burn-on iterations. As mentioned above, these MCMC algorithms have a "memoryless" property that in the convergence of the sampling distribution, the initial values are forgotten. Thus, the initial and other beginning iterations do not need to be considered when making or calculating inferences. This bears no difference in the resulting convergence.

Convergence of sampling distributions with non-standard forms for MCMC methods is pertinent to achieve approximate inferences. The assessment of convergence many unanswered questions [Congdon, 2007]. Therefore, for this dissertation, convergence was checked visually and by acceptance rates. Visual inspection requires plotting the chain of some parameter. If convergence is met, the chain will result in

a relatively linear fashion but is at discretion of the observer. Optimal acceptance rates are around 37% [Blum, 2010].

## 1.4 Existing Methods for Clustering Analysis

Since the focus of the dissertation is on clustering, in this section we review clustering methods, supervised and unsupervised methods.

Methods for clustering have become increasing popular across various fields of study. They are often used in disciplines such as data mining, document retrieval, image segmentation and pattern classification [Filippone et al., 2008]. The general aim of clustering methods is to create partitions, groups, or clusters based on the similarity (or dissimilarity) of a specified criteria [Filippone et al., 2008]. The similarity criteria and similarity measure is defined by the particular method. Some methods derive better cluster qualities depending on the type of data clustered, categorical, continuous, ordinal and binary [Zaït and Messatfa, 1997]. Clustering methods can be divided into four distinct categories of clustering methods: hierarchical clustering, partitioning clustering, density-based clustering, model-based clustering, and Bayesian clustering.

### 1.4.1 Hierarchical Clustering

Hierarchical clustering is probably considered the most popular of the four categories with 6,723 articles related to "hierarchical clustering" via a PubMed search and aims at clustering subjects or some variable. In general, this method clusters by levels where at the initial level, all observations desired to be clustered are considered to be in a cluster of their own. At each subsequent level, observations or groups of observations are clustered based on their similarity to each other. The clustering continues until all observations are clustered into a single cluster. Different methods use different similarity measures to cluster and to decipher which level produces the optimal

clustering [Milligan and Cooper, 1987]. The distance between two observations or clusters is most often used as similarity measure. This is based on the underlying assumption that the closer two objects are the more likely the two are similar. The agglomerative approach just described is not the only way hierarchical clustering is achieved. Other methods of hierarchical clustering using a divisive approach where the initial level clusters all observations into a single cluster. At each subsequent level clusters divide until all observations are single separate clusters. Regardless of the agglomerative or divisive approach, clusters never overlap and once observations or groups are clustered or unclustered, they remain so. These are often depicted visually in a dendrogram. Specific examples of the different methods for this clustering category include: single linkage, complete linkage, average linkage, centroid method, Ward's method, two-stage density and the Kth nearest neighbor [Zaït and Messatfa, 1997]. The methods differ in the derivation of their similarity criteria. For example, in single-linkage clustering, the two clusters with the smallest minimum pairwise Euclidean distance are grouped for that level while in complete linkage, the two clusters with the smallest maximum pairwise Euclidean distance are grouped for that level. Although Euclidean is the most popular, other distance measures may be used as well. These methods come with an ease of application but results may not always be easily inferred and potentially requires large CPU time and memory space [Zaït and Messatfa, 1997].

### 1.4.2 Partitioning Clustering

Similar to hierarchical clustering, this method aims to group subjects or some variable of interest. However, partition clustering or sometimes called centroid-based clustering differs in that it partitions in a single step rather than repetitive levels of sub-partitions as in the hierarchical clustering [Filippone et al., 2008]. This non-hierarchical clustering method also requires less CPU time and memory. The

partitions are achieved by the optimization of an appropriate objective function [Filippone et al., 2008]. Perhaps the most common partitioning method is K-means [MacQueen et al., 1967, Pollard, 1982, Lloyd, 1982] but also include relational data analysis (RDA), Autoclass, Fuzzy c-Means, self-organizing maps (SOM or Kohonen maps) and Neural Gas to name a few [Zaït and Messatfa, 1997, Filippone et al., 2008]. The general algorithm for partition clustering take the following steps:

1. Determine the number of clusters if not prespecified.
2. Initialize the cluster centers.
3. Compute partitioning for data.
4. Compute (update) cluster centers.
5. If the partitioning is unchanged (or the algorithm has converged), stop; otherwise, return to step 3.

If the number of clusters is unknown, the partitive algorithm can be repeated for a different set of number of clusters typically from two to $\sqrt{(}N)$ where $N$ is the number of samples in the dataset [Vesanto and Alhoniemi, 2000]. For illustration purposes, an example of the error function in K-means that is minimized is:

$$E = \sum_{k=1}^{C} \sum_{x \in Q_k} ||x - c_k||^2$$

where $C$ is the number of clusters and $c_k$ is the center of cluster $k$ [Vesanto and Alhoniemi, 2000]. To choose between different partitioning, an validity index may be calculated. Several have been suggested in Bezdek and Pal [1998] and Milligan and Cooper [1985] but for K-means the Davies-Bouldin index [Davies and Bouldin, 1979] is most appropriate because of its low values that indicate good clustering for spherical clusters [Vesanto and Alhoniemi, 2000]. The Davies-Bouldin index is calculated:

$$\frac{1}{C} \sum_{k=1}^{C} \left\{ \frac{S_c(Q_k) + S_c(Q_l)}{d_{ce}(Q_k, Q_l)} \right\}$$

where $S_c$ represents within cluster distance, $d_{ce}$ for between clusters distance and $C$ is the number of clusters. Although partitioned clusters can be computationally quicker, they are limited by generally requiring the number of clusters to be pre-specified and assume a spherical form; K-means clusters by searching for spheres [Vesanto and Alhoniemi, 2000].

### 1.4.3 Density-based Clustering

Density-based clustering imitates the steps of partition clustering but with the exception that the number of clusters does not have to be specified and the distance between two groups or "measure of connectivity" is only considered if their density (most often estimated by kernels), $P(x)$, is above a pre-specified threshold $\lambda$ [Kriegel et al., 2011]. Generally speaking, if two groups or observations have relatively high densities and are in close proximity to each other, they will be clustered together. Those data with densities below $\lambda$ are considered "noise" and therefore no need to measure their "connectivity" with other data [Ester et al., 1996, Kriegel et al., 2011]. Thus, as with partition clustering, the different methods of density-based clustering address the different manners in which the threshold and distance or connectivity are defined. The most popular method is density-based spatial clustering of applications with noise or DBSCAN introduced in Ester et al. [1996]. BDSCAN groups a data point $p$ with another data point $q$ if $p$ is within a certain distance, $\varepsilon$, from $q$ and if $q$ is surrounded by a certain number of other data points. Thus, the BDSCAN has two parameters that must be defined by the user, the minimal distance $\varepsilon$ and the minimal number of point required for a dense region, $minPt$. The distance can typically be estimated using a K-distance graph but the parameter $minPt$ is set by the user. The advantage to this method of clustering is the flexibility of the shape of the clusters as there are no restrictions on the density distribution and that the number of clusters is data-driven and not pre-specified. The disadvantages is the parameter

pre-specification from the user, points that lie on the edge of clusters can often be grouped into one cluster or another, and DBSCAN has a difficult time clustering data with large differences in densities [Ester et al., 1996, Kriegel et al., 2011].

### 1.4.4 Model-based Clustering

Model-based clustering or sometimes known as probability models model data under the assumption that it follows a specific distribution. The different methods of model-based clustering arise from the various ways to model or implement these distributions as well as the form the models are elected to take. More specifically, the estimation of the parameters via frequentist or Bayesian approach lead to various methods. The main advantage to these models are the flexibility they provide and allow them to be implemented in numerous disciplines and applications such as character recognition [Murtagh and Raftery, 1984], tissue segmentation [Banfield and Raftery, 1993], minefield and seismic fault detection [Dasgupta and Raftery, 1998], identification of textile flaws from images [Campbell et al., 1997], and classification of astronomical data [Celeux and Govaert, 1995].

Model-based clustering or probability models can be divided in to roughly 5 classifications depending on the nature of the data. The classifications are partition-type models for data vectors, partition-type models for dissimilarity data, partition-type models for random similarity relations and random graphs, testing for homogeneity and for a clustering structure, and probabilistic models for hierarchical and tree-like classifications [Bock, 1996]. Given the data this dissertation explores, we will only review partition-type models for data vectors which can be defined as mixture models or point cluster processes [Bock, 1996]. Other partition-type models such as the fixed-classification model, multi-modality and high-density (density-contour) clusters, and mode clusters are summarized in Bock [1996].

#### 1.4.4.1 Clustering-based on Mixture Models

The finite mixture models assume that given the data $\mathbf{y} = (y_1, ..., y_n)$ there exist $M$ number of partitions or clusters in which this data may be grouped. Each cluster shares a known parametric family $f(\cdot; \theta)$ and therefore also contains specific family parameters $\underline{\theta} = (\theta_1, ..., \theta_m)$ such that the resulting likelihood is:

$$L(\theta; \mathbf{y}) = \prod_{i=1}^{n} \sum_{k=1}^{M} \pi_k f_k(y_i|\theta_k)$$

where $f(\cdot)$ is the density function and its respective parameter $\theta$ of the $k^{th}$ cluster and $\pi_k$ is the mixing proportion or probability that an observation is assigned to the $k^{th}$ cluster ($\pi = (\pi_1, ..., \pi_m); \sum_M^{k=1} \pi_k = 1$) [Bock, 1996, Duda and Hart, 1973, Binder, 1978, Scott and Symons, 1971]. Mixture parameters may be estimated in a number of ways with the two most popular methods being the expectation maximization (EM) algorithm and Markov chain Monte Carlo.

The expectation maximization (EM) algorithm was originally introduced in Dempster et al. [1977] and utilizes the maximum likelihood to estimate parameters. It treats the individual data clustering assignment as a missing or latent variable $z$ such that $\mathbf{z} = (z_1, ..., z_n)$ [Dempster et al., 1977]. The EM algorithm works in two steps, "E-step" and "M-step". During the "E-step" the conditional expectation of the complete-data log-likelihood given the observed data and current parameter estimates is calculated. The complete-data refers to the observed data and its latent variable and can be notated as $x_i = (y_i, z_i)$. The log-likelihood for the complete data is

$$l(\theta, \pi, \mathbf{z}|\mathbf{x}) = \sum_{i=1}^{n} \sum_{k=1}^{M} log(\pi_k f(y_i|\theta_k))$$

with $\mathbf{z}$ being estimated by the expectation of

$$Q(\theta, \pi|\theta^{(t)}, \pi^{(t)}) = E\left[\sum_{i=1}^{n} \sum_{k=1}^{M} log(\pi_k f(y_i|\theta_k))\right].$$

During the "M-step" the log-likelihood is maximized in terms of $\pi$ and $\theta$ with $\mathbf{z}$ held constant at the values calculated in the "E-step" as illustrated below

$$\theta^{(\mathbf{t+1})} = \underset{\theta}{\operatorname{argmin}}\, Q(\theta, \pi|\theta^{(t)}, \pi^{(t)})$$
$$\pi^{(\mathbf{t+1})} = \underset{\pi}{\operatorname{argmin}}\, Q(\theta, \pi|\theta^{(t)}, \pi^{(t)}).$$

Some general limitations with EM are its potentially slow numerical convergence and the convergence to the maximum likelihood may not always be the global maximum [Wu, 1983].

Another method to estimate mixture and model parameters is by sampling from their posterior distribution calculated using the Bayes' therom discussed earlier. One of the most popular ways to achieve this is through the Birth-Death process. The Birth-death process was originally introduced in Preston [1975] and is continual-time Markov process. During the MC chain, "births" and "deaths" occur at certain time points that jump throughout the chain that allow the number of components to increase by one or decrease by one, respectively [Moller and Waagepetersen, 2004]. Births and deaths are binominally modeled with non-negative rates, birth rate $\beta(x)$ and a death rate $\delta(x)$, to randomly select if a birth or a death will at that specific time point. This jumping time point is typically modeled with the exponential distribution [Moller and Waagepetersen, 2004]. Similar to the EM algorithm, most Birth-death algorithms incorporate a latent variable, $z$, to model the clustering assignment for each observations such that $z = (z_1, ..., z_n)$. Once the selection of a birth or death has been calculated, all individual observation must be assigned to a cluster using the multinominally probabilities from the mixture proportions and all mixture parameters must be updated. Non-mixture parameters, such as $\theta$, are sampled from their conditional posterior distributions at every iteration of the Markov chain Monte Carlo (MCMC) simulations. Several different algorithms to implement the Birth-Death process can be found in the literature [Preston, 1975, Moller and Waagepetersen, 2004]. These algorithms differ in how the birth rate, death rate, and

jumping rates are defined as well as the algorithm implementation of Birth-Death. For this dissertation, we elect to use the algorithm defined in Stephens [2000] and provide algorithm details in Chapter 3. The primary advantage to this type of process is not only are we able to select the best model but are able to use the results and information form over models such as the exchangeability of the mixture components throughout the MC chain [Stephens, 2000].

Another method of modeling clusters with mixtures is through the Dirichlet process (DP). Originally discussed in [Antoniak, 1974, Ferguson, 1973], the DP is a simpler and more elegant way to model latent classes that can explain dependencies between observations [Neal, 2000]. The DP is a mixture of probability distributions and thus when assigned as the distribution of a parameter it becomes a distribution of probability distributions. Its this discreetness property that allows for the partition of probabilities and thus clustering. The general DP can be applied in the form

$$\begin{aligned} y|\theta_i &\sim F(\theta_i) \\ \theta_i &\sim G \\ G &\sim DP(G_0, \alpha) \end{aligned}$$

where $\theta_i \in (\theta_1, ..., \theta_n)$ is the parameter to be estimated. Please note that the notation $\sim$ means "distributed as". It is assumed that the observed data $y$ are model as some probability model $F(\theta)$, and $G$ is the prior distribution assigned to $\theta_i$ and $G$ is assigned as a DP with a base distribution $G_0$ and precision parameter $\alpha$. The DP is based on assumptions of discreteness, independency, and exchangeability meaning its is assumed that the distribution is from a mixture of distributions, each parameter $\theta_i$ is independent and exchangeable (order of the data is of not matter) [Ferguson, 1973, Antoniak, 1974, Neal, 2000]. The DP itself is defined as

$$P(\theta_i|\theta^{(i)}) \sim \frac{\alpha G_0}{\alpha + n - 1} + \frac{\sum_{j=1, j\neq i}^{n} \delta_{\theta_j}(\theta_i)}{\alpha + n - 1},$$

where $\theta^{(i)}$ denotes all parameters excluding $\theta_i, \theta^{(i)} = (\theta_1, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_n)$, and $\delta_{\theta_j}(\theta_i)$ denotes the unit point mass at $\theta = \theta_i$ or the distribution concentrate at the single point $\theta$ [Escobar and West, 1995, Neal, 2000]. Without the assumption of independency and exchangeability this probability could not be calculated.

Following the Bayesian framework, in order to sample values for $\theta_i$ we need to sample from the conditional posterior distribution, which is proportional to $P(y|\theta_i)P(\theta_i)$. Depending on the probability model assigned to $y$ the prior for $\theta_i$ may be a conjugate or non-conjugate prior. Models with conjugate priors are easily implemented via Gibbs sampler but non-conjugate priors cause Gibbs sampler to have complex and potentially impossible numerical integration [Neal, 2000]. Some have attempted to provide methods to approximate these complex numerical integrations such as West who implemented Monte Carlo approximation but produced a large error or MacEachern and Müller [1998] who attempted to handle non-conjugate priors using an exact approach using a mapping from a set of auxiliary parameters. Their algorithms to implement these proved inefficient [Neal, 2000]. Walker and Damien [1998] used these same algorithms with different auxiliary parameters but proved unsuitable in a general sense [Neal, 2000]. Other existing methods for handling non-conjugate priors are summarized in Neal [2000]. However, also presented in Neal [2000] is a sampling algorithm similar to that of MacEachern and Müller [1998] with the incorporation of auxiliary parameters. This algorithm differs in that the auxiliary parameter are temporary and therefore allows more flexibility [Neal, 2000]. We elect to implement this algorithm in both of our methods and is described in detail in Chapter 2 section 2.2.2.

#### 1.4.4.2 Point Processes

In this dissertation, the Birth-Death process noted in section 1.4.4.1 is implemented in the context of point process. In this section, we provide a brief introduction on

this topic.

Point processes are a type of spatial statistics which can be very useful for identifying clusters when spatial information is available. They model data (spatial points) as a random element whose values are "point patterns" on a set $S$. This random process is for point data, $X_1, X_2, ...X_n$ whose realizations are a random locally finite subset of a space $S$, assuming $S \subset R^p$ [Bock, 1996, Moller and Waagepetersen, 2004]. For any subset $x \subseteq S$, let $n(x)$ denote the cardinality of $x$ and when the point configuration $x$ is restricted to B where $B \subseteq S$ is bounded, denoted $x_B = x \cap B$, then $X$ takes the values in the space defined by $N = \{x \subseteq S : n(x_B < \infty$ for all bounded $B \subseteq S)\{$ [Bock, 1996, Moller and Waagepetersen, 2004]. Point process methods include marked point process, poisson point process, marked poisson point process, Cox process and Markov point process [Moller and Waagepetersen, 2004]. A primary concern with these types of models are with the estimation of their associated parameters [Bock, 1996]. We elect to use a spatial Cox process in one of the methods presented in this body of work. More details are provided in Chapter 2.

#### 1.4.4.3 Goodness-of-fit

Model-based clustering attempts to fit pre-specified probability model to observed data. It is well known that models with larger number of parameters may often over fit the data and therefore raises the issue of how to determine if and how well a model fits the data. The model and its respective clusters can been chosen based on a any number of criteria. For this dissertation, we will only look at the three most popular, Bayes information criterion (BIC) [Schwarz et al., 1978, Raftery, 1995, 1999], Akaike information criterion (AIC) [Akaike, 1987], and deviance information criterion (DIC) [Spiegelhalter et al., 2002].

The Bayes information criteria is a criteria for model selection that is a function the likelihood while adjusting for the number of parameters estimated. The BIC has

the following formula for data $y$ and model $m$:

$$BIC = log\left[p(y|\theta_m)\right] - \left(\frac{d}{2}\right) log\left[n\right]$$

where $p(y|\theta_m)$ is the likelihood of the observed data given the parameters in model $m$, $d$ is the number of estimable parameters and $n$ is the number of observations. This criteria penalizes the likelihood for the model's complexity (greater number of estimable parameters) according to the log of the sample size [Congdon, 2007, Raftery, 1995]. Similarly, AIC was constructed to again penalize the likelihood for the model's complexity but arguable does not penalize to the extent of BIC as it does not incorporate sample size

$$AIC = 2d - 2log\left[p(y|\theta_m)\right].$$

The DIC criterion of Spiegelhalter et al. [2002] is a generalization of BIC and AIC and is particular useful for models with parameters estimated using the Bayesian paradigm. It estimates the expected deviance while adjusting for the model's complexity. Mathematically, the DIC is defined as

$$DIC = 2\overline{D} - D(\overline{\theta}|y),$$

where

$$\begin{aligned} \overline{D} &= \frac{1}{T}\sum_{t=1}^{T} D(y, \theta^{(t)}) \\ D(y, \theta^{(t)}) &= -2logP(y|\theta^{(t)}) \\ D(\overline{\theta}|y) &= D(y, \overline{\theta}). \end{aligned}$$

In the above, $\theta^{(t)}$ is the parameter estimate(s) at current time $t$, $\overline{\theta}$ is mean or median of the parameter estimate(s) for times $t = 1, ..., T$ and $y$ is the data [Congdon, 2007].For all three methods, the smaller the criteria the better the model fits the data. The advantage of DIC over BIC and AIC, particularly for the Bayesian setting, is its easy of calculation while the other two require calculating the likelihood at its maximum over $\theta$, which is not readily available from the MCMC simulation.

### 1.4.5 Bayesian Clustering

Bayesian clustering methods are typically implemented via computer programs. They are also the dominant method for clustering genetic data and other molecular data [Chen et al., 2007]. Specifically, they aim to identify groups of individuals for inferences at the population level with out assuming predefined populations [Chen et al., 2007]. The most popular Bayesian clustering programs which are built for genetic data are STRUCTURE [Falush et al., 2003, Pritchard et al., 2003, 2000], PARTITION [Dawson and Belkhir, 2001], BAPS [Corander et al., 2003, 2004, Corrander et al., 2006], GENELAND [Guillot et al., 2005], GENECLUST [François et al., 2006], and TESS [Chen et al., 2007]. The first three methods, STRUCTURE, PARTITION, and BAPS are non-spatial methods that cluster by minimizing Hardy-Weinberg and linkage disequilibria. Each individual observation is then assigned to a respective cluster probabilistically [Chen et al., 2007]. These methods perform best with genetic differential is relatively low [Latch et al., 2006]. The other three methods incorporate the spatial information of the data by incorporating the geographical coordinates of individuals in their prior distributions [Chen et al., 2007, Guillot et al., 2005, François et al., 2006, Wasser et al., 2004]. GENELAND is specific for populations within a designed study area and implements a hidden partition model [Chen et al., 2007]. GENECLUST and TESS are almost identical in that they both exercise the concept of Hidden Markov Random Field (HMRF) to model the spatial dependency [Chen et al., 2007]. TESS differs in that it's program allows for more algorithm options like data structures, proposal kernels, and other numerical options [Chen et al., 2007]. When these three methods were compared with STRUCTURE in Chen et al. [2007], they proved just as efficient in certain criteria. TESS was most appropriate when identifying number of populations, correct assignment to populations when there was moderate geographical admixture, and identifying recent migrants. STRUCTURE proved superior for correction assignment to populations when there was high geo-

graphical admixture and at detecting clinal variation. GENELAND and TESS performed equally at correctly assigning to populations when there was no geographical admixture. STRUCTURE and TESS had equal computation speeds [Chen et al., 2007].

## 1.5 Outline

The remainder of this dissertation is laid out as three chapters. The following chapter, Chapter 2, outlines and describes the first method, a spatial Cox point process model. This model aims to identify regions of activation within the brain using fMRI meta-data. This was performed by clustering on two levels, latent foci center and study activation center, with a spatial Cox point process utilizing the Dirichlet process to describe the distribution of foci. Intensity was modeled as a function of distance between the focus and the center of the cluster of foci using Gaussian kernels. All parameters were estimated using Bayesian methods. Simulations are conducted to demonstrate and asses the method. A real meta-analysis dataset was also explored and the results and conclusions are discussed.

Chapter 3 presents the second method, a mixture model. The distribution of the foci is assumed to be conditional, multivariate normal with a mean being the mixture of a study effect and individual foci effect. The study effect was assumed to follow a distribution generated from a Dirichlet process and the individual foci effect assumed a mixture of clusters, where the foci in each cluster was also assumed to follow a distribution generated from a Dirichlet process. The Bayesian paradigm was once again employed. As in Chapter 2, besides the presentation of the method, we conduct intensive simulations and perform real data applications.

Overall conclusions and discussions on the two methods presented appear in Chapter 4. It is here that limitations and further research directions are mentioned.

# Chapter 2

# Spatial Cox Process

This chapter presents a spatial Cox point process model with an intensity function modeled using a Gaussian Kernel that aims to identify regions of activation within the brain using fMRI meta-analysis data.

## 2.1 Introduction

Point processes, specifically Cox point processes, are often used for the determination of spatial patterns as they served as the "tractable model class" for spatial randomness [Moller and Waagepetersen, 2004]. The general spatial point process $X$ can be defined by a random countable subset of a space S, such that $S \subseteq \Re^3$ and the realization of $X$ is restricted to locally finite subsets of S. Furthermore, for any subset $s \subseteq S$ let $n(x)$ denote the cardinality of $x$ and is locally finite, $n(x_W) < \infty$, whenever $W \subseteq S$ is bounded, and where $x_W = x \cap W$ [Moller and Waagepetersen, 2004]. This naturally extends to a spatial Poisson point process which is defined by an intensity function $\varphi : S \to [0, \infty)$ that is locally integrable, $\int_W \varphi(\xi) d\xi < \infty$ for all bounded $W \subseteq S$. This calculated integration is called an intensity measure $\mu$ defined as $\mu(W) = \int_W \varphi(\xi) d\xi, W \subseteq S$. When the intensity function is defined as a realisation of a random field, the result is a doubly stochastic Poisson process called Cox processes [Moller and Waagepetersen, 2004, Cox, 1995]. For notation purpose, suppose $Z = Z(\xi) : \xi \in S$ be a nonnegative random field with probability one, $\xi \to Z(\xi)$ is locally integrable function. The density of a Cox process $X$ restricted

to a set $W \subseteq S$ with $|W| < \infty$ is defined as:

$$f(x) = exp\left(|W| - \int_W Z(\xi)d\xi\right) \prod_{\xi \in x} Z(\xi).$$

Due to the process' ability to model and cluster within spatial randomness, we can directly apply this model to any spatial data but in particular, coordinate-based meta-analysis (CBMA) data. The rest of this chapter is designated to explore the implementation of this process to coordinate-based meta-analysis brain data. The next section will explicitly layout the format of the model and discuss the spatial kernel and Bayesian framework. Following, various simulated dataset assess the sensitivity and robustness of the proposed method and the results. Section 2.4 implements a real meta-analysis dataset and discusses the results and implications. Section 2.5 is included to present conclusions and a discussion of the proposed method.

## 2.2 The Model

Let $s_{ij} = (x, y, z)$ denote a single focus which represents a talairach coordinate, for study $i$, $i = 1, ..., I$, and the $j^{th}$ foci in study $i$, $j = 1, ..., J_i$. We have $\sum_{i=1}^{I} J_i = n$, where $n$ is the total number of observed foci. Denoted by $\boldsymbol{s} = s_{1,1}...s_{I,J_I}$ represents all foci in the CBMA study. Continuing, a study effect must be considered to adjust for the differences or similarities between studies, denoted $p_i$ for each study $i$, while letting $\theta_{ij}$ represent each individual foci effect for the $j^{th}$ foci in study $i$.

The N-dimensional Gaussian kernel takes the form:

$$K(x) = \frac{1}{(\sqrt{2\pi}\sigma)^N} exp\left(\frac{||x||^2}{2\sigma^2}\right).$$

Similar to the standard deviation in a general Gaussian or normal distribution, $\sigma$ controls the width of the kernel and by nature always positive. The first term, $\int_{-\infty}^{\infty} exp^{\frac{-x^2}{2\sigma^2}} dx = \sqrt{2\pi}\sigma$, is called the normalizing constant which ensures that the area under the curve remains unity as the kernel width, $\sigma$, changes.

To describe the distribution pattern of $\boldsymbol{s}$, we follow the suggestion by Kang et al. [2011] via a Cox point process,

$$f(\boldsymbol{s}) = f[\lambda(s)] \prod_{s_{ij} \in \boldsymbol{s}} \lambda(s_{ij}),$$

where

$$f[\lambda(s)] = exp\left(|B| - \mu(B)\right) \propto exp\left(-\int_B \lambda(s) ds\right),$$

and $B$ represents the brain space, and $\lambda(s)$ is the intensity at focus $s$. To model the intensity $\lambda(s)$, we use Gaussian kernels aiming to gain flexibility. We thus have for focus $s_{ij}$,

$$\begin{aligned} f[\lambda(s)] &\propto exp(-\int_B exp(a_{ij} K(s_{ij} - p_i - \theta_{ij})) ds), \\ \prod_{s_{ij} \in \boldsymbol{s}} \lambda(s_{ij}) &= \prod_{s_{ij} \in \boldsymbol{s}} exp(a_{ij} K(s_{ij} - p_i - \theta_{ij})), \\ K(s_{ij} - p_i - \theta_{ij}) &= exp\left\{-\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho}\right\}, \end{aligned}$$

where $a_{ij}$ is individual cluster effect, $p_i$ is study effect for study $i$ and $\theta_{ij}$ is the individual cluster center for the $j^{th}$ foci in study $i$. Function $K(\cdot)$ is a Gaussian kernel with regulation parameter $\rho$. A Gaussian kernel function is in similar manners as exponential and Laplacian kernels [Shawe-Taylor and Cristianini, 2004]. This leads to

$$\begin{aligned} f(\boldsymbol{s}) &= exp\left[-\int_B exp\left(a_{ij} exp\left\{-\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho}\right\}\right) ds_{ij}\right] \\ &\quad \times \prod_{s_{ij} \in \boldsymbol{s}} exp\left[a_{ij} exp\left\{-\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho}\right\}\right] \\ &\approx exp\left[-\frac{\sum_B exp\left(a_{ij} exp\left\{-\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho}\right\}\right)}{n}\right. \\ &\quad \left. + \sum_{s_{ij} \in \boldsymbol{s}} a_{ij} exp\left\{-\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho}\right\}\right]. \end{aligned}$$

We let the contribution of each focus be comprised of two components: expected 3-D individual focus effect ($\theta_{ij}$) and expected 3-D study effect ($p_i$). The parameter

$a_{ij}$ allows a multiplicative focus impact on the intensity. It is possible that the foci are in groups and each groups represents specific types of functions. To this end, we assume the realization of each $s_{ij}$ is from the following mixture for individual clusters, $c = (1, .., C)$:

$$f(s) \propto \sum_{c=1}^{C} exp\left(-\int_B exp\left[a_c exp\left(a_c exp\left(-\frac{||s_{ij} - p_i - \theta_c||^2}{\rho}\right)\right)\right] ds_{ij}\right),$$

after adjusting for study effect. To infer clusters and their centers, we implement a fully Bayesian approach. In the next section we discuss the choice of prior distributions.

### 2.2.1 Prior and hyperprior Distributions

Following the standard Bayesian frame work, all estimable parameters are assigned a prior distribution. We assume the study effect, $p_i$, follows distribution $G_1$, which is generated from a Dirichlet process (DP), $p_i \sim G_1$ and $G_1 \sim DP(\alpha_1, G_{01})$ where $\alpha_1$ is a precision parameter and $G_{01}$ is a base distribution. The base distribution, $G_{01}$, is set to $MVN_3(\boldsymbol{\mu}, \Sigma_1)$ with hyper-prior distribution $\boldsymbol{\mu} \sim MVN_3(\mathbf{0}, 0.1 * I_3)$ and $\Sigma_1 = \sigma_1^2 I_3$, with $\sigma_1^2 \sim IG(0.5, 0.5)$, where $I_3$ is a 3-dimensional identity matrix. The hyper-prior distribution for a noninformative prior was chosen as suggested by Kass and Wasserman [1995]. Similarly, we assume $\theta_{ij} \sim G_2$ and $G_2 \sim DP(\alpha_2, G_{02})$, with a base distribution of $G_{02} = MVN_3(\boldsymbol{c_0}, \Sigma_2)$ where $\boldsymbol{c_0}$ takes the median of the observed data in each dimension, and $\Sigma_2 = \sigma_2^2 I_3$, with $\sigma_2^2 \sim IG(0.5, 0.5)$. In both DPs, we assume the precision parameters $\alpha_1$ and $\alpha_2$ are known. We discuss their selection in the section "selection of $\alpha$". Lastly, the prior for multiplicative effect $a_{ij}$ is conditional on clusters and $a_{ij}|\theta_c \sim N(0, \sigma_a^2)$ with $\sigma_a^2$ known and large. The normalizing constant, $\rho$ is set to 1.

### 2.2.2 Conditional posterior distributions and posterior computing

Posterior inference of $p_i$, $\theta_{ij}$, and $a_{ij}$ is obtained by successfully sampling values from their full conditional posterior distributions through the Markov Chain Monte Carlo (MCMC) simulations, specifically, the Gibbs sampling scheme. Given the observed CBMA data, we would like to sample values for $p_i$, $\theta_{ij}$, $a_{ij}$, and their hyper-prior parameters from the joint posterior:

$$\begin{aligned} P(p_i, \theta_{ij}, a_{ij}, \boldsymbol{\mu}, \Sigma_1, \alpha_1, \Sigma_2, \alpha_2 | \boldsymbol{s}) \quad \propto \quad & P(\boldsymbol{s}|p_i, \theta_{ij}, a_{ij}, \boldsymbol{\mu}, \Sigma_1, \alpha_1, \Sigma_2, \alpha_2) \prod_i^I P(p_i|G_1) \\ & \times P(G_1|\alpha_1, G_{01}) \times P(p_i|G_{01}; \boldsymbol{\mu}, \Sigma_1) P(\boldsymbol{\mu}) P(\Sigma_1) \\ & \times \prod_{c \in C} \prod_{i,j \in c} P(a_{ij}|\theta_{ij}) \prod_{j \in J_i} P(\theta_{ij}|G_2) \\ & \times P(G_2|\alpha_2, G_{02}) P(\theta_{ij}|G_{02}; \boldsymbol{c_0}, \Sigma_2) P(\Sigma_2). \end{aligned}$$

Following the Gibbs sampling scheme, the full conditional posteriors for $p_i$, $\theta_{ij}$, $a_{ij}$, along with their hyper-priors are defined below with the notation "$\cdot$" representing all other parameters,

$$\begin{aligned} P(p_i|s_{ij}, \theta_{ij}, \cdot) \quad \propto \quad & P(s_{ij}|p_i, \theta_{ij}, \cdot) P(\theta_{ij}|\cdot) P(p_i|G_1) P(G_1|\alpha_1, G_{01}) \\ = \quad & exp\left[ -\frac{\sum_B exp\left( a_{ij} exp\left\{ -\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho} \right\} \right)}{n} \right. \\ & \left. + \sum_{j=1}^{J_i} a_{ij} exp\left\{ -\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho} \right\} \right] \\ & \times \frac{\alpha_1}{(\alpha_1 + I - 1)} MVN_3(\boldsymbol{\mu}, \Sigma_1) + \frac{\sum_{q=1, q \neq i}^I \delta_{p_q}(p_i)}{(\alpha_1 + I - 1)} \end{aligned}$$

$$\begin{aligned} P(\boldsymbol{\mu}|\Sigma_1, p_i, \cdot) \quad \propto \quad & \prod_i^I P(p_i|G_{01}; \boldsymbol{\mu}, \Sigma_1) P(\Sigma_1) P(\boldsymbol{\mu}) \\ = \quad & exp\left[ -\frac{1}{2} \left\{ \sum_i^I (p_i - \boldsymbol{\mu})' \Sigma_1^{-1} (p_i - \boldsymbol{\mu}) + \boldsymbol{\mu}' (0.1 I_3)^{-1} \boldsymbol{\mu} \right\} \right] \\ \boldsymbol{\mu}|\cdot \quad \sim \quad & MVN_3\left( (10 + I\sigma_p^{-2})^{-1} (I\sigma_p^{-2} \overline{p}), (10 + I\sigma_p^{-2})^{-1} I_3 \right) \end{aligned}$$

$$
\begin{aligned}
P(\sigma_1^2|\boldsymbol{\mu}, p_i, \cdot) &\propto \prod_i^I P(p_i|G_{01}; \boldsymbol{\mu}, \Sigma_1) P(\boldsymbol{\mu}) P(\Sigma_1) \\
&= exp\left[-\frac{1}{2}\left\{\sum_i^I (p_i - \boldsymbol{\mu})'(\sigma_1^2 I_3)^{-1}(p_i - \boldsymbol{\mu})\right\} + -\frac{0.5}{\sigma_1^2}\right] \\
\sigma_1^2|\cdot &\sim IG\left(\frac{1+I}{2}, \frac{\sum_i^I\{||p_i - \boldsymbol{\mu}||^2\} + 1}{2}\right)
\end{aligned}
$$

$$
\begin{aligned}
P(\theta_{ij}|s_{ij}, p_i, \cdot) &\propto P(s_{ij}|\theta_{ij}, p_i, \cdot) P(p_i|\cdot) P(\theta_{ij}|G_2) P(G_2|\alpha_2, G_{02}) \\
&= exp\left[-\frac{\sum_B exp\left(a_{ij} exp\left\{-\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho}\right\}\right)}{n}\right. \\
&\quad \left. + a_{ij} exp\left\{-\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho}\right\}\right] \\
&\quad \times \frac{\alpha_2}{(\alpha_2 + n - 1)} MVN_3(\boldsymbol{c_0}, \Sigma_2) + \frac{\sum_{q=1, q\neq ij}^n \delta_{\theta_q}(\theta_{ij})}{(\alpha_2 + I - 1)}
\end{aligned}
$$

$$
\begin{aligned}
P(\sigma_2^2|\boldsymbol{c_0}, \theta_{ij}, \cdot) &\propto \prod_i^I \prod_j^{J_i} P(\theta_{ij}|G_{02}; \boldsymbol{c_0}, \sigma_2^2) P(\sigma_2^2) \\
&= exp\left[-\frac{1}{2}\left\{\sum_i^I \sum_j^{J_i} (\theta_{ij} - \boldsymbol{c_0})'(\sigma_2^2 I_3)^{-1}(\theta_{ij} - \boldsymbol{c_0})\right\} + -\frac{0.5}{\sigma_2^2}\right] \\
\sigma_2^2|\cdot &\sim IG\left(\frac{1+n}{2}, \frac{\sum_{i,j}^n ||\theta_{ij} - \boldsymbol{c_0}||^2 + 1}{2}\right)
\end{aligned}
$$

$$
\begin{aligned}
P(a_{ij}|s_{ij}, p_i, \theta_{ij}, c, \cdot) &\propto \prod_{i,j\in c} P(s_{ij}|\theta_{ij}, p_i, \cdot) P(\theta_{ij}|c, \cdot) P(p_i|\cdot) P(a_{ij}) \\
&= exp\left[-\frac{\sum_B exp\left(a_{ij} exp\left\{-\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho}\right\}\right)}{n}\right. \\
&\quad \left. + \sum_{i,j\in c} a_{ij} exp\left\{-\frac{||s_{ij} - p_i - \theta_{ij}||^2}{\rho}\right\} - \frac{(a_{ij})^2}{50}\right]
\end{aligned}
$$

Non-standard conditional posteriors, $a_{ij}$, are updated using Metropolis-Hastings steps in the Gibbs sampler. The parameters $p_i$ and $\theta_{ij}$ are sampled via using an algorithm introduced in Neal [2000] that is discussed below.

The algorithm, specifically algorithm 8, noted in Neal [2000], is appropriate for models with non-conjugate priors. It introduces $m$ auxiliary parameters to represent potential values for our parameter of interest, $p_i$ or $\theta_{ij}$, that are not associated with any other observations [Neal, 2000]. The original algorithm for updating cluster assignments, $c$, is as follows:

- Let the state of the Markov chain consist of $c = \{c_1, ..., c_n\}$ and $\Phi = (\phi_c \; c \in c_1, ..., c_n)$ with $\phi_c$ density cluster parameters. In our application, $\phi_c$ refers to the center of individual cluster or study cluster $c$. Repeatedly sample as follows:

- For $i = 1, ..., n$ : Let $k^-$ be the number of distinct $c_l$ for $l \neq i$, and let $h = k^- + m$. Label these $c_l$ with values in $\{1, ..., k^-\}$. If $c_i = c_l$ for some $l \neq i$, draw values independently from base distribution $G_0$ for those $\phi_c$ for which $k^- < c \leq h$. If $c_i \neq c_l$ for all $l \neq i$, let $c_i$ have the label $k^- + 1$, and draw values independently from $G_0$ for those $\phi_c$ for which $k^- + 1 < c \leq h$. Draw a new value for $c_i$ from $\{1, ..., h\}$ using the following probabilities:

$$P(c_i = c | c_{-i}, y_i, \phi_1, ..., \phi_h) \propto \begin{cases} \frac{n_{-i,c}}{n-1+\alpha} F(y_i, \phi_c) & \text{for } 1 \leq c \leq k^- \\ \frac{(\alpha/m)}{n-1+\alpha} F(y_i, \phi_c) & \text{for } k^- < c \leq h \end{cases},$$

  where $F(y_i, \theta_c)$ is the likelihood with $\theta_c$ and observation $i$, $y_i$, involved. In our case, the observed data $y_i$ is $s_{ij}$.

- Where $n_{-i,c}$ is the number of $c_l$ for $l \neq i$ that are equal to $c$. Change the state to contain only those $\phi_c$ that are now associated with one or more observation.

- For all $c \in \{c_1, ..., c_n\}$: Draw a new values from $P(\phi_c | y_i)$ such that $c_i = c$, or perform some other update to $\phi_c$ that leaves this distribution invariant [Neal, 2000].

To illustrate, $F(y_i, \phi_c)$ and $P(\phi_c | y_i)$ in our application are $F(s_{ij}, \theta_c)$ for $\theta_{ij}$ and

$P(\theta_c|s_{ij}, c, \cdot)$, respectively, with:

$$\begin{aligned} F(s_{ij}, \theta_c) &= exp\left\{-\int_B \left(exp\left\{a_{ij}exp\left[-\frac{||s_{ij}-p_i-\theta_c||^2}{\rho}\right]\right\}\right) ds_{ij} \right. \\ &\quad \left. + a_{ij}exp\left[-\frac{||s_{ij}-p_i-\theta_c||^2}{\rho}\right]\right\}, \end{aligned}$$

and

$$\begin{aligned} P(\theta_c|s_{ij}, c, \cdot) \propto\ & exp\left\{-\frac{\sum_B \left(exp\left\{a_{ij}exp\left[-\frac{||s_{ij}-p_i-\theta_c||^2}{\rho}\right]\right\}\right)}{n}\right. \\ & \left. + \sum_{s_{ij}\in c} a_{ij}exp\left[-\frac{||s_{ij}-p_i-\theta_c||^2}{\rho}\right]\right\} \\ & \times exp\left(-\frac{1}{2}(\theta_c - \boldsymbol{c_0})'\Sigma_2^{-1}(\theta_c - \boldsymbol{c_0})\right), \end{aligned}$$

where $\theta_c$ is the center for individual foci cluster $c$.

To accommodate the 3-D nature of our data and to improve sampling efficiency, we modified algorithm 8 by introducing auxiliary parameters into one, two, or three dimensions of the centers at the current MCMC iteration. Taking $\theta_{ij}$ as an example, we take $m = 7$ auxiliary parameters. Let $\theta_{c_i} = \{\phi_{c_i}^{(x)}, \phi_{c_i}^{(y)}, \phi_{c_i}^{(z)}\}$ be the center for focus $i$ and let $\phi_0 = \{\phi_0^{(x)}, \phi_0^{(y)}, \phi_0^{(z)}\}$ be a single draw generated from $G_{02}$. An auxiliary parameter can be chosen as any combination of the current cluster center and the $G_{02}$ sampled center such as $(\phi_{c_i}^{(x)}, \phi_{c_i}^{(y)}, \phi_0^{(z)})$ which gives:

$$P(c_i = c|c, s_{ij}, \theta_1, ..., \theta_m) = \begin{cases} \frac{n_c}{n+\alpha}F(s_{ij}, \theta_c) & \text{for } 1 \le c \le k \\ \frac{\alpha/m}{n+\alpha}F(s_{ij}, (\phi_{c_i}^{(x)}, \phi_{c_i}^{(y)}, \phi_0^{(z)})) & \text{for } c = k+1 \\ \frac{\alpha/m}{n+\alpha}F(s_{ij}, (\phi_{c_i}^{(x)}, \phi_0^{(y)}, \phi_{c_i}^{(z)})) & \text{for } c = k+2 \\ \frac{\alpha/m}{n+\alpha}F(s_{ij}, (\phi_0^{(x)}, \phi_{c_i}^{(y)}, \phi_{c_i}^{(z)})) & \text{for } c = k+3 \\ \frac{\alpha/m}{n+\alpha}F(s_{ij}, (\phi_{c_i}^{(x)}, \phi_0^{(y)}, \phi_0^{(z)})) & \text{for } c = k+4 \\ \frac{\alpha/m}{n+\alpha}F(s_{ij}, (\phi_0^{(x)}, \phi_{c_i}^{(y)}, \phi_0^{(z)})) & \text{for } c = k+5 \\ \frac{\alpha/m}{n+\alpha}F(s_{ij}, (\phi_0^{(x)}, \phi_0^{(y)}, \phi_{c_i}^{(z)})) & \text{for } c = k+6 \\ \frac{\alpha/m}{n+\alpha}F(s_{ij}, (\phi_0^{(x)}, \phi_0^{(y)}, \phi_0^{(z)})) & \text{for } c = m, \end{cases} \tag{2.1}$$

where $F(s_{ij}, \theta_c)$ was defined in equation 2.1. By defining auxillary parameters as in equation 2.1, the variation between iteration will be smaller, which potentially improves convergence efficiency. Similar settings are applied to $p_i$. Related probability density functions for $p_i$ for study clusters $k \in K$ are,

$$\begin{aligned} F(s_{ij}, p_k) &= exp\left\{ -\int_B \left( exp\left\{ a_{ij} exp\left[ -\frac{||s_{ij} - p_k - \theta_{ij}||^2}{\rho} \right] \right\} \right) ds_{ij} \right. \\ &\quad \left. + \sum_{j=1}^{J_i} a_{ij} exp\left[ -\frac{||s_{ij} - p_k - \theta_{ij}||^2}{\rho} \right] \right\}, \end{aligned}$$

where $p_k$ is the center for study cluster $k$, and

$$\begin{aligned} P(p_k|k, s_{ij}, \cdot) &\propto \prod_{p_i \in k} P(s_{ij}|p_k, k, \cdot) P(p_k|G_1) P(G_1|\alpha, G_{01}) P(G_{01}|k, \boldsymbol{\mu}, \Sigma_1) P(\boldsymbol{\mu}) P(\Sigma_1) \\ &= \prod_{p_i \in k} exp\left\{ -\frac{\sum_B \left( exp\left\{ a_{ij} exp\left[ -\frac{||s_{ij} - p_k - \theta_{ij}||^2}{\rho} \right] \right\} \right)}{n} \right. \\ &\quad \left. + \sum_{i \in k} \sum_{j=1}^{J_i} a_{ij} exp\left[ -\frac{||s_{ij} - p_k - \theta_{ij}||^2}{\rho} \right] \right\} \\ &\quad \times exp\left( -\frac{1}{2} (p_k - \boldsymbol{\mu})' \Sigma_1^{-1} (p_k - \boldsymbol{\mu}) \right). \end{aligned}$$

### 2.2.3 Determining the clusters

To infer point estimates for cluster centers and cluster assignment, we implement the least-squared Euclidean distance method introduced in Dahl [2006]. This method draws the inferences based on a set of converged MCMC iterations and chooses one iteration as the final estimates on the clusters and related parameters. This final MCMC iteration is selected due to its smallest Euclidean distance to the expected cluster assignments estimated based on a set of independent converged MCMC iterations. The procedure is outline as follows,

1. After $B$ iterations for burn-in, run the MCMC chain for an additional $W$ iterations and estimated the expected cluster assignments as,

$$P_{ij} = \frac{\#\text{ of iterations such that } c_i = c_j}{W},$$

where $P_{ij}$ represents the probability that observations $i$ and $j$ are in one cluster. This will form an $n \times n$ matrix with entry $(i, j)$ being $P_{ij}$.

2. Run another $T$ iterations, and select one iteration such that the cluster assignment in that iteration minimizes the following Euclidean distance,

$$\underset{t\in(1,\ldots T)}{\operatorname{argmin}} \sum_{h=1}^{ij} \sum_{g=1}^{ij} \left(\delta_{h,g}^{(t)} - P_{h,g}\right)^2$$

where $\delta_{h,g}^{(t)} = 1$ if observations $h$ and $g$ are in the same cluster and 0 otherwise.

3. The iteration determined in step 2) provides a point estimate on the number of clusters along with estimates on other parameters.

Inferring clusters in this manner, incorporates all clustering information in the MCMC sample process [Dahl, 2006]. We would like to note that the clustering pattern to be summarized is for individual foci clusters as they are our primary interest.

### 2.2.4 Selection of $\alpha$

Selection of $\alpha$ can have a potentially significant effect on the number of clusters identified due to its direct impact on the aggregation of $G$ about $G_0$. A smaller choice of $\alpha$ places less weight upon the base distribution, therefore resulting in a smaller number of clusters. A larger choice of $\alpha$ indicates a greater weight placed on the base distribution and therefore a large number of clusters. The extent of the precision parameter's sensitivity and various ways to estimate this parameter have been discussed in a number of studies, e.g., Liu [1996], McAuliffe et al. [2006], Kyung et al. [2010], Dorazio et al. [2008], Doss [2008, 2012], Naskar and Das [2004, 2006] and many others. Several of these articles [Liu, 1996, McAuliffe et al., 2006, Dorazio et al., 2008] collectively suggest an empirical Bayes approach where posterior inferences are computed conditional on the maximum likelihood estimator of the precision parameter. As Dorazio et al. [2008] acknowledges that this calculation is

very computationally intensive and may not capture a true maximum in the situation of a flat likelihood [Kyung et al., 2010]. Naskar and Das [2004, 2006] implemented the Monte Carlo expectation and maximization algorithm to empirically estimate $\alpha$ but did not further investigate the estimate's properties while Doss [2008, 2012], Kyung et al. [2010] calculated an estimate with a marginal or profile likelihood. These methods can not avoid intensive computational burden and require repeated Gibbs samplers. So far, it seems that an objective and efficient method for determining $\alpha$ is not available.

Given the importance of $\alpha$, we decided to select an estimate based on information in the data and chose $\alpha$ based on the posterior likelihood. More specifically, the choice of $\alpha_1$ for $p_i$ and $\alpha_2$ for $\theta_{ij}$ were selected iteratively based on a grid search on a set of possible values for $\alpha_1$ and $\alpha_2$ that optimize the deviance information criterion (DIC) [Congdon, 2007]. The DIC is an estimate for the expected deviance that is adjusted for the models complexity as to not overfit the data [Congdon, 2007, Spiegelhalter et al., 2002]. Specifically, DIC is defined as

$$DIC = 2\overline{D} - D(\overline{\theta}|\boldsymbol{s}),$$

where

$$\begin{aligned} \overline{D} &= \frac{1}{T}\sum_{t=1}^{T} D(\boldsymbol{s}, \theta^{(t)}) \\ D(y, \theta^{(t)}) &= -2logP(\boldsymbol{s}|\theta^{(t)}) \\ D(\overline{\theta}|\boldsymbol{s}) &= D(\boldsymbol{s}, \overline{\theta}). \end{aligned}$$

In the above, $\theta^{(t)}$ is the parameter estimate(s) at current time $t$, $\overline{\theta}$ is mean or median of the parameter estimate(s) for times $t = 1, ..., T$ and $\boldsymbol{s}$ is the data [Congdon, 2007]. A smaller DIC indicates a better fit of the model.

## 2.3 Simulation Studies

Simulations were used to demonstrate and asses the proposed method. In total, 50 studies each with 10 foci were considered. Three individual foci clusters are spatially centered at $(1,1,1)^T$, $(2,2,2)^T$, and $(4,4,4)^T$ containing 150, 150, and 200 foci, respectively. Two study clusters are assumed with centers held at $(0.1,0.1,0.1)^T$ and $(0.4,0.4,0.4)^T$ with each including 25 studies (250 foci each). In addition, we considered the following simulation scenarios,

1. For the purpose of illustration, we simulate the data for each cluster via multivariate normal with mean set at the individual foci centers and variance $\Sigma = 0.002I_3$. Thus each cluster is sphere with small variation and we expect the method to have the ability to correctly identify the clusters.

2. To demonstrate its ability to cluster outliers, we follow the same setting as in scenario 1) but added an additional focus in the third individual foci cluster located in the right 5% tail in a multivariate normal distribution with mean $(4,4,4)^T$ and covariance matrix $0.002I_3$

3. It is important to examine the robustness of the method with respect to abnormal patterns. The same scenarios as in 1) are followed to simulate individual foci clusters 1 and 2. Cluster 3 is simulated using truncated normal distribution with mean $(4,4,4)^T$ and variance $0.002I_3$ with a lower bound $(1,1,1)^T$.

4. The last scenario is designed to asses the robustness of the method with respect to the distance between and among clusters. To this end, besides $\Sigma = 0.002I_3$, we considered four additional levels of $\Sigma$: $\Sigma = 0.01I_3, 0.05I_3, 0.1I_3$, and $0.2I_3$ representing gradually closer distances among clusters. Other settings are as in scenario 1).

In total, 100 Monte Carlo (MC) replicates are generated for each scenario. For each setting, we randomly chose one dataset to estimate $\alpha_1$ and $\alpha_2$ through gird search by minimizing DIC. Possible values of $\alpha_1$ and $\alpha_2$ are ranged from 0.1 to 1.5 with 0.1 representing small influence of the base distribution on the number of clusters and 1.5 large influence. After $\alpha_1$ and $\alpha_2$ are chosen for each scenario, for each dataest, after burn in, we run 2,500 working iterations to determine the probability matrix noted in section 2.4 and 1,000 additional iterations to infer the number of clusters and individual foci cluster centers.

Model assessment consists of three evaluations: sensitivity, specificity, and percentage of correct clustering. Sensitivity is defined per cluster as the proportion of foci that are correctly assigned to that given cluster, Se=TP/(TP+FN) and specificity is defined per cluster as the proportion of foci that are correctly not assigned to a cluster, Sp=TN/(TN+FP). In these definitions, true positive (TP) denotes a focus in that respective cluster is also assigned to that cluster, false negative (FN) denotes a focus in that respective cluster but not assigned to that cluster, true negative (TN) is a focus that is not in the respective cluster and not assigned to that cluster, and false positive (FP) denotes a focus that is not in that respective cluster but assigned to that cluster. Percentage of correct clustering is an overall measure defined as the proportion of foci that are correctly clustered. Note that the definition of correctness takes into account both TP and TN.

To illustrate the selection of $\alpha_1$ and $\alpha_2$ via grid search, we use scenario 1). All possible combination of candidate values for $\alpha_1$ and $\alpha_2$ are considered and the DIC for each combination is calculated based on the converged 3,500 MCMC samples. As indicated in Figure 2.1, the best DIC is achieved with $\alpha_1 = 1.5$ and $\alpha_2 = 0.5$. These two precision parameters are then applied in to infer the individual foci clusters.

Table 2.1 summaries the findings on individual foci cluster identification and the quality of the identified clusters. Overall, the method is robust with respect to outlier,

skewness, and large variation. Among the 100 MC replicates, the proposed method correctly assigned most foci to clusters except for the situation of large variance,($\Sigma = 0.2I_3$) for generating individual foci clusters. When the variance is comparable to the study effect it can severely impact the estimate of study effect, which might have been the cause for low sensitivity and correctness rates.

The scenarios we chose represent important facets of variability that a spatial models need to be able to handle and overcome in order to accurately perform. Based on our simulations, it can be inferred that across all the scenarios our proposed model in general performs well in correctly identifying the individual foci clusters.

## 2.4 Real Data Analysis

In this section, we apply the proposed model to a meta-analysis dataset. This dataset was first discussed in Kober et al. [2008] and further studied in Kang et al. [2011]. The analysis consists of a total of 162 neuroimaging publications with 57 PET and 105 fMRI were considered. PET scans are very similar to fMRI in respect to its data smoothness and interpretation of the signal [Feng et al., 2004]. Among these 162 publications, there were 437 contrasts or studies. Only those foci that were deemed significantly activated by their study specific criteria were included for a total of 2,478 foci. This meta-analysis analyzed emotions and therefore there exist specific brain regions that were of interest to researchers. Foci that lie within these regions were noted. As seen in Table 2.2, there was an average of 15.11 foci per publication and 5.67 foci per study. Additionally there was an average of 2.67 studies per publication. The emotion "affective" (Table 2.3) was the most frequent emotion found in 175 studies and surprise the least frequent emotion found in only 2 studies. Of the total 2,478 foci, 711 foci fell within regions of interest (ROI). Figure 2.2 presents an illustration of the meta-data.

To assist with the magnitude of the likelihood calculations, the data was scaled

down by 10. Iterative grid search was implemented to calculate precision parameters via DIC. Potential precision parameters values for $\alpha_1$ were 0.1, 0.5, 1.0, and 1.5, and for $\alpha_2$ were 0.1, 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0. Each combination was performed over 8,000 iterations, 4,000 of those for burn-in, 3,000 for the probability matrix calculation, and final 1,000 to infer individual clusters and their centers.

It was found that the precision parameter combination of $\alpha_1 = 0.5$ and $\alpha_2 = 1.5$ produced the smallest DIC. Convergence over 8,000 iterations, with the initial 4,000 discarded, was checked visibly. Based on the proposed method, we identified 13 study clusters and 53 individual clusters (Tables 2.4, 2.5). The break down of each cluster by it's center location, foci frequency and emotion frequency can be seen in Tables 2.6, 2.7, 2.8, 2.9. Of the 53 individual clusters, only 3 of those contained more than five percent of the total number of foci: cluster 1 centered at (-17.08,-10.75,-5.88), cluster 2 centered at (26.67,-9.91,-7.14), and cluster 3 centered at (45.29,14.25,0.76). However, less than 50% of those foci fell within regions of interest (Table 2.4). Of the 53 clusters, 40 of those contained foci related to at least 6 of the 8 emotions. More specifically, with the exception of the emotion "affective" that was present in every cluster with more than 5 foci, 35 clusters had a majority of foci that catered to a specific emotion: 12 clusters identified foci associated mainly with fear, 12 clusters identified foci associated mainly with sadness, 10 clusters identified foci associated mainly with disgust and one cluster identified foci associated mainly with happiness(Tables 2.6, 2.7, 2.8, 2.9). This implies that multiple regions in the brain contribute to one emotion Another eight clusters did not have a majority foci related to a specific emotion; the percentages of the foci for different emotions found within that cluster were equal (Tables 2.6, 2.7, 2.8, 2.9). The remaining nine clusters were not included in this comparison has they had less than 5 foci (Tables 2.6, 2.7, 2.8, 2.9).

When only interested in those foci that fell within ROI boundaries, 48 clusters were identified with 7 clusters identifying a majority of foci associated with fear,

14 clusters identifying a majority of foci associated with sadness, 5 clusters clusters identifying a majority of foci associated with disgust, two clusters clusters identifying a majority of foci associated with happiness and one cluster identifying a majority of foci associated with mixed emotions (Tables 2.10, 2.11, 2.12). Another 10 clusters did not have a majority foci related to a specific emotion; the percentages of the foci for different emotions found within that cluster were equal (Tables 2.10, 2.11, 2.12). The remaining eight clusters were not included in this comparison has they had less than 5 foci (Tables 2.10, 2.11, 2.12).

This clustering demonstrates the model's ability to incorporate spatial information after adjusting for similarities between studies and adequately cluster foci. From here, the primary interest would be assess what physiological similarities exist between the clusters with the help of a neuroscientist to determine the precision of the clustering. Based on these results and due to the Bayesian nature, some of the priors may need to be adjusted.

## 2.5 Conclusion and Discussion

The proposed spatial Cox point process model with a Gaussian kernel driven intensity function was motivated by the need to spatially cluster coordinated-based meta analysis data to identify activated regions with the brain. Furthermore, the Gaussian kernel incorporated a study and sub-study effects that were estimated using a Dirichlet Process. The advantage of implementing a DP is that it allows the sub-study clusters to not only include spatial information and adjust for study effect but to have flexible distributions and therefore irregularly shaped clusters.

Simulation studies were performed to asses the model's accuracy, sensitivity, flexibility, and robustness. With the exception of one setting that simulated three spherical clusters that overlapped, the model performed extremely well with average sensitivity, specificity, and percent of correct clustering ranging from 82-100% for all

sub-study or individual clusters. Specifically, when the model estimated skewed or irregularly shaped clusters it correctly identified all three individual cluster that were simulated.

The model was further applied to an emotion meta dataset in which it identified 53 individual foci clusters. These clusters can not be directly compared with those derived in previous methods such as MKDA (Kober et al. [2008]) and hierarchical spatial clustering (Kang et al. [2011]) due to inconsistent interpretability. However, several regions of interest appear in multiple clusters suggesting the overlapping over clusters. It also stands to mention the natural limitation this meta-data as it contains studies using both fMRI and PET scans. As discussed in Chapter 1, there is potential for differences between fMRI and PET scans due to different resolutions and time constraints. Therefore, activations identified in one may not be consistent in the other. For more consistent and interpretable results, the proposed method should be applied to additional fMRI data that is not a combination of fMRI and PET scans.

Other than the advantage of having the flexibility to identify irregular patterns, this model can also be extended to analyze any type of spatial data and adjust for any number of covariates. There are no assumptions or restrictions placed on the model that require it to be only fMRI brain data. Therefore, this proposed model can be applied to a variety of settings.

The method was limited by the inability to identify the correct cluster centers if the study effect was large. If the study effect center was shifted, the sub-study or individual cluster centers mimicked to offset this shift and therefore the incorrect center was identified. Generally these shifts were small and stricter priors helped to minimize this shift. The inclusion of a contrast would set some reference estimate between two studies and provide more accurate cluster center estimates. Additionally, the clustering nature of the DP aims to identify mixtures of distributions. However, given the precision parameter and variance of the base distribution the DP may be

over sensitive and too flexible such that it identifies multiple clusters as a single multimodal distribution. This model is further limited by it's inability to explicitly test whether distribution peaks should be separate clusters with a different distribution or a single cluster with multimodes. Therefore, we propose to model the distribution of the individual foci effects as a mixture of Dirichlet processes. Handling the $\theta_{ij}$ in this manner will still allow the clusters, or components of the mixture, the flexibility to capture irregular shapes but while explicitly modeling if a distribution peak is generated from a different base distribution or a mode or peak within another cluster. We implement this setting in the next chapter.

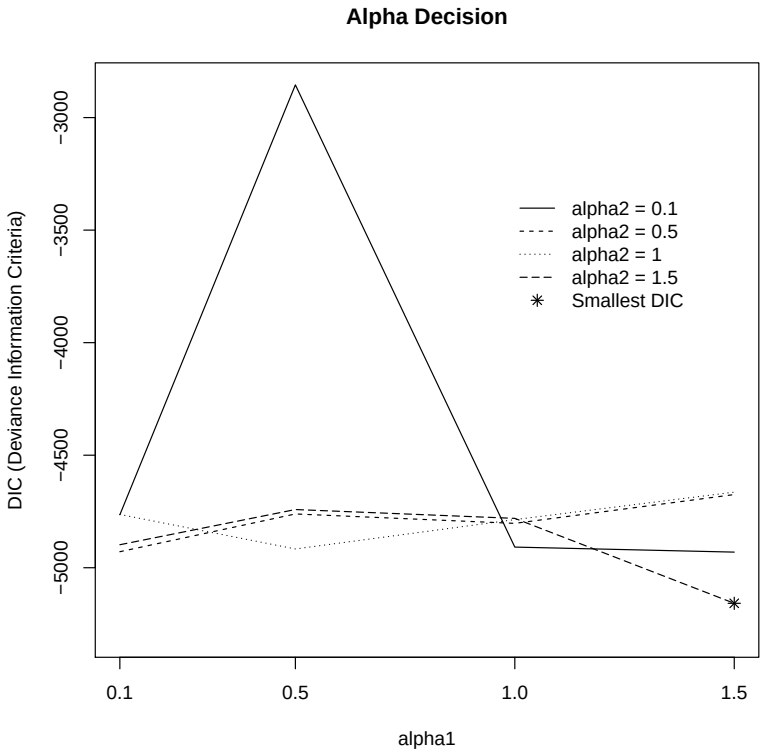


Figure 2.1 Example of grid search for selection of $\alpha_1$ and $\alpha_2$ based on DIC

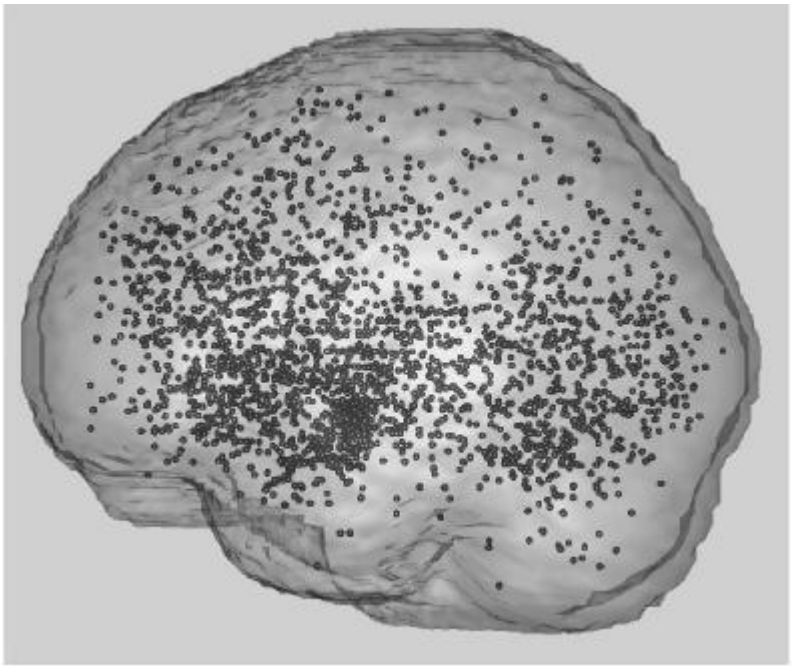

Figure 2.2 FMRI Meta-data

Table 2.1 Simulation assessments

| Scenario: | Median Num. of Clusters (SD)* | Cluster index | Average Sensitivity (SD)* | Average Specificity (SD)* | Average % Correctness rate (SD)* |
|---|---|---|---|---|---|
| Normal | **IC: 3 (0.55) | 1 | 1.00 (0.00) | 1.00 (0.00) | 1.00 (0.00) |
| | | 2 | 1.00 (0.01) | 1.00 (0.00) | |
| | | 3 | 1.00 (000) | 1.00 (0.00) | |
| | **SC: 3 (0.55) | 1 | 0.89 (0.14) | 1.00 (0.00) | 0.92 (0.09) |
| | | 2 | 0.94 (0.11) | 1.00 (0.00) | |
| Outlier | IC: 3 (0.86) | 1 | 0.80 (0.26) | 0.94 (0.1) | 0.78 (0.22) |
| | | 2 | 0.79 (0.29) | 0.91 (0.11) | |
| | | 3 | 0.77 (0.28) | 0.92 (0.13) | |
| | SC: 5 (1.99) | 1 | 0.55 (0.29) | 0.84 (0.26) | 0.53 (0.21) |
| | | 2 | 0.51 (0.33) | 0.87 (0.23) | |
| | | 3 | 0.44 (0.5) | 0.93 (0.11) | |
| Skewed | IC: 3 (0.35) | 1 | 1.00 (0.00) | 1.00 (0.02) | 0.99 (0.04) |
| | | 2 | 1.00 (0.00) | 1.00 (0.03) | |
| | | 3 | 0.99 (0.1) | 1.00 (0.00) | |
| | SC: 2 (0.35) | 1 | 0.99 (0.04) | 1.00 (0.01) | 0.99 (0.03) |
| | | 2 | 0.99 (0.05) | 1.00 (0.00) | |
| Large1 | IC: 3 (1.02) | 1 | 0.81 (0.32) | 0.94 (0.1) | 0.82 (0.19) |
| | | 2 | 0.8 (0.31) | 0.89 (0.15) | |
| | | 3 | 0.83 (0.25) | 0.93 (0.16) | |
| | SC: 2 (1.02) | 1 | 0.36 (0.41) | 1.00 (0.02) | 0.61 (0.19) |
| | | 2 | 0.86 (0.17) | 0.48 (0.48) | |
| Large2 | IC: 3 (0.45) | 1 | 0.99 (0.01) | 1.00 (0.00) | 1.00 (0.00) |
| | | 2 | 0.99 (0.01) | 1.00 (0.00) | |
| | | 3 | 1.00 (0.00) | 1.00 (0.00) | |
| | SC: 2 (0.45) | 1 | 0.99 (0.02) | 0.99 (0.02) | 0.99 (0.02) |
| | | 2 | 0.99 (0.03) | 1 (0.01) | |
| Large3 | IC: 3 (1.09) | 1 | 0.91 (0.14) | 0.98 (0.04) | 0.90 (0.16) |
| | | 2 | 0.89 (0.21) | 0.96 (0.08) | |
| | | 3 | 0.89 (0.21) | 0.97 (0.07) | |
| | SC: 3 (1.09) | 1 | 0.86 (0.14) | 0.96 (0.05) | 0.86 (0.13) |
| | | 2 | 0.86 (0.15) | 0.95 (0.06) | |
| Large4 | IC: 13 (2.57) | 1 | 0.47 (0.18) | 0.98 (0.01) | 0.46 (0.1) |
| | | 2 | 0.44 (0.16) | 0.98 (0.01) | |
| | | 3 | 0.47 (0.16) | 0.99 (0.01) | |
| | SC: 5 (2.57) | 1 | 0.21 (0.28) | 0.92 (0.13) | 0.41 (0.17) |
| | | 2 | 0.62 (0.18) | 0.53 (0.25) | |

*SD: standard deviation across 100 MC replicates;**IC: individual foci cluster; SC: study effect clusters

Table 2.2 Descriptive statistics*

| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|---|
| Number of foci per pub. | 1.00 | 5.75 | 10.00 | 15.11 | 17.25 | 110.00 |
| Number of foci per study | 1.00 | 2.00 | 4.00 | 5.67 | 7.00 | 47.00 |
| Number of subjects per pub. | 4.00 | 9.00 | 11.00 | 12.26 | 14.00 | 40.00 |
| Number of studies per pub. | 1.000 | 1.000 | 2.000 | 2.67 | 4.000 | 12.000 |

*Min: minimum, 1st Qu: 25% percentile, 3rd Qu: 75% percentile, Max: maximum, pub: publication

Table 2.3 Frequency of emotions

| Emotions | Frequency of studies (% of total studies) | Frequency of foci (% of total foci) |
|---|---|---|
| aff* | 175 (40.05%) | 881 (35.55%) |
| anger | 26 (5.95%) | 166 (6.7%) |
| disgust | 44 (10.07%) | 337 (13.6%) |
| fear | 68 (15.56%) | 367 (14.81%) |
| happy | 36 (8.24%) | 178 (7.18%) |
| mixed | 41 (9.38%) | 195 (7.87%) |
| sad | 45 (10.3%) | 348 (14.04%) |
| surprise | 2 (0.46%) | 6 (0.24%) |
| Total | 437 | 2478 |

*aff:affective

Table 2.4 Meta-data cluster results

| Individual Foci Clusters | | | |
|---|---|---|---|
| Cluster Centers/ Brain Regions | Cluster Index | # of foci per cluster(% of total foci) | # of studies per cluster(% of all studies) |
| (-17.08,-10.75,-5.88)/R Extra-Nuclear | 1 | 183 (7.38) | 143 (32.72) |
| (26.67,-9.91,-7.14)/R Inferior Temporal Gyrus | 2 | 149 (6.01) | 119 (27.23) |
| (45.29,14.25,0.76)/R Insula | 3 | 126 (5.08) | 100 (22.88) |
| (47.48,-72.58,1.09)/L Precuneus | 4 | 112 (4.52) | 92 (21.05) |
| (-0.3,-14.27,7.4)/R Caudate | 5 | 90 (3.63) | 71 (16.25) |
| (1.87,-30.38,3.51)/Inter-Hemispheric | 6 | 86 (3.47) | 74 (16.93) |
| (-35.31,-71.65,-4.73)/Inter-Hemispheric | 7 | 86 (3.47) | 75 (17.16) |
| (-1.37,31.21,6.94)/R Extra-Nuclear | 8 | 82 (3.31) | 66 (15.1) |
| (3.35,47.18,31.23)/R Cingulate Gyrus | 9 | 74 (2.99) | 63 (14.42) |
| (55.65,-46.89,15.04)/L Culmen | 10 | 74 (2.99) | 62 (14.19) |
| (-2.12,33.27,32.1)/R Lentiform Nucleus | 11 | 66 (2.66) | 54 (12.36) |
| (48.51,0.71,-10.4)/L Extra-Nuclear | 12 | 64 (2.58) | 58 (13.27) |
| (-37.69,24.09,0.55)/R Thalamus | 13 | 63 (2.54) | 57 (13.04) |
| (27.35,-4.86,-3.95)/L Lentiform Nucleus | 14 | 62 (2.5) | 57 (13.04) |
| (41.85,-57.43,-7.42)/L Insula | 15 | 60 (2.42) | 55 (12.59) |
| (-39.8,4.44,35.17)/L Lentiform Nucleus | 16 | 59 (2.38) | 50 (11.44) |
| (-38.99,11.35,3.37)/R Superior Temporal Gyrus | 17 | 57 (2.3) | 49 (11.21) |
| (21.18,-12.28,-5.21)/Superior Frontal Gyrus | 18 | 57 (2.3) | 52 (11.9) |
| (34.18,23.77,3.01)/R Middle Frontal Gyrus | 19 | 55 (2.22) | 49 (11.21) |
| (24,-61.44,-1.15)/R Lingual Gyrus | 20 | 54 (2.18) | 49 (11.21) |
| (-29.1,7.9,-6.26)/L Extra-Nuclear | 21 | 50 (2.02) | 48 (10.98) |
| (-0.03,16.64,-3.9)/R Inferior Frontal Gyrus | 22 | 49 (1.98) | 43 (9.84) |
| (0.8,-69.87,28.61)/Inter-Hemispheric | 23 | 48 (1.94) | 45 (10.3) |
| (43.57,9.69,9.73)/L Extra-Nuclear | 24 | 47 (1.9) | 42 (9.61) |
| (18.77,-23.02,6.35)/L Extra-Nuclear | 25 | 46 (1.86) | 38 (8.7) |
| (5.67,3.4,65.88)/R Sub-Gyral | 26 | 45 (1.82) | 40 (9.15) |
| (4.62,18.8,37.87)/L Lentiform Nucleus | 27 | 45 (1.82) | 38 (8.7) |
| (-27.34,-0.65,-4.96)/L Superior Temporal Gyrus | 28 | 44 (1.78) | 39 (8.92) |
| (-28.99,-12.74,-3.07)/R Extra-Nuclear | 29 | 42 (1.69) | 40 (9.15) |
| (-18.91,21.88,-4.63)/R Lentiform Nucleus | 30 | 39 (1.57) | 37 (8.47) |
| (-41.65,-7.56,7.19)/L Insula | 31 | 37 (1.49) | 36 (8.24) |
| (10.46,1.51,7.67)/R Declive | 32 | 37 (1.49) | 34 (7.78) |
| (-51.89,-45.89,16.5)/L Extra-Nuclear | 33 | 36 (1.45) | 35 (8.01) |
| (12.03,16.41,-1.39)/L Insula | 34 | 32 (1.29) | 30 (6.86) |
| (54.01,19.46,13.7)/Inter-Hemispheric | 35 | 32 (1.29) | 29 (6.64) |
| (-29.4,-11.36,-4.44)/L Precentral Gyrus | 36 | 30 (1.21) | 28 (6.41) |

R: right hemisphere, L: left hemisphere

Table 2.5 Meta-data cluster results continued

| Individual Foci Clusters | | | |
|---|---|---|---|
| Cluster Centers/ Brain Regions | Cluster Index | # of foci per cluster(% of total foci) | # of studies per cluster(% of all studies) |
| (-9.18,-65.39,-4.97)/L Sub-Gyral | 37 | 30 (1.21) | 26 (5.95) |
| (3.14,20.59,26.82)/R Insula | 38 | 26 (1.05) | 23 (5.26) |
| (-23.15,-18.05,8.45)/R Insula | 39 | 21 (0.85) | 20 (4.58) |
| (-22.9,26.18,7.38)/L Lentiform Nucleus | 40 | 17 (0.69) | 16 (3.66) |
| (47.44,-11.21,18.75)/L Extra-Nuclear | 41 | 13 (0.52) | 13 (2.97) |
| (19.72,-44.58,13.05)/R Insula | 42 | 12 (0.48) | 12 (2.75) |
| (29.06,-57.28,-15.73)/L Extra-Nuclear | 43 | 11 (0.44) | 10 (2.29) |
| (-24.27,27.54,-5.33)/L Cingulate Gyrus | 44 | 10 (0.4) | 10 (2.29) |
| (49.2,14.09,33.83)/L Extra-Nuclear | 45 | 5 (0.2) | 5 (1.14) |
| (-8.85,-29.91,2.45)/L Extra-Nuclear | 46 | 3 (0.12) | 3 (0.69) |
| (20.87,20.34,-4.63)/R Extra-Nuclear | 47 | 2 (0.08) | 2 (0.46) |
| (42.09,-57.93,-22.84)/R Extra-Nuclear | 48 | 2 (0.08) | 2 (0.46) |
| (21.63,-11.55,-4.57)/L Thalamus | 49 | 2 (0.08) | 2 (0.46) |
| (-16.54,-47.93,2.98)/L Parahippocampal Gyrus | 50 | 2 (0.08) | 2 (0.46) |
| (45.29,-12.79,1.23)/R Insula | 51 | 2 (0.08) | 2 (0.46) |
| (-19.18,3.8,19.44)/R Culmen | 52 | 1 (0.04) | 1 (0.23) |
| (-37.84,-6.45,0.55)/R Extra-Nuclear | 53 | 1 (0.04) | 1 (0.23) |

R: right hemisphere, L: left hemisphere

Table 2.6 Breakdown of emotions and their frequencies by individual foci cluster*

| Cluster Index: Total foci in that cluster<br>Emotion Frequency of emotion (% of total cluster foci) | | | | | |
|---|---|---|---|---|---|
| Cluster: 1 | 183 | Cluster: 2 | 149 | Cluster: 3 | 126 |
| aff | 62 (33.88) | aff | 56 (37.58) | aff | 49 (38.89) |
| anger | 9 (4.92) | anger | 10 (6.71) | anger | 8 (6.35) |
| disgust | 21 (11.48) | disgust | 19 (12.75) | disgust | 11 (8.73) |
| **fear** | **45 (24.59)** | **fear** | **24 (16.11)** | fear | 18 (14.29) |
| happy | 11 (6.01) | happy | 7 (4.7) | happy | 9 (7.14) |
| mixed | 14 (7.65) | mixed | 14 (9.4) | mixed | 10 (7.94) |
| sad | 21 (11.48) | sad | 19 (12.75) | **sad** | **21 (16.67)** |
| Cluster: 4 | 112 | Cluster: 5 | 90 | Cluster: 6 | 86 |
| aff | 33 (29.46) | aff | 32 (35.56) | aff | 32 (37.21) |
| anger | 10 (8.93) | anger | 5 (5.56) | anger | 5 (5.81) |
| disgust | 15 (13.39) | disgust | 9 (10) | **disgust** | **18 (20.93)** |
| fear | 18 (16.07) | fear | 9 (10) | fear | 10 (11.63) |
| happy | 9 (8.04) | happy | 6 (6.67) | happy | 7 (8.14) |
| mixed | 7 (6.25) | mixed | 5 (5.56) | mixed | 6 (6.98) |
| **sad** | **20 (17.86)** | **sad** | **23 (25.56)** | sad | 8 (9.3) |
| | | surprise | 1 (1.11) | | |
| Cluster: 7 | 86 | Cluster: 8 | 82 | Cluster: 9 | 74 |
| aff | 29 (33.72) | aff | 32 (39.02) | aff | 20 (27.03) |
| anger | 8 (9.3) | anger | 4 (4.88) | anger | 5 (6.76) |
| disgust | 11 (12.79) | disgust | 9 (10.98) | **disgust** | **16 (21.62)** |
| **fear** | **12 (13.95)** | fear | 5 (6.1) | fear | 8 (10.81) |
| happy | 9 (10.47) | **happy** | **16 (19.51)** | happy | 7 (9.46) |
| mixed | 8 (9.3) | mixed | 4 (4.88) | mixed | 9 (12.16) |
| sad | 9 (10.47) | sad | 12 (14.63) | sad | 8 (10.81) |
| | | | | surprise | 1 (1.35) |
| Cluster: 10 | 74 | Cluster: 11 | 66 | Cluster: 12 | 64 |
| aff | 29 (39.19) | aff | 20 (30.3) | aff | 22 (34.38) |
| anger | 4 (5.41) | anger | 9 (13.64) | anger | 5 (7.81) |
| disgust | 7 (9.46) | disgust | 8 (12.12) | disgust | 7 (10.94) |
| fear | 7 (9.46) | **fear** | **10 (15.15)** | fear | 7 (10.94) |
| happy | 5 (6.76) | happy | 5 (7.58) | happy | 9 (14.06) |
| mixed | 5 (6.76) | mixed | 7 (10.61) | mixed | 3 (4.69) |
| **sad** | **17 (22.97)** | sad | 7 (10.61) | **sad** | **10 (15.62)** |
| | | | | surprise | 1 (1.56) |

*aff:affective

Table 2.7 Breakdown of emotions continued*

| Cluster Index: Total foci in that cluster<br>Emotion Frequency of emotion (% of total cluster foci) | | | | | |
|---|---|---|---|---|---|
| Cluster: 13 | 63 | Cluster: 14 | 62 | Cluster: 15 | 60 |
| aff | 20 (31.75) | aff | 23 (37.1) | aff | 21 (35) |
| anger | 7 (11.11) | anger | 4 (6.45) | anger | 3 (5) |
| disgust | 6 (9.52) | disgust | 7 (11.29) | disgust | 9 (15) |
| **fear** | **10 (15.87)** | **fear** | **15 (24.19)** | fear | 9 (15) |
| happy | 4 (6.35) | happy | 2 (3.23) | happy | 2 (3.33) |
| **mixed** | **10 (15.87)** | mixed | 6 (9.68) | mixed | 5 (8.33) |
| sad | 6 (9.52) | sad | 5 (8.06) | **sad** | **11 (18.33)** |
| Cluster: 16 | 59 | Cluster: 17 | 57 | Cluster: 18 | 57 |
| aff | 22 (37.29) | aff | 17 (29.82) | aff | 15 (26.32) |
| anger | 4 (6.78) | anger | 4 (7.02) | anger | 4 (7.02) |
| disgust | 8 (13.56) | disgust | 8 (14.04) | **disgust** | **13 (22.81)** |
| **fear** | **9 (15.25)** | fear | 8 (14.04) | fear | 7 (12.28) |
| happy | 6 (10.17) | happy | 2 (3.51) | happy | 5 (8.77) |
| mixed | 2 (3.39) | mixed | 4 (7.02) | mixed | 2 (3.51) |
| sad | 8 (13.56) | **sad** | **12 (21.05)** | sad | 11 (19.3) |
| | | surprise | 2 (3.51) | | |
| Cluster: 19 | 55 | Cluster: 20 | 54 | Cluster: 21 | 50 |
| aff | 27 (49.09) | aff | 21 (38.89) | aff | 18 (36) |
| anger | 3 (5.45) | anger | 4 (7.41) | anger | 2 (4) |
| **disgust** | **10 (18.18)** | disgust | 5 (9.26) | disgust | 4 (8) |
| fear | 3 (5.45) | **fear** | **13 (24.07)** | **fear** | **11 (22)** |
| happy | 1 (1.82) | happy | 5 (9.26) | happy | 2 (4) |
| mixed | 2 (3.64) | mixed | 4 (7.41) | mixed | 4 (8) |
| sad | 9 (16.36) | sad | 2 (3.7) | sad | 9 (18) |
| Cluster: 22 | 49 | Cluster: 23 | 48 | Cluster: 24 | 47 |
| aff | 19 (38.78) | aff | 23 (47.92) | aff | 16 (34.04) |
| anger | 3 (6.12) | anger | 4 (8.33) | anger | 3 (6.38) |
| **disgust** | **9 (18.37)** | **disgust** | **7 (14.58)** | disgust | 10 (21.28) |
| fear | 6 (12.24) | fear | 6 (12.5) | **fear** | **11 (23.4)** |
| happy | 2 (4.08) | happy | 2 (4.17) | happy | 2 (4.26) |
| mixed | 4 (8.16) | sad | 6 (12.5) | mixed | 1 (2.13) |
| sad | 6 (12.24) | | | sad | 4 (8.51) |

*aff:affective

Table 2.8 Breakdown of emotions continued*

| Cluster Index: Total foci in that cluster<br>Emotion Frequency of emotion (% of total cluster foci) | | | | | |
|---|---|---|---|---|---|
| Cluster: 25 | 46 | Cluster: 26 | 45 | Cluster: 27 | 45 |
| aff | 18 (39.13) | aff | 18 (40) | aff | 19 (42.22) |
| anger | 3 (6.52) | anger | 6 (13.33) | **disgust** | **11 (24.44)** |
| disgust | 6 (13.04) | disgust | 1 (2.22) | fear | 4 (8.89) |
| **fear** | **9 (19.57)** | fear | 4 (8.89) | happy | 3 (6.67) |
| happy | 4 (8.7) | happy | 4 (8.89) | mixed | 4 (8.89) |
| mixed | 3 (6.52) | mixed | 5 (11.11) | sad | 4 (8.89) |
| sad | 3 (6.52) | **sad** | **7 (15.56)** | | |
| Cluster: 28 | 44 | Cluster: 29 | 42 | Cluster: 30 | 39 |
| aff | 15 (34.09) | aff | 14 (33.33) | aff | 13 (33.33) |
| anger | 5 (11.36) | anger | 2 (4.76) | anger | 3 (7.69) |
| **disgust** | **10 (22.73)** | **disgust** | **7 (16.67)** | **disgust** | **7 (17.95)** |
| fear | 4 (9.09) | **fear** | **7 (16.67)** | fear | 6 (15.38) |
| happy | 2 (4.55) | happy | 2 (4.76) | mixed | 5 (12.82) |
| mixed | 4 (9.09) | mixed | 2 (4.76) | sad | 5 (12.82) |
| sad | 4 (9.09) | **sad** | **7 (16.67)** | | |
| | | surprise | 1 (2.38) | | |
| Cluster: 31 | 37 | Cluster: 32 | 37 | Cluster: 33 | 36 |
| aff | 16 (43.24) | aff | 13 (35.14) | aff | 11 (30.56) |
| anger | 1 (2.7) | anger | 1 (2.7) | anger | 1 (2.78) |
| disgust | 5 (13.51) | disgust | 5 (13.51) | **disgust** | **7 (19.44)** |
| fear | 2 (5.41) | fear | 5 (13.51) | fear | 4 (11.11) |
| happy | 1 (2.7) | happy | 4 (10.81) | happy | 2 (5.56) |
| **mixed** | **6 (16.22)** | mixed | 3 (8.11) | mixed | 5 (13.89) |
| **sad** | **6 (16.22)** | **sad** | **6 (16.22)** | sad | 6 (16.67) |
| Cluster: 34 | 32 | Cluster: 35 | 32 | Cluster: 36 | 30 |
| aff | 11 (34.38) | aff | 13 (40.62) | aff | 9 (30) |
| anger | 4 (12.5) | anger | 3 (9.38) | anger | 2 (6.67) |
| **disgust** | **5 (15.62)** | disgust | 3 (9.38) | **disgust** | **5 (16.67)** |
| **fear** | **5 (15.62)** | fear | 2 (6.25) | **fear** | **5 (16.67)** |
| happy | 3 (9.38) | happy | 2 (6.25) | happy | 3 (10) |
| mixed | 1 (3.12) | mixed | 3 (9.38) | mixed | 1 (3.33) |
| sad | 3 (9.38) | **sad** | **6 (18.75)** | **sad** | **5 (16.67)** |
| Cluster: 37 | 30 | Cluster: 38 | 26 | Cluster: 39 | 21 |
| aff | 14 (46.67) | aff | 4 (15.38) | aff | 3 (14.29) |
| **disgust** | **4 (13.33)** | anger | 3 (11.54) | anger | 2 (9.52) |
| **fear** | **4 (13.33)** | disgust | 2 (7.69) | disgust | 3 (14.29) |
| **happy** | **4 (13.33)** | fear | 2 (7.69) | **fear** | **5 (23.81)** |
| **sad** | **4 (13.33)** | happy | 2 (7.69) | happy | 4 (19.05) |
| | | mixed | 5 (19.23) | mixed | 4 (19.05) |
| | | **sad** | **8 (30.77)** | | |

*aff:affective

Table 2.9 Breakdown of emotions continued*

| Cluster Index: Total foci in that cluster<br>Emotion Frequency of emotion (% of total cluster foci) | | | | | |
|---|---|---|---|---|---|
| Cluster: 40 | 17 | Cluster: 41 | 13 | Cluster: 42 | 12 |
| aff | 6 (35.29) | aff | 7 (53.85) | aff | 3 (25) |
| anger | 1 (5.88) | **fear** | **3 (23.08)** | disgust | 1 (8.33) |
| disgust | 1 (5.88) | **mixed** | **3 (23.08)** | **fear** | **4 (33.33)** |
| fear | 1 (5.88) | | | happy | 1 (8.33) |
| happy | 2 (11.76) | | | mixed | 1 (8.33) |
| mixed | 2 (11.76) | | | sad | 2 (16.67) |
| **sad** | **4 (23.53)** | | | | |
| Cluster: 43 | 11 | Cluster: 44 | 10 | Cluster: 45 | 5 |
| aff | 4 (36.36) | aff | 6 (60) | aff | 2 (40) |
| **disgust** | **3 (27.27)** | **disgust** | **3 (30)** | **anger** | **1 (20)** |
| **fear** | **3 (27.27)** | sad | 1 (10) | **mixed** | **1 (20)** |
| sad | 1 (9.09) | | | **sad** | **1 (20)** |
| Cluster: 46 | 3 | Cluster: 47 | 2 | Cluster: 48 | 2 |
| aff | 1 (33.33) | **mixed** | **1 (50)** | aff | 1 (50) |
| **fear** | **2 (66.67)** | **sad** | **1 (50)** | **fear** | **1 (50)** |
| Cluster: 49 | 2 | Cluster: 50 | 2 | Cluster: 51 | 2 |
| aff | 1 (50) | **disgust** | **1 (50)** | **fear** | **2 (100)** |
| **fear** | **1 (50)** | **fear** | **1 (50)** | | |
| Cluster: 52 | 1 | Cluster: 53 | 1 | | |
| **anger** | **1 (100)** | **aff** | **1 (100)** | | |

*aff:affective

Table 2.10 Breakdown of emotions and their frequencies by individual foci cluster for ROI*

| Cluster Index: Total foci in that cluster<br>Emotion Frequency of emotion (% of total cluster foci) | | | | | |
|---|---|---|---|---|---|
| Cluster: 1 | 74 | Cluster: 2 | 51 | Cluster: 3 | 48 |
| aff | 25 (33.78) | aff | 21 (41.18) | aff | 20 (41.67) |
| anger | 2 (2.7) | anger | 4 (7.84) | anger | 2 (4.17) |
| disgust | 13 (17.57) | **disgust** | **11 (21.57)** | disgust | 4 (8.33) |
| **fear** | **14 (18.92)** | fear | 8 (15.69) | fear | 6 (12.5) |
| happy | 6 (8.11) | happy | 2 (3.92) | happy | 3 (6.25) |
| mixed | 5 (6.76) | mixed | 2 (3.92) | mixed | 1 (2.08) |
| sad | 9 (12.16) | sad | 3 (5.88) | **sad** | **12 (25)** |
| Cluster: 4 | 35 | Cluster: 5 | 25 | Cluster: 6 | 24 |
| aff | 16 (45.71) | aff | 12 (48) | aff | 10 (41.67) |
| anger | 3 (8.57) | disgust | 3 (12) | disgust | 2 (8.33) |
| disgust | 2 (5.71) | **fear** | **5 (20)** | **happy** | **5 (20.83)** |
| **fear** | **6 (17.14)** | happy | 1 (4) | mixed | 1 (4.17) |
| happy | 4 (11.43) | mixed | 1 (4) | sad | 6 (25) |
| sad | 4 (11.43) | sad | 3 (12) | | |
| Cluster: 7 | 24 | Cluster: 8 | 24 | Cluster: 9 | 23 |
| aff | 12 (50) | aff | 12 (50) | aff | 10 (43.48) |
| anger | 1 (4.17) | anger | 1 (4.17) | anger | 1 (4.35) |
| disgust | 2 (8.33) | disgust | 2 (8.33) | disgust | 2 (8.7) |
| **fear** | **3 (12.5)** | fear | 2 (8.33) | fear | 3 (13.04) |
| **happy** | **3 (12.5)** | happy | 2 (8.33) | happy | 1 (4.35) |
| mixed | 1 (4.17) | **mixed** | **4 (16.67)** | mixed | 2 (8.7) |
| sad | 2 (8.33) | sad | 1 (4.17) | **sad** | **4 (17.39)** |
| Cluster: 10 | 23 | Cluster: 11 | 22 | Cluster: 12 | 22 |
| aff | 11 (47.83) | aff | 9 (40.91) | aff | 8 (36.36) |
| **disgust** | **5 (21.74)** | anger | 2 (9.09) | anger | 2 (9.09) |
| fear | 1 (4.35) | disgust | 3 (13.64) | disgust | 1 (4.55) |
| happy | 1 (4.35) | fear | 2 (9.09) | fear | 3 (13.64) |
| **sad** | **5 (21.74)** | mixed | 1 (4.55) | mixed | 3 (13.64) |
| | | **sad** | **5 (22.73)** | **sad** | **5 (22.73)** |
| Cluster: 13 | 21 | Cluster: 14 | 19 | Cluster: 15 | 19 |
| aff | 7 (33.33) | aff | 11 (57.89) | aff | 5 (26.32) |
| disgust | 1 (4.76) | disgust | 1 (5.26) | anger | 1 (5.26) |
| **fear** | **4 (19.05)** | fear | 1 (5.26) | disgust | 2 (10.53) |
| happy | 3 (14.29) | **sad** | **6 (31.58)** | **fear** | **4 (21.05)** |
| mixed | 3 (14.29) | | | happy | 1 (5.26) |
| sad | 3 (14.29) | | | **mixed** | **4 (21.05)** |
| | | | | sad | 2 (10.53) |

*ROI: region of interest; aff: affective

Table 2.11 Breakdown of emotions and their frequencies by individual foci cluster for ROI continued*

| Cluster Index: Total foci in that cluster<br>Emotion Frequency of emotion (% of total cluster foci) | | | | | |
|---|---|---|---|---|---|
| Cluster: 16 | 17 | Cluster: 17 | 16 | Cluster: 18 | 16 |
| aff | 9 (52.94) | aff | 10 (62.5) | aff | 10 (62.5) |
| anger | 1 (5.88) | anger | 1 (6.25) | fear | 2 (12.5) |
| disgust | 1 (5.88) | disgust | 1 (6.25) | happy | 1 (6.25) |
| **fear** | **4 (23.53)** | **fear** | **2 (12.5)** | **sad** | **3 (18.75)** |
| sad | 2 (11.76) | **sad** | **2 (12.5)** | | |
| Cluster: 19 | 16 | Cluster: 20 | 16 | Cluster: 21 | 16 |
| aff | 8 (50) | aff | 5 (31.25) | aff | 6 (37.5) |
| **disgust** | **2 (12.5)** | anger | 1 (6.25) | **disgust** | **4 (25)** |
| fear | 1 (6.25) | disgust | 2 (12.5) | **fear** | **4 (25)** |
| **happy** | **2 (12.5)** | fear | 3 (18.75) | sad | 2 (12.5) |
| mixed | 1 (6.25) | happy | 1 (6.25) | | |
| **sad** | **2 (12.5)** | **sad** | **4 (25)** | | |
| Cluster: 22 | 15 | Cluster: 23 | 15 | Cluster: 24 | 14 |
| aff | 8 (53.33) | aff | 4 (26.67) | aff | 7 (50) |
| **disgust** | **6 (40)** | anger | 2 (13.33) | **disgust** | **2 (14.29)** |
| mixed | 1 (6.67) | disgust | 2 (13.33) | **fear** | **2 (14.29)** |
| | | fear | 1 (6.67) | happy | 1 (7.14) |
| | | happy | 1 (6.67) | mixed | 1 (7.14) |
| | | mixed | 1 (6.67) | sad | 1 (7.14) |
| | | **sad** | **4 (26.67)** | | |
| Cluster: 25 | 13 | Cluster: 26 | 13 | Cluster: 27 | 13 |
| aff | 5 (38.46) | aff | 5 (38.46) | aff | 7 (53.85) |
| anger | 1 (7.69) | **disgust** | **2 (15.38)** | anger | 1 (7.69) |
| disgust | 2 (15.38) | **fear** | **2 (15.38)** | **disgust** | **3 (23.08)** |
| fear | 1 (7.69) | **mixed** | **2 (15.38)** | sad | 2 (15.38) |
| mixed | 1 (7.69) | **sad** | **2 (15.38)** | | |
| **sad** | **3 (23.08)** | | | | |
| Cluster: 28 | 13 | Cluster: 29 | 13 | Cluster: 30 | 12 |
| aff | 6 (46.15) | aff | 7 (53.85) | aff | 6 (50) |
| anger | 1 (7.69) | anger | 1 (7.69) | disgust | 1 (8.33) |
| disgust | 2 (15.38) | **disgust** | **2 (15.38)** | **fear** | **5 (41.67)** |
| mixed | 1 (7.69) | fear | 1 (7.69) | | |
| **sad** | **3 (23.08)** | mixed | 1 (7.69) | | |
| | | sad | 1 (7.69) | | |

*ROI: region of interest; aff: affective

Table 2.12 Breakdown of emotions and their frequencies by individual foci cluster for ROI continued*

| Cluster Index: Total foci in that cluster<br>Emotion Frequency of emotion (% of total cluster foci) | | | | | |
|---|---|---|---|---|---|
| Cluster: 31 | 11 | Cluster: 32 | 11 | Cluster: 33 | 11 |
| aff | 6 (54.55) | aff | 6 (54.55) | aff | 1 (9.09) |
| **disgust** | **2 (18.18)** | disgust | 1 (9.09) | disgust | 1 (9.09) |
| fear | 1 (9.09) | fear | 1 (9.09) | fear | 1 (9.09) |
| **sad** | **2 (18.18)** | **sad** | **3 (27.27)** | happy | 2 (18.18) |
| | | | | mixed | 2 (18.18) |
| | | | | **sad** | **4 (36.36)** |
| Cluster: 34 | 10 | Cluster: 35 | 9 | Cluster: 36 | 9 |
| aff | 6 (60) | aff | 4 (44.44) | aff | 2 (22.22) |
| anger | 1 (10) | anger | 1 (11.11) | **anger** | **2 (22.22)** |
| disgust | 1 (10) | fear | 1 (11.11) | disgust | 1 (11.11) |
| **happy** | **2 (20)** | **happy** | **2 (22.22)** | **fear** | **2 (22.22)** |
| | | sad | 1 (11.11) | happy | 1 (11.11) |
| | | | | sad | 1 (11.11) |
| Cluster: 37 | 9 | Cluster: 38 | 6 | Cluster: 39 | 6 |
| aff | 6 (66.67) | aff | 2 (33.33) | aff | 5 (83.33) |
| **disgust** | **3 (33.33)** | disgust | 1 (16.67) | **fear** | **1 (16.67)** |
| | | fear | 1 (16.67) | | |
| | | **sad** | **2 (33.33)** | | |
| Cluster: 40 | 5 | Cluster: 41 | 5 | Cluster: 42 | 4 |
| aff | 2 (40) | aff | 3 (60) | aff | 2 (50) |
| **anger** | **1 (20)** | **disgust** | **2 (40)** | **disgust** | **2 (50)** |
| **disgust** | **1 (20)** | | | | |
| **mixed** | **1 (20)** | | | | |
| Cluster: 43 | 3 | Cluster: 44 | 2 | Cluster: 45 | 2 |
| aff | 2 (66.67) | aff | 1 (50) | aff | 1 (50) |
| **sad** | **1 (33.33)** | **sad** | **1 (50)** | **anger** | **1 (50)** |
| Cluster: 46 | 1 | Cluster: 47 | 1 | Cluster: 48 | 1 |
| **fear** | **1 (100)** | **aff** | **1 (100)** | **disgust** | **1 (100)** |

*ROI: region of interest; aff: affective

# Chapter 3

# Mixture Model

Motivated by the limitations of the spatial Cox process discussed in Chapter 2, this chapter presents a second method to identify areas of brain activation using fMRI meta-data. In this model, the mean of the data is assumed to be a mixture of unknown finite number of components. Conditional on the mean of the data, the distribution of random error satisfies a multivariate normal distribution.

## 3.1 Introduction

Finite mixture models are generally used to model data thought to be grouped or clustered [Stephens, 2000, Aitkin and Rubin, 1985, McLachlan and Basford, 1988]. These mixture components typically share a common parametric family with each component containing different parameters [Stephens, 2000, Aitkin and Rubin, 1985]. Each component also has a mixing proportion or weight that is respective to the frequency of the component in the data as a whole [Stephens, 2000]. Because of the model's ease of implementation, this allows various applications such as pattern recognition, computer vision, signal and image analysis, and machine learning to list a few [Figueiredo and Jain, 2002]. A comprehensive review of applications is in Titterington et al. [1985], McLachlan and Basford [1988], and McLachlan and Peel [2004].

The general form of a finite mixture model is applied to a random sample $\boldsymbol{X} = (X_1^T, ..., X_n^T)$ where $^T$ denotes the transpose and $X_j \in (X_1, ..., X_n)$ denotes a $d$-dimensional random vector. Let $\boldsymbol{x} = (x_1, ..., x_n)$ denote the realization of $\boldsymbol{X}$ with

a probability density $P(x_j)$ on $\Re^d$ such that

$$P(x_j) = \sum_{k=1}^{K} \pi_k f(x_j; \theta_k),$$

where $f(x_j; \theta_k)$ are the densities and their respective parameters, $\theta_k$, of each unique component $k = 1, ..., K$ and $\pi_k$ are the mixing proportions that sum to unity and satisfy $0 \leq \pi_k \leq 1$ for $k = (1, ..., K)$ [McLachlan and Peel, 2004]. The number of finite components may be known or unknown but range from 1 to $n$, with one component resulting in a fully parametric model while more than one component results in a nonparametric distribution, assuming the component distribution is of standard form [McLachlan and Peel, 2004]. Based on a PubMed web search, the most common distribution applied to each component for continuous data is the Gaussian family (univariate or multivariate). For a few examples see Rasmussen [1999], Figueiredo and Jain [2002] and McLachlan and Peel [2004]. Other applicable distributions include the exponential family [Feldmann and Whitt, 1997], the log-normal family [McLaren et al., 1986, McLachlan and Jones, 1988], and the Wiebull family [Zhang et al., 2001]. For discrete data, the most common distribution applied to each component is a Poisson distribution via PubMed search. For some examples see Royle [2004], Leroux and Puterman [1992] and Chen et al. [2001]. Other distributions applicable include the binomial family [Kéry et al., 2005] and negative binomial family [Zhou and Carin, 2013], to name a few. However, components may also be assigned non-parametric distributions such as the Dirichlet process (DP), discussed in detail in Chapter 1, section 1.4.4.1.

The decision of the component density typically depends on the structure of the observed data. Thus, the density decision directly affects how well the model fits the data; if an inappropriate distribution is selected or a standard distribution is unable to accurately explain the behavior of the data, the model may fit poorly. Because standard form distributions are fixed to their distribution pattern, this is a primary advantage of applying the DP. The flexible selection of the base distribution

and precision parameter create a mixture distribution that can better explain non-standard patterns. Furthermore, if outliers are present, the robustness of DP can better model these while mixtures with standard densities can have a difficult time recognizing and handling them [Figueiredo and Jain, 2002]. Mixture parameters may be estimated in a number of ways with the two most popular methods being the expectation maximization (EM) algorithm and Markov chain Monte Carlo methods.

Another issue to consider with mixture models is the estimate of the unknown number of components, $K$. Often times the model fit is sensitive to the number of components identified. If too many components are identified, the model may over-fit the data, while inversely, if the model does not identify enough components, the nature of the data may not accurately described [Figueiredo and Jain, 2002]. Some of the most popular methods for identifying $K$ for mixture models is the EM algorithm [Dempster et al., 1977], the reversible jump Markov chain Monte Carlo (RJMCMC) [Green, 1995, Green and Hastie, 2009], and birth-death process [Preston, 1975, 1976, Stephens, 2000].

The expectation maximization (EM) algorithm was originally introduced in Dempster et al. [1977] and utilizes the maximum likelihood to estimate parameters including the number of componnents, $K$. It treats the individual data clustering assignment as a missing or latent variable $z$ such that $\mathbf{z} = (z_1, ..., z_n)$ [Dempster et al., 1977]. The EM algorithm works in two steps, "E-step" and "M-step". During the "E-step", the conditional expectation of the complete-data log-likelihood given the observed data and current parameter estimates is calculated. The complete-data refers to the observed data and its latent variables. The log-likelihood for the complete data is

$$l(\boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{z}|\boldsymbol{x}) = \sum_{j=1}^{n} \sum_{k=1}^{K} log(\pi_k f(x_j, z_j|\theta_k))$$

with $\boldsymbol{z}$ being estimated by the expectation of

$$Q(\theta, \pi|\theta^{(t)}, \pi^{(t)}) = E\left[\sum_{j=1}^{n} \sum_{k=1}^{K} log(\pi_k P(x_j, z_j|\theta_k))\right].$$

During the "M-step" the log-likelihood is maximized in terms of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ with $\boldsymbol{z}$ held constant at the values calculated in the "E-step" as illustrated below

$$\boldsymbol{\theta^{(t+1)}} = \underset{\theta}{\operatorname{argmin}}\, Q(\theta, \pi|\theta^{(t)}, \pi^{(t)})$$
$$\boldsymbol{\pi^{(t+1)}} = \underset{\pi}{\operatorname{argmin}}\, Q(\theta, \pi|\theta^{(t)}, \pi^{(t)}).$$

The advantage to the EM algorithm is it's easy of implementation to any situation where a likelihood can be calculated. However, the EM has potentially slow numerical convergence, local maximum and not global maximum likelihoods may be identified, and uncertainty in how to identify the number of components [Wu, 1983]. Attempts have been made with some progress to rectify the disadvantage of convergence speed by the implementation of constraints such as the expectation conditional maximization (ECM) algorithm that updates each compartment parameter(s) individually and conditionally in the M step [Meng and Rubin, 1993] and the generalized expectation maximization (GEM) that performs the same individual conditional updating but in both the E and M steps [Neal and Hinton, 1998]. Also, because the likelihood function of a mixture model is not always unimodal, the maximum likelihood may converge at a local maximum or at the boundary of the parameter space thus produce parameter estimates that have no logical interpretation [Figueiredo and Jain, 2002]. In addition, the calculation of these likelihoods is conditional on the number of components, which leads to the question of how to choose the number of components and the distinction between these varying choices.

The alternative methods, RJMCMC and birth-death process, for model inferences fall under the Bayesian paradigm. Under this framework, we can simultaneously estimate the number of components and parameter values. It is also worth mentioning that if a DP is assigned as the prior for component distributions, by it's discreteness aspect and influence of the precision parameter, it will naturally determine the number of clusters. The reversible jump MCMC (RJMCMC), introduced in Green [1995], is a type of Metropolis-Hastings (MH) algorithm that simultaneously considers mul-

tiple models with various number of components $k$ and corresponding parameters. In summary, this method allows the simulation of the posterior distribution to vary in dimensions, meaning the number of clusters and corresponding parameters may "jump" from its current model state with $k$ components to another with a larger or smaller number of components. The chance of the model changing states is based on the acceptance probability defined in Green [1995] or Green and Hastie [2009]. The major disadvantage to the RJMCMC is that it can be computationally challenging and intensive when determining the acceptance probability due to a Jacobian calculation [Marin et al., 2005, Green and Hastie, 2009]. Also, inefficient proposal distributions can lead to slow convergence [Green and Hastie, 2009].

The birth-death process, formally introduced in Preston [1976], is very similar to the RJMCMC in that the current model state, including the number of components and corresponding parameters, is allowed to jump throughout the MC chain such that the number of components may increase or decrease. The difference is that at every jump the change is always accepted and the component change is always an increment of one; the number of clusters always increases (a birth) or decrease (a death) by one. Both births or deaths occur continuously over time, but births occur constantly at a rate of $\lambda_b$ while deaths occur at a rate relative to the stationary distribution of the process [Stephens, 2000]. When the state of the model changes (number of components changes) the respective mixture parameters, such as component parameters and mixing proportions, are immediately updated [Marin et al., 2005]. The advantage of the birth-death process in comparison to the RJMCMC is that by modeling each jump as an accepted birth or death, this removes the Jacobian calculation in the acceptable probability needed for the RJMCMC. Also, it's straightforward implementation utilizes the missing data formulation discussed in EM algorithm section above for data component assignment so that identifiable is not an issue [Marin et al., 2005].

Regardless of the estimation method for $K$, it is important to statistical differentiate between model fits of different models with different number of clusters and parameter estimates. In order decipher the optimal model, a selection criterion can be applied. For example, a set of candidate models may be obtained via post-EM algorithm implementation, each corresponding to a different number of components. In order to determine which model contains the optimal number of components, each is given a selection criteria that can generally be described as

$$\mathcal{C}(\hat{\theta}(k), k) = -logP(\boldsymbol{x}|\hat{\theta}(k)) + \mathcal{P}(k),$$

where $P(\boldsymbol{x}|\hat{\theta}(k))$ is the likelihood conditional on $\hat{\theta}(k)$ which is an estimate of the model parameters for $k$ components and $\mathcal{P}(k)$ is an increasing function penalizing higher values of $k$ to prevent model over-fitting [Figueiredo and Jain, 2002]. The smaller the selection criterion, the better the model fit to the data. Some examples of selection criterions are Laplace-empirical criterion (LEC) [McLachlan and Peel, 2000], Bayesian inference criterion (BIC) [Schwarz et al., 1978], minimum description length (MDL) [Rissanen, 1989], Akaike's information criterion (AIC) [Akaike, 1987], classification likelihood criterions (CLC) [Biernacki and Govaert, 1997], and integrated classification likelihood (ICL) [Biernacki et al., 2000], to name a few. According to McLachlan and Peel [2000] the ICL and LEC outperform all other criteria mentioned above. One last selection criterion worth mentioning is the deviance information criterion (DIC) discussed in Chapters 1 and 2.

In this chapter, we examine an application of a mixture of Dirichlet processes to estimate the expected value of our observed data after adjusting for a study effect. More specifically, we model our observed data as a linear association with it's respective study effect, individual foci cluster effect, and some standard multivariate normal random error with the individual foci cluster effect modeled as a mixture of unknown number of Dirichlet processes. We elect to utilize the birth-death process to make model inferences. This is motivated from the limitation of the spatial Cox

process application and it's inability to statistically distinguish between clusters. In that application we modeled the individual foci cluster effect a single DP. However, by modeling the clusters in this manner, there was no statistical determination in the number of clusters. In other words, there was no statistical differentiation between clusters and to check whether two clusters are actually a single cluster with two modes. Therefore, in this model we let the individual foci clusters be modeled as a mixture of DPs and incorporate a statistical determination of the number of clusters by exercising the birth-death process while still allowing flexibility by modeling the components as DPs.

The remainder of this section is dedicated to further reviewing the birth-death process and an algorithm introduced in Stephens [2000] to assist in it's implementation. The next section will explicitly layout the format of the model, it's notation, and discuss the Bayesian framework under which parameters are estimated. Following, we present two simulated datasets to asses the sensitivity and robustness of the proposed method, those results, and a comparison with the same datasets analyzed in Chapter 2. Section 3.4 reanalyzes the fMRI meta-analysis discussed in Chapter 2 and discusses those results, implications, and comparison with the results from the spatial Cox process. Section 3.5 summarizes the conclusions and a discussion of the proposed method.

The birth-death process: As briefly discussed in Chapter 1 and above, a mixture model assumes the data $\boldsymbol{x} = (x_1, ..., x_n)$ is able to be partitioned into $K$ number of components that each share a common distribution family $f(\cdot; \phi_k, \eta)$ with an associated parameter $\phi_k$. Let $\boldsymbol{\phi} = (\phi_1, ..., \phi_K, \eta)$ be a vector of cluster parameters and $\eta$ is a common parameter to all components. The likelihood for the mixture is:

$$P(\boldsymbol{x}|\boldsymbol{\phi}) = \prod_{j=1}^{n} \sum_{k=1}^{K} \pi_k f(x_j; \phi_k, \eta)$$

where $\pi_k$ is the mixing proportion of each cluster such that $\sum_{k=1}^{K} \pi_k = 1$ and let $\boldsymbol{\pi} = (\pi_1, ..., \pi_K)$. This parameter represents the multinominal probability of an observation

being assigned to a particular cluster [Stephens, 2000].

The birth-death process is a type of continuous-time Markov chain originally introduced in Preston [1975]. This type of process is often used to simulate realizations of point processes as they can be difficult to directly sample from [Stephens, 2000]. These realizations are then further used for likelihood inferences for model parameters [Stephens, 2000]. The birth-death scheme allows events to randomly occur throughout the chain; these events are either a "birth" or "death". If a birth occurs, the number of components increases by one while if a death occurs, the number of components decreases by one.

Considering a finite mixture for data $\mathbf{x} = (x_1, ..., x_n)$ that are assumed independently distributed with each generated from one of the $K$ distributions, $f(x; \phi_1, \eta)$,..., $f(x; \phi_K, \eta)$, i.e.,

$$P(x|\boldsymbol{\pi}, \boldsymbol{\phi}, \eta) = \pi_1 f(x; \phi_1, \eta) + \cdots + \pi_k f(x; \phi_k, \eta)$$

where $K$ is unknown, $\boldsymbol{\pi} = (\pi_1, ..., \pi_K)$ are the mixing proportions, $\boldsymbol{\phi} = (\phi_1, ..., \phi_K)$ are the component specific parameters and $\eta$ is a common parameter to all components. In addition, an index variable $z_j,\ j = 1, ..., n$, is introduced to indicate the component assignment of observation $j$ and $Z_j$ takes the values of 1 to $K$. Denoted $z_j \in \boldsymbol{z}$ such that $\boldsymbol{z} = (z_1, ..., z_n)$ represents the realization of independent and identically distributed discrete random variables $\boldsymbol{Z} = (Z_1, ..., Z_n)$ with probability mass function

$$P(Z_j = k|\boldsymbol{\pi}, \boldsymbol{\phi}, \eta) = \pi_k \ (j = 1, ..., n; k = 1, ..., K).$$

Given the component assignment, the $x_j$'s are independently distributed with density $P(x_j; \phi_k, \eta)$. The likelihood function is then given as

$$L(K, \boldsymbol{\pi}, \boldsymbol{\phi}, \eta) = \prod_{j=1}^{n} \left[\pi_1 f(x_j; \phi_1, \eta) + \cdots + \pi_k f(x_j; \phi_k, \eta)\right].$$

To carry out a Bayesian inference, prior distributions are assigned to $K$, $\boldsymbol{\pi}$, and $\boldsymbol{\phi}$, denoted by $P(K, \boldsymbol{\pi}, \boldsymbol{\phi})$. The joint posterior distribution, up to a normalizing constant,

is

$$P(K, \boldsymbol{\pi}, \boldsymbol{\phi}|\boldsymbol{x}, \eta) \propto L(K, \boldsymbol{\pi}, \boldsymbol{\phi}, \eta)P(K, \boldsymbol{\pi}, \boldsymbol{\phi}).$$

From here, the birth-death algorithm and Markov chain can be described.

1. Starting with the initial model $y = \{(\pi_1, \phi_1), ..., (\pi_K, \phi_K)\}$ let the birth rate $\beta(y) = \lambda_b$.

2. Calculate the death rate for each component:

$$\delta_k(y) = \frac{L(y \backslash (\pi_k, \phi_k))}{L(y)} \frac{P(K-1|\eta, \cdot)}{KP(K|\eta, \cdot)} (k = 1, ..., K).$$

3. Calculate the total death rate $\delta(y) = \sum_{k=1}^{K} \delta_k(y)$. To quicken convergence, we elected not to model the time to next jump as exponential and allowed an event to occur at each iteration of the Markov chain.

4. Simulate the type of event, birth or death with the respective probabilities

$$P(birth) = \frac{\beta(y)}{\beta(y) + \delta(y)}, P(death) = \frac{\delta(y)}{\beta(y) + \delta(y)}.$$

5. Adjust the model $y$ to reflect the birth or death by

   - Birth: Simulate new component $(\pi_{K+1}, \phi_{K+1})$ from each parameters respective (independent) prior distributions, $\pi_{K+1}$ from $Dir(\boldsymbol{\gamma} = 1) \propto K(1-\pi)^{K-1}$ and $\phi_{K+1}$ from it's prior distribution $\tilde{P}(\phi|\eta, \cdot)$ such that the model becomes $y = \{(\pi_1, \phi_1), ..., (\pi_K, \phi_K), (\pi_{K+1}, \phi_{K+1})\}$. It can be mentioned that $K(1-\pi)^{K-1}$ is the Beta distribution with parameters $(1, K)$ and can be easily simulated from $Y_1 \sim \Gamma(1, 1)$ and $Y_2 \sim \Gamma(K, 1)$ such that $\pi_{K+1} = \frac{Y_1}{Y_1+Y_2}$.
   - Death: Select a component to die with the probabilities $\delta_k(y)/\delta(y)$ for $k = 1, ..., K$ such that the model becomes $y = \{(\pi_1, \phi_1), ..., (\pi_{K-1}, \phi_{K-1})\}$.

6. Given the current state of the model at time $t$, simulate values for all parameters:

- Sample $(\boldsymbol{z})^{(t+1)}$ from $P(\boldsymbol{z}|K^{(t+1)}, \boldsymbol{\pi}^{(t)}, \boldsymbol{\phi}^{(t)}, \eta^{(t)}, \cdot^{(t)}, \boldsymbol{x})$
- Sample $\eta^{(t+1)}, \ldots^{(t+1)}$ from $P(\eta, \cdot|K^{(t+1)}, \boldsymbol{\pi}^{(t)}, \boldsymbol{\phi}^{(t)}, \boldsymbol{x}, \boldsymbol{z}^{(t+1)})$
- Sample $\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\phi}^{(t+1)}$ from $P(\boldsymbol{\pi}, \boldsymbol{\phi}|K^{(t+1)}, \eta^{(t+1)}, \cdot^{(t+1)}, \boldsymbol{x}, \boldsymbol{z}^{(t+1)})$

7. Go to step 2.

## 3.2 The Model

Following similar notation as in Chapter 2, let $s_{rj} = (x, y, z)$ denote a single focus which represents a talairach coordinate defined in the brain space, for study $r$, $r = 1, ..., R$, and the $j^{th}$ foci in study $r$, $j = 1, ..., J_r$. We have $\sum_{r=1}^{R} J_r = n$, where $n$ is the total number of observed foci. Denoted by $\boldsymbol{s} = s_{1,1}...s_{R,J_R}$ represents all foci in the CBMA study. We model $s_{rj}$ as

$$s_{rj} = p_r + \theta_{rj} + \epsilon$$

where $p_r$ denotes the effect of each study $r$, while $\theta_{rj}$ represents the mean of $s_{rj}$ for the $j^{th}$ foci in study $r$ after adjusting for study effect, $p_r$, and $\epsilon$ denotes some random error. By modeling the random error as a standard multivariate normal distribution, the distribution of $s_{rj}$ satisfies,

$$s_{rj} \sim MVN(p_r + \theta_{rj}, \Sigma)$$

with

$$P(s_{rj}|p_r, \theta_{rj}, \Sigma) = (2\pi)^{\frac{3}{2}}|\Sigma|^{\frac{1}{2}} exp\left[-\frac{1}{2}(s_{rj} - p_r - \theta_{rj})'\Sigma^{-1}((s_{rj} - p_r - \theta_{rj}))\right],$$

where $\Sigma = \sigma^2 I_3$ is the covariance matrix .

### 3.2.1 Prior and hyperprior Distributions

Since the foci included in the meta-study are results from different emotions, it is possible that foci from certain types of emotions are clustered together. To fulfill this goal, we assumed the prior of $\theta_{rj}$ is a mixture of DPs, that is, $\theta_{rj} \sim$

$\sum_{k=1}^{K} \pi_k G_k$ where $\boldsymbol{\pi} = (\pi_1, ..., \pi_K)$ is assumed to follow a Dirichlet distribution, $Dir(\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_K)) = \frac{1}{B(\gamma)} \prod_k^K \pi_k^{\gamma_k - 1}$, with $\boldsymbol{\gamma} = 1$ and $G_k \sim DP(\alpha, G_{0k})$, $\alpha$ specified later, $G_{0k} = MVN_3(\boldsymbol{\mu_k}, \Sigma_k)$. Each compartments' hyper-parameters originate from the the same hyper-prior distributions, $\boldsymbol{\mu_k} \sim MVN_3(\boldsymbol{\xi}, \kappa^{-1})$ where $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$, the respective midpoint of observed intervals of variation of the data, and

$$\kappa = \begin{bmatrix} \frac{1}{R_1^2} & 0 & 0 \\ 0 & \frac{1}{R_2^2} & 0 \\ 0 & 0 & \frac{1}{R_3^2} \end{bmatrix}$$

and $\Sigma_k = \sigma_k * I_3$ with $\sigma_k \sim IG(20, 0.5)$. For study effect, $p_r$, we assume $p_r \sim G_p$ where $G_p \sim DP(\alpha_p, G_{0p})$ with $\alpha_p$ specified later and $G_{0p} = MVN_3(\boldsymbol{\mu_p}, \Sigma_p)$. We let $\boldsymbol{\mu_p} \sim MVN_3(\mathbf{0}, I_3)$ and $\Sigma_p = \sigma_p^2 I_3$, with $\sigma_p^2 \sim IG(3, 0.5)$. The variance component of the random error, $\Sigma = \sigma^2 I_3$, $\sigma^2$ is assumed to follow Inverse Gamma (IG) distribution, $\sigma^2 \sim IG(0.5, 0.5)$. In both the component and study DPs, we assume the precision parameters $\alpha$ and $\alpha_p$ are known and discuss their selection in the section "selection of $\alpha$". We assign a truncated Poisson distribution, $P(K) = \frac{\lambda^K}{K!} exp(-\lambda), (K = 1, ..., n)$, as the prior distribution on the number of components, $K$. The birth-death process is conditional on the pre-specified birth rate, $\lambda_b$. This birth rate, that controls how often a "birth" of a new component occurs, was set to $0.2 \times 10^{-4}$.

### 3.2.2 Conditional posterior distributions and posterior computing

Sampling parameter estimates from their posterior distributions can be achieved via Gibbs sampler. The joint posterior distribution is, up to a normalizing constant,

$$\begin{aligned} P(\boldsymbol{\Phi}|\boldsymbol{s}) \quad \propto \quad & \prod_r^R P(p_r|G_p) P(G_p|\alpha, G_{0p}) G_{0p}(p_r; \mu_p, \sigma_p^2) P(\mu_p) P(\sigma_p^2) \prod_j^{J_r} P(s_{rj}|\boldsymbol{\phi}) \\ & \times P(\theta_{rj}|\boldsymbol{z}, \boldsymbol{\pi}, G_1, ..., G_K) \prod_{k=1}^{K} P(G_k|\alpha, G_{0k}) G_{0k}(\theta_{rj}; \mu_k, \sigma_k^2) P(\mu_k) P(\sigma_k^2) \\ & \times P(\boldsymbol{\pi}) P(K) P(\boldsymbol{Z}) P(\sigma^2), \end{aligned}$$

where $\boldsymbol{\Phi} = (p_r, \theta_{rj}, \sigma^2, \alpha_p, \boldsymbol{\mu_p}, \sigma_p^2, K, \boldsymbol{\mu_1}, ..., \boldsymbol{\mu_K}, \sigma_1^2, ..., \sigma_K^2, \alpha, \boldsymbol{\pi})$ is a vector of all estimable parameters. The likelihood for removing cluster $i$ is

$$\begin{aligned} P(\boldsymbol{\Phi}\backslash(\pi_i, G_i)|\boldsymbol{s}) \quad \propto \quad & \prod_r^R P(p_r|G_p)P(G_p|\alpha, G_{0p})G_{0p}(p_r; \mu_p, \sigma_p^2)P(\mu_p)P(\sigma_p^2)\prod_j^{J_r} P(s_{rj}|\boldsymbol{\phi}) \\ & \times P(\theta_{rj}|\boldsymbol{z}, \boldsymbol{\pi}, G_1, ..., G_K)\prod_{k=1}^{K^{(i)}} P(G_k|\alpha, G_{0k})G_{0k}(\theta_{rj}; \mu_k, \sigma_k^2) \\ & \times P(\mu_k)P(\sigma_k^2)P(\boldsymbol{\pi})P(K)P(\boldsymbol{Z})P(\sigma^2), \end{aligned}$$

where $K^{(i)} = i \notin (1, ..., K)$. Given the decision of a birth, the new cluster's parameters $\mu_{K+1}$,$\sigma_{K+1}^2$, and $\pi_{K+1}$ are sampled from their prior distributions,

$$\begin{aligned} \mu_{K+1} \quad & \sim \quad MVN_3(\boldsymbol{\xi}, \kappa^{-1}) \\ \sigma_{K+1}^2 \quad & \sim \quad IG(20, 0.5) \\ \pi_{K+1} \quad & \sim \quad K(1-\boldsymbol{\pi})^{K-1}. \end{aligned}$$

The mixing proportions are adjusted by multiplying all current proportions by $(1 - \pi_{K+1})$ if a birth occurs or dividing by $(1 - \pi_i)$ if a death occurs. To implement the Gibbs sampler, we present the conditional posterior distribution below where "$\cdot$" denotes data and other parameters conditional on. The conditional posterior of $Z_{rj}$ is

$$\begin{aligned} P(Z_{rj} = k|\boldsymbol{\pi}, \theta_{rj}, G_k, \cdot) \quad \propto \quad & P(\theta_{rj}|Z_{rj} = k, G_k, \cdot)P(G_k|\cdot)P(Z_{rj} = k) \\ = \quad & \left\{ \frac{\alpha}{\alpha + n_k - 1} MVN_3(\boldsymbol{\mu_k}, \Sigma_k) + \frac{\sum_{c=1}^{C_k} \delta_c(\theta_{rj})}{\alpha + n_k - 1} \right\} \pi_k \end{aligned}$$

where $c = 1, ..., C_k$ is the number of sub-clusters for cluster $k \in (1, ..., K)$, $n_k$ is the number of foci in cluster $k$, and $\delta_c(\theta_{rj})$ denotes the unit point mass. The conditional posterior distribution of $\boldsymbol{\pi}$ only depends on $\boldsymbol{Z}$.

$$\begin{aligned} P(\boldsymbol{\pi}|\boldsymbol{Z}) \quad \propto \quad & P(\boldsymbol{Z}|\boldsymbol{\pi})P(\boldsymbol{\pi}) \\ = \quad & \frac{1}{B(\boldsymbol{\gamma})}\prod_{k=1}^K \pi_k^{\gamma-1} \times \frac{1!}{n_1! \cdots n_K!}\pi_1^{n_1} \cdots \pi_K^{n_K} \\ \propto \quad & \pi_1^{n_1} \cdots \pi_K^{n_K} \\ \boldsymbol{\pi}|\boldsymbol{Z} \quad \sim \quad & Dir(n_1 + 1, ..., n_K + 1). \end{aligned}$$

$$
\begin{aligned}
P(\theta_{rj}|\cdot) &\propto \prod_{Z_{rj}\in k} P(s_{rj}|p_r,\theta_{rj},\cdot)P(\theta_{rj}|Z_{rj}=k,G_k)P(G_k|G_{0k},\alpha)G_{0k}(\theta_{rj};\mu_k,\sigma_k^2) \\
&= \prod_{Z_{rj}\in k}\Big\{MVN_3(p_r+\theta_{rj},\sigma^2 I_3)\Big\}\Big\{\frac{\alpha}{\alpha+n_k-1}MVN_3(\boldsymbol{\mu_k},\sigma_k^2 I_3) \\
&\quad +\frac{\sum_{c=1}^{C_k}\delta_c(\theta_{rj})}{\alpha+n_k-1}\Big\} \\
&\propto exp\left[-\frac{1}{2}\left\{\sum_{Z_{rj}\in k}\frac{(s_{rj}-p_r-\theta_{rj})'(s_{rj}-p_r-\theta_{rj})}{\sigma^2}\right.\right. \\
&\quad \left.\left.+\frac{(\theta_{rj}-\boldsymbol{\mu_k})'(\theta_{rj}-\boldsymbol{\mu_k})}{\sigma_k^2}\right\}\right] \\
\theta_{rj}|\cdot &\sim MVN_3((\sigma_k^{-2}+n_k\sigma^{-2})^{-1}(\sigma_k^{-2}\boldsymbol{\mu_k}+n_k\sigma^{-2}\overline{(s_{rj}-p_r)}),(\sigma_k^{-2}+n_k\sigma^{-2})^{-1}I_3),
\end{aligned}
$$

where $\delta_c(\theta_{rj})$ denotes the unit point mass and $n_{k,c}$ are the number of foci in some cluster $k \in (1,...,K)$ and sub-cluster $c \in (C_1,...,C_k)$,

$$
\begin{aligned}
P(\boldsymbol{\mu_k}|\cdot) &\propto \prod_{Z_{rj}\in k} P(\theta_{rj}|Z_{rj}=k,G_{0k},\boldsymbol{\mu_k},\sigma_k^2)G_{0k}(\theta_{rj};\mu_k,\sigma_k^2)P(\boldsymbol{\mu_k}) \\
&= exp\left[-\frac{1}{2}\left\{\sum_{Z_{rj}\in k}\frac{(\theta_{rj}-\boldsymbol{\mu_k})'(\theta_{rj}-\boldsymbol{\mu_k})}{\sigma_k^2}+(\boldsymbol{\mu_k}-\boldsymbol{\xi})'\kappa(\boldsymbol{\mu_k}-\boldsymbol{\xi})\right\}\right] \\
\boldsymbol{\mu_k}|\cdot &\sim MVN_3((\kappa+n_k\sigma_k^{-2}I_3)^{-1}(\kappa\boldsymbol{\xi}+n_k\sigma_k^{-2}I_3\bar{\theta}_{rj}),(\kappa+n_k\sigma_k^{-2}I_3)^{-1}),
\end{aligned}
$$

where the notation $\bar{}$ denotes the mean, and

$$
\begin{aligned}
P(\sigma_k^2|\cdot) &\propto \prod_{Z_{rj}\in k} P(\theta_{rj}|Z_{rj}=k,G_k)P(G_k|G_{0k})G_{0k}(\theta_{rj};\mu_k,\sigma_k^2)P(\sigma_k^2) \\
&= exp\left[-\frac{1}{2}\left\{\sum_{Z_{rj}\in k}\frac{(\theta_{rj}-\boldsymbol{\mu_k})'(\theta_{rj}-\boldsymbol{\mu_k})}{\sigma_k^2}+\right\}+\left(-\frac{3}{\sigma_k^2}\right)\right] \\
\sigma_k^2|\cdot &\sim IG(\frac{n_k}{2}+3,0.5+\frac{1}{2}\sum_{Z_{rj}\in k}(\theta_{rj}-\boldsymbol{\mu_k})'(\theta_{rj}-\boldsymbol{\mu_i})),
\end{aligned}
$$

where $n_k$ is the number of foci in cluster $k$. The conditional posterior of $p_r$ is still a DP,

$$\begin{aligned}
P(p_r|\cdot) &\propto \prod_{r=1}^{R} P(s_{rj}|p_r,\theta_{rj},\cdot)P(\theta_{rj}|\cdot)P(p_r|G_p)P(G_p|G_{0p},\alpha_p)G_{0p}(p_r;\mu_p,\sigma_p^2) \\
&= \prod_{r=1}^{R}\left\{MVN_3(p_r+\theta_{rj},\sigma^2 I_3)\right\}\left\{\frac{\alpha_p}{\alpha_p+R}MVN_3(\boldsymbol{\mu_p},\sigma_p^2 I_3)+\frac{\sum_{c=1}^{C_p}\delta_c(p_r)}{\alpha_p+R}\right\}, \\
&\propto exp\left[-\frac{1}{2}\left\{\sum_{r\in R}\frac{(s_{rj}-p_r-\theta_{rj})'(s_{rj}-p_r-\theta_{rj})}{\sigma^2}\right.\right. \\
&\quad \left.\left.+\frac{(p_r-\boldsymbol{\mu_p})'(p_r-\boldsymbol{\mu_p})}{\sigma_p^2}\right\}\right] \\
p_r|\cdot &\sim MVN_3((\sigma_p^{-2}+R\sigma^{-2})^{-1}(\sigma_p^{-2}\boldsymbol{\mu_p}+R\sigma^{-2}\overline{(s_{rj}-\theta_{rj})}),(\sigma_p^{-2}+R\sigma^{-2})^{-1}I_3).
\end{aligned}$$

The conditional posterior distribution for the related hyper-parameters are,

$$\begin{aligned}
P(\boldsymbol{\mu_p}|\cdot) &\propto \prod_{r=1}^{R} P(p_r|G_p)P(G_p|\alpha_p,G_{0p})G_{0p}(p_r;\mu_p,\sigma_p^2)P(\boldsymbol{\mu_p}) \\
&= exp\left[-\frac{1}{2}\left\{\sum_{r=1}^{R}\frac{(p_r-\boldsymbol{\mu_p})'(p_r-\boldsymbol{\mu_p})}{\sigma_p^2}+(\boldsymbol{\mu_p})'(\boldsymbol{\mu_p})\right\}\right] \\
\boldsymbol{\mu_p}|\cdot &\sim MVN_3((1+R\sigma_p^{-2})^{-1}(R\sigma_p^{-2}(\overline{p_r})),(1+R\sigma_p^{-2})^{-1}I_3),
\end{aligned}$$

and

$$\begin{aligned}
P(\sigma_p^2|\cdot) &\propto \prod_{r=1}^{R} P(p_r|G_p)P(G_p|\alpha_p,G_{0p})G_{0p}(p_r;\mu_p,\sigma_p^2)P(\sigma_p^2) \\
&= exp\left[-\frac{1}{2}\left\{\sum_{r=1}^{R}\frac{(p_r-\boldsymbol{\mu_p})'(p_r-\boldsymbol{\mu_p})}{\sigma_p^2}\right\}+\left(-\frac{0.5}{\sigma_p^2}\right)\right] \\
\sigma_p^2|\cdot &\sim IG(3+\frac{R}{2},\frac{\sum_{r=1}^{R}(p_r-\boldsymbol{\mu_p})^2+1}{2}),
\end{aligned}$$

where $C_p$ are the unique study clusters. Lastly, the sampling distribution for $\Sigma$ is:

$$\begin{aligned}
P(\sigma^2|\cdot) &\propto \prod_{r}^{R}\prod_{j}^{J_r} P(s_{rj}|p_r,\theta_{rj},\sigma^2,\cdot)P(\theta_{rj}|\cdot)P(p_r|\cdot)P(\sigma^2) \\
&= exp\left[-\frac{1}{2}\left\{\sum_{Z_{rj}\in n}\frac{(s_{rj}-p_r-\theta_{rj})'(s_{rj}-p_r-\theta_{rj})}{\sigma^2}\right\}+-\frac{0.5}{\sigma^2}\right] \\
\sigma^2|\cdot &\sim IG(\frac{n+1}{2},\frac{1+\sum_{Z_{rj}\in n}(s_{rj}-p_r-\theta_{rj})'(s_{rj}-p_r-\theta_{rj})}{2}).
\end{aligned}$$

Neal's algorithm 8 [Neal, 2000], as discussed in Chapter 2, can still be exercised to sample unique values for $\theta_{rj}$ and $p_r$ for their respective clusters. The value for $\theta_{rj}$ for those foci in cluster $k$ and sub-cluster $c$ may be sampled from

$$\begin{aligned}
P(\theta_{k,c}|\cdot) &\propto \prod_{Z_{rj}\in k,c} P(s_{rj}|p_r,\theta_{rj},Z_{rj}=k,\cdot)P(\theta_{rj}|Z_{rj}=k,G_k)P(G_k|G_{0k},\alpha) \\
&\quad \times G_{0k}(\theta_{k,c};\mu_k,\sigma_k^2) \\
&= \prod_{Z_{rj}\in k,c} MVN_3(p_r+\theta_{rj},\Sigma)\{\frac{\alpha}{\alpha+n_k-1}MVN_3(\boldsymbol{\mu_k},\Sigma_k)+\frac{n_{k,c}}{\alpha+n_k-1}\} \\
&= exp\left[-\frac{1}{2}\left\{\sum_{Z_{rj}\in k,c}\frac{(s_{rj}-p_r-\theta_{rj})'(s_{rj}-p_r-\theta_{rj})}{\sigma^2}\right.\right. \\
&\quad \left.\left.+\frac{(\theta_k-\boldsymbol{\mu_k})'(\theta_k-\boldsymbol{\mu_k})}{\sigma_k^2}\right\}\right] \\
\theta_{k,c}|\cdot &\sim MVN_3((\sigma_k^{-2}+n_{k,c}\sigma^{-2})^{-1}(\sigma_k^{-2}\boldsymbol{\mu_k}+n_{k,c}\sigma^{-2}\overline{(s_{rj}-p_r)}), \\
&\quad (\sigma_k^{-2}+n_k\sigma^{-2})^{-1}I_3),
\end{aligned}$$

where $n_{k,c}$ are the number of foci in some cluster $k$ and sub-cluster $c$. The value for $p_r$ for those studies in cluster $c$ may be sampled from

$$\begin{aligned}
P(p_c|\cdot) &\propto \prod_{r\in c} P(s_{rj}|p_c,c,\cdot)P(p_c|G_p)P(G_p|\alpha_p,G_{0p})G_{0p}(p_c;\mu_p,\sigma_p^2) \\
&= \prod_{p_r\in c} MVN_3(p_r+\theta_{rj},\Sigma)\{\frac{\alpha_p}{\alpha_p+R}MVN_3(\boldsymbol{\mu_p},\Sigma_p)+\frac{n_c}{\alpha_p+R}\} \\
&= exp\left[-\frac{1}{2}\left\{\sum_{r\in c}\frac{(s_{rj}-p_r-\theta_{rj})'(s_{rj}-p_r-\theta_{rj})}{\sigma^2}\right.\right. \\
&\quad \left.\left.+\frac{(p_c-\boldsymbol{\mu_p})'(p_c-\boldsymbol{\mu_p})}{\sigma_p^2}\right\}\right] \\
p_c|\cdot &\sim MVN_3((\sigma_p^{-2}+n_c\sigma^{-2})^{-1}(\sigma_p^{-2}\boldsymbol{\mu_p}+n_c\sigma^{-2}\overline{(s_{rj}-\theta_{rj})}),(\sigma_p^{-2}+n_c\sigma^{-2})^{-1}I_3),
\end{aligned}$$

where $n_c$ is the number of studies in cluster $c$.

### 3.2.3 Determining the clusters

To estimate the number of clusters and the center of each cluster and cluster assignment, we implement the least-squared Euclidean distance method introduced in Dahl

[2006]. This method draws inference of clusters based on a set of converged MCMC iterations and chooses one iteration as the final estimates on the clusters and related parameters. This final MCMC iteration is selected due to its smallest Euclidean distance to the expected cluster assignments estimated based on a set of independent converged MCMC iterations. This approach incorporates all clustering information in the MCMC sampling process [Dahl, 2006]. As in Chapter 2, the clustering pattern to be summarized is for individual foci clusters as they are our primary interest. For more details, see section 2.2.3.

### 3.2.4 Selection of $\alpha$

As discussed in Chapter 2 section 2.2.4, the selection of $\alpha$ can have a potentially significant effect on the number of clusters identified due to its direct impact on the aggregation of $G$ about the base distribution, $G_0$. Given the importance of $\alpha$, we decided to select estimates by again implementing the grid search that minimized the deviance information criterion (DIC) [Congdon, 2007]. The DIC is an estimate for the expected deviance that is adjusted for the models complexity as to not overfit the data [Congdon, 2007, Spiegelhalter et al., 2002]. Specifically, DIC was defined as

$$DIC = \overline{D} + \frac{var(D)}{2},$$

where

$$\begin{aligned} \overline{D} &= \frac{1}{T}\sum_{t=1}^{T} D(\boldsymbol{s}, \boldsymbol{\Phi}^{(t)}) \\ D(\boldsymbol{s}, \boldsymbol{\Phi}^{(t)}) &= -2logP(\boldsymbol{s}|\boldsymbol{\Phi}^{(t)}) \\ var(D) &= \frac{1}{T}\sum_{t=1}^{T}(D(\boldsymbol{s}, \boldsymbol{\Phi}^{(t)}) - \overline{D})^2. \end{aligned}$$

In the above, $\boldsymbol{\Phi}^{(t)}$ denotes the parameter estimate(s) at current time $t$ and $\boldsymbol{s}$ is the data. A smaller DIC indicates a better fit of the model.

## 3.3 Simulation Studies

Simulations were utilized to illustrate and asses the proposed method. In total, 50 studies each with 10 foci were considered. Three individual foci clusters are spatially centered at $(1,1,1)^T$, $(2,2,2)^T$, and $(4,4,4)^T$ containing 150, 150, and 200 foci, respectively. Two study clusters are assumed with centers held at $(0.1,0.1,0.1)^T$ and $(0.4,0.4,0.4)^T$ with each including 25 studies (250 foci each). In addition, we considered the following simulation scenarios,

1. We simulate the data for each cluster via multivariate normal with mean set at the individual foci centers and variance $\Sigma = 0.002I_3$. This creates spheres with little variation and we expect the method to have the ability to correctly identify the clusters.

2. The method's ability to cluster in the presence abnormal patterns is an important factor in spatial clustering. The same scenarios as in 1) are followed to simulate individual foci clusters 1 and 2. Cluster 3 is simulated using truncated normal distribution with mean $(4,4,4)^T$ and variance $0.002I_3$ with a lower bound $(1,1,1)^T$.

3. The last scenario is designed to asses the robustness of the method with respect to the distance between and among clusters. To this end, besides $\Sigma = 0.002I_3$, we considered four additional levels of $\Sigma$: $\Sigma = 0.01I_3, 0.05I_3, 0.1I_3$, and $0.2I_3$ representing gradually closer distances among clusters. Other settings are as in scenario 1).

For each setting, we implemented a grid search for a single dataset to estimate values of $\alpha_p$ and $\alpha$ based on the minimization of DIC. We let precision parameter values be 0.1, 0.5, 1, 2, and 5. Based on $\alpha_p$ and $\alpha$ estimates, 100 MC datasets were generated with 300 burn in iterations, 100 working iterations to determine the

probability matrix noted in section 2.2.3, and 100 additional iterations to infer the number of clusters and individual foci cluster centers.

Model assessment consists of three evaluations: sensitivity, specificity, and percentage of correct clustering. Sensitivity is defined per cluster as the proportion of foci that are correctly assigned to that given cluster, Se=TP/(TP+FN) and specificity is defined per cluster as the proportion of foci that are correctly not assigned to a cluster, Sp=TN/(TN+FP). In these definitions, true positive (TP) denotes a focus in that respective cluster is also assigned to that cluster, false negative (FN) denotes a focus in that respective cluster but not assigned to that cluster, true negative (TN) is a focus that is not in the respective cluster and not assigned to that cluster, and false positive (FP) denotes a focus that is not in that respective cluster but assigned to that cluster. Percentage of correct clustering is an overall measure defined as the proportion of foci that are correctly clustered. Note that the definition of correctness takes into account both TP and TN.

Table 3.1 summaries the findings on individual foci cluster identification and the quality of the identified clusters. The method adequately assigned foci to their correct clusters approximately 50% of the time or lower for all simulations. This produced low sensitivity percentages across all scenarios. It had a more difficult time identifying those foci that were in the two clusters centered at $(1, 1, 1)^T$ and $(2, 2, 2)^T$ than the last cluster centered at $(4, 4, 4)^T$. This was indicated by the higher sensitivity percentages for this latter cluster across all settings. Although the model did a poor job at identify which foci belong to the first two clusters, it did an excellent job of deciphering which foci truly did not belong in those clusters, which is indicated by the high specificity percentages for those two clusters. It is worth mentioning that when the model could not properly adjust for study effect, which is suggested by the high standard deviation of the median number of study clusters (larger than 1 across almost all settings) and low average of the correctness rate (roughly 50% or less), the model also had a difficult

time identifying the correct number of individual foci clusters and their centers. The incorrect number of individual clusters is suggested by the high standard deviation for median number of clusters, low sensitivity rates, and low correctness rate.

When compared to the results for these same scenarios analyzed by the spatial Cox process, the spatial Cox process out-performed this method across all measures and scenarios. However, both methods performed poorly when clusters were large and overlapping such as when simulated variances were allowed to be 0.1 and 0.2. This suggest both models lack the sensitivity to identify clusters when there is not distinct grouping within the data. The current proposed method performed more poorly out of the two methods.

## 3.4 Real Data Analysis

For this application, we applied the proposed method to the same meta-analysis dataset as in Chapter 2 (section 2.4). Recall this data consists of a total of 162 neuroimaging publications with 57 PET and 105 fMRI were considered. Among these 162 publications, there were 437 contrasts or studies. Only those foci that were deemed significantly activated by their study specific criteria were included for a total of 2,478 foci. This meta-analysis analyzed emotions and therefore there exist specific brain regions that were of interest to researchers. Foci that lie within these regions were noted. Summary statistics for this data can be seen in tables 2.2 and 2.3.

As with the simulation studies, grid search and DIC were exercised to estimate values for $\alpha_p$ and $\alpha$. Potential precision parameters values were 0.1, 0.5, 1.0, 2.0 and 5. Each combination was performed over 1,300 iterations, 800 of those for burn-in, 400 for the probability matrix calculation, and final 100 to infer individual clusters and their centers. To assist with the magnitude of the likelihood calculations, the data was scaled down by 10.

It was found that the precision parameter combination of $\alpha_p = 0.1$ and $\alpha = 0.1$

produced the smallest DIC. Convergence over 1,300 iterations, with the initial 800 discarded, was checked visibly. Based on the proposed method, we identified two study clusters and three individual foci clusters (Tables 3.2). The break down of the three individual foci clusters by center location, foci frequency and emotion frequency can be seen in Table 3.3. Of the three individual clusters, cluster 1 centered at (30.66, -5.83, -29.58) contained 1646 foci, cluster 2 centered at (26.73, -1.29, -26.38) contained 763 foci, and cluster 3 centered at (0.91, -6.03, -2.52) contained 69 foci. The neutral emotion, affective, was present in all three clusters as well as the emotion of fear being the second dominating emotion. However, when only analyzing those foci that fell within the region of interest, as seen in Table 3.4, the dominating emotion, other than affective, was sadness (clusters 1 and 2) and fear (cluster 3). When compared to the meta-analysis results from Chapter 2 using the spatial Cox process, there were significantly fewer clusters identified and no clusters appeared have similar cluster centers. Following the patterns of disagreement between the simulation studies of the two methods, this is not surprising.

The low number of clusters illustrates the model's lack of sensitivity, especially when the data is not distinctly grouped. This particular data does not visually indicate distinct clusters and is closer to a more uniform distribution throughout the brain which may lead to a smaller number of identifiable clusters. However, it can be inferred from the difference of clusters and simulation studies that the spatial Cox process is a more sensitive method.

## 3.5 Conclusion and Discussion

The modeling of the observed foci as a linear association with study effect, individual foci cluster effect and a standard multivariate normal random error, was motivated by the limitation of the spatial Cox process to statistically distinguish between a cluster and a mode or peak of a cluster. It's overall aim is to identify activated regions

within the brain using fMRI CBMA data. By modeling the data in this fashion, it was hopeful that the distribution could statistical differential between clusters and modes of clusters while retaining the flexibility and robustness to mimick the behavior of the data.

Simulation studies demonstrated that the method does not readily fit data generated from normal or abnormal distributions. Although the model has the ability to correctly identify clusters, it was accurate only about 50% of the time. If study effects were not correctly identified, the method tended to also incorrectly identify individual clusters which resulted in low sensitivity percentages. When compared to the spatial Cox process method, this proposed method was significantly out-performed. However, both methods were unable to correctly identify clusters when they were large and overlapped.

When applied to a fMRI meta dataset, the method identified a very small number of clusters. Given the low sensitivity findings in the simulated studies, it can be concluded that these clusters have a high likelihood of being incorrect. When the same data was analyzed with the spatial Cox process, the difference in the results was extreme. Not only were the number of clusters substantially less, but none of the cluster centers identified from the proposed method came close to those identified in the first method. Given the more favorable simulation results of the spatial Cox process, this further suggests a poor performance by the proposed method. It's worth mentioning that the meta data is not distinctly grouped and is more uniformly distributed throughout the brain. Therefore, there may not be enough variation within the data for the mixture of DPs to identify a large number of clusters. Also, the limitations of this dataset discussed in Chapter 2 (combining fMRI and PET) still hold and would be beneficial to explore this proposed method using additional fMRI data.

The primary advantage to this method, besides it's flexibility, is it's statistical dif-

ferentiation between clusters with different base distribution or a multimodal cluster. Because of its adaptable nature, this model can also adjust for any covariate of interest. However, based on simulation studies and the fMRI meta-data application, the proposed method tends to be insensitive and has a difficult time identifying clusters, especially when the data fails to encompass natural groupings. It's clustering ability is limited by the identification of study effects which may be improved by stronger restrictions. Another potential improvement, also mentioned in the first method, would be to include contrasts that would allow comparisons between foci to better estimate study and individual foci effects.

Table 3.1 Simulation assessments

| Scenario: | Median Num. of Clusters (SD)* | Cluster index | Average Sensitivity (SD)* | Average Specificity (SD)* | Average % Correctness rate (SD)* |
|---|---|---|---|---|---|
| Normal | **IC: 3 (1.12) | 1 | 0.36 (0.43) | 0.92 (0.16) | 0.48 (0.30) |
| | | 2 | 0.27 (0.43) | 0.97 (0.10) | |
| | | 3 | 0.72 (0.28) | 0.59 (0.36) | |
| | **SC: 2 (0.98) | 1 | 0.26 (0.43) | 0.98 (0.09) | 0.57 (0.24) |
| | | 2 | 0.88 (0.16) | 0.35 (0.40) | |
| Skewed | IC: 3 (1.25) | 1 | 0.38 (0.41) | 0.88 (0.18) | 0.46 (0.26) |
| | | 2 | 0.19 (0.39) | 0.99 (0.06) | |
| | | 3 | 0.72 (0.26) | 0.58 (0.34) | |
| | SC: 10 (4.35) | 1 | 0.27 (0.38) | 0.95 (0.09) | 0.38 (0.30) |
| | | 2 | 0.49 (0.27) | 0.73 (0.2) | |
| Large1 | IC: 3 (1.07) | 1 | 0.34 (0.44) | 0.94 (0.15) | 0.49 (0.29) |
| | | 2 | 0.26 (0.43) | 0.97 (0.1) | |
| | | 3 | 0.78 (0.27) | 0.54 (0.39) | |
| | SC: 3 (1.47) | 1 | 0.26 (0.42) | 0.98 (0.07) | 0.54 (0.26) |
| | | 2 | 0.82 (0.19) | 0.42 (0.37) | |
| Large2 | IC: 3 (1.06) | 1 | 0.32 (0.4) | 0.86 (0.21) | 0.43 (0.21) |
| | | 2 | 0.1 (0.3) | 0.99 (0.06) | |
| | | 3 | 0.75 (0.25) | 0.47 (0.36) | |
| | SC: 1 (0.47) | 1 | 0.01 (0.1) | 1 (0.01) | 0.5 (0.05) |
| | | 2 | 0.98 (0.04) | 0.02 (0.1) | |
| Large3 | IC: 2 (1.05) | 1 | 0.11 (0.26) | 0.93 (0.17) | 0.34 (0.09) |
| | | 2 | 0.01 (0.07) | 1 (0.02) | |
| | | 3 | 0.76 (0.26) | 0.29 (0.3) | |
| | SC: 11 (3.06) | 1 | 0.1 (0.15) | 0.94 (0.09) | 0.25 (0.09) |
| | | 2 | 0.4 (0.18) | 0.65 (0.21) | |
| Large4 | IC: 2 (0.92) | 1 | 0.09 (0.23) | 0.93 (0.18) | 0.35 (0.06) |
| | | 2 | 0 (0.05) | 1 (0.04) | |
| | | 3 | 0.79 (0.25) | 0.25 (0.28) | |
| | SC: 1 (0.48) | 1 | 0 (0) | 1 (0) | 0.49 (0.02) |
| | | 2 | 0.98 (0.04) | 0.02 (0.05) | |

*SD: standard deviation across 100 MC replicates;**IC: individual foci cluster; SC: study effect clusters

Table 3.2 Meta-data cluster results

| Individual Foci Clusters | | | |
|---|---|---|---|
| Cluster Centers/ Brain Regions | Cluster Index | # of foci per cluster(% of total foci) | # of studies per cluster(% of all studies) |
| (30.66,-5.83,-29.58)/R Third Ventricle | 1 | 1646 (66.42) | 402 (91.99) |
| (26.73,-1.29,-26.38)/R Uncus | 2 | 763 (30.79) | 320 (73.23) |
| (0.91,-6.03,-2.52)/R Uncus | 3 | 69 (2.78) | 61 (13.96) |

R: right hemisphere, L: left hemisphere

Table 3.3 Breakdown of emotions and their frequencies by individual foci cluster*

| Cluster Index: Total foci in that cluster<br>Emotion Frequency of emotion (% of total cluster foci) | | | | | |
|---|---|---|---|---|---|
| Cluster: 1 | 1646 | Cluster: 2 | 763 | Cluster: 3 | 69 |
| aff | 586 (35.6) | aff | 269 (35.26) | aff | 26 (37.68) |
| anger | 116 (7.05) | anger | 44 (5.77) | anger | 6 (8.7) |
| disgust | 222 (13.49) | disgust | 110 (14.42) | disgust | 5 (7.25) |
| **fear** | **243 (14.76)** | **fear** | **111 (14.55)** | **fear** | **13 (18.84)** |
| happy | 118 (7.17) | happy | 55 (7.21) | happy | 5 (7.25) |
| mixed | 128 (7.78) | mixed | 61 (7.99) | mixed | 6 (8.7) |
| sad | 230 (13.97) | sad | 110 (14.42) | sad | 8 (11.59) |
| surprise | 3 (0.18) | surprise | 3 (0.39) | | |

*aff: affective

Table 3.4 Breakdown of emotions and their frequencies by individual foci cluster for ROI*

| Cluster Index: Total foci in that cluster<br>Emotion Frequency of emotion (% of total cluster foci) | | | | | |
|---|---|---|---|---|---|
| Cluster: 1 | 514 | Cluster: 2 | 247 | Cluster: 3 | 18 |
| aff | 219 (42.61) | aff | 116 (46.96) | aff | 7 (38.89) |
| anger | 26 (5.06) | anger | 7 (2.83) | anger | 1 (5.56) |
| disgust | 66 (12.84) | disgust | 34 (13.77) | disgust | 2 (11.11) |
| fear | 72 (14.01) | fear | 25 (10.12) | **fear** | **3 (16.67)** |
| happy | 33 (6.42) | happy | 10 (4.05) | happy | 2 (11.11) |
| mixed | 24 (4.67) | mixed | 15 (6.07) | mixed | 1 (5.56) |
| **sad** | **74 (14.4)** | **sad** | **40 (16.19)** | sad | 2 (11.11) |

*ROI: region of interest; aff: affective

# Chapter 4

# Conclusion and Future Work

This dissertation has presented two methods that were motivated by the desire to identify regions of significant brain activation using fMRI meta-data. These methods included a spatial Cox point process and multivariate normal mixture of Dirichlet processes model.

FMRI data has proven to be an informative and noninvasive method of accurately measuring the brain functionality with specific locations and intensities over a pre-specified period of time. However, this method is relatively costly and typically only contains a small number of subjects. To offset this disadvantage, meta-data analysis has become very popular for both image-based and coordinate-based fMRI. The latter is not only more readily available in the literature but is easier to handle and combine. Current methods for analyzing coordinate-based fMRI meta-data include activation likelihood estimation (ALE), kernel density analysis (KDA), and spatial point process modeling. The overall disadvantage for these models is that each contains some specified parameter or distribution that directly effects the size, number, and shape of the clusters identified. In both ALE and KDA, voxel based kernels are incorporated that estimate voxel densities with a specified bandwidth measurement. The size of the bandwidth controls the radius of the kernel and directly plays a role in the density calculation. The dimension of the voxel can further alter the kernel density. Thus, both voxel and bandwidth size affect the models' capability and are set by the user. The spatial point process is modeled under the Bayesian framework and allows clustering inferences at the individual, study, and population level. However,

at each of these levels, the distribution is assumed multivariate normal, potentially limiting or missing irregular spatial patterns. Our focus was to present two methods that aim to eradicate user dependency and allow spatial information to play a role in creating more flexible clusters. Specifically, in both methods we incorporated a Diriclet process to model cluster distributions which relaxed the previous methods' restrictions and allowed for more flexible and irregular spatial patterns.

In the first method we proposed, the spatial Cox point process, we clustered on two levels, latent foci center and study activation center. We utilized the Dirichlet process to describe the distribution of foci. The intensity measure associated with the Cox point process was modeled as a function of distance between the focus and the center of the cluster of foci using Gaussian kernels. Simulation studies provided an illustration of the model's sensitivity and robustness with respect to cluster identification and underlying data distributions. We provided an additional demonstration with an fMRI meta-dataset. This clustering method implemented in this model incorporates spatial information via Gaussian kernels while adjusting for some covariate, study effect in this case. Therefore, this method may be extended to any spatial data and may include any number of covariate of interest. The Dirichlet process prior assigned to the foci distribution uniquely identities cluster with regular or irregular patterns. It was noted that when study effect was large, the method had a difficult time identifying the correct individual foci centers. This identifiability issue was partial rectified by placing prior knowledge restrictions on the study effect distribution. By comparing two studies and estimating their differences (use a contrast) may be an alternate solution to this issue. Also, as a natural limitation to the Dirichlet process, the method lacked the ability to statistically differentiate if a peak was a cluster or a cluster with multiple modes. This led to the motivation of the second proposed method.

In our second method we proposed to model the data as a linear association with

a study effect, individual foci effect, and a random error term. The study effect was assumed to follow a Dirichlet process, the individual foci effect was assumed to be a mixture of unknown number of Dirichlet processes, and the random error was modeled as standard multivariate normal. Therefore, the individual foci effect represents the mean of the data after adjusting for the study effect. Simulation studies were conducted to asses the model's sensitivity and robustness and indicated a mediocre performance. When compared with the same simulation studies analyzed by the spatial Cox process, the proposed method was out-performed and produced significantly lower sensitivity measures. However, both methods performed poorly when simulated clusters were large and overlapped. As a further demonstration, we applied the model to the same fMRI meta data analyzed above and identified only a fraction of the clusters with no similar cluster centers. The advantage to this method was it's ability to statistically differentiate between peaks of a distribution as being a cluster or a cluster with multimodes. Although this method was also implemented for modeling irregular shaped distribution patterns, it lacked the sensitivity to properly identify clusters.

# Bibliography


Murray Aitkin and Donald B Rubin. Estimation and hypothesis testing in finite mixture models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 67–75, 1985.

Hirotugu Akaike. Factor analysis and aic. *Psychometrika*, 52(3):317–332, 1987.

Charles E Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, pages 1152–1174, 1974.

Peter A Bandettini, Eric C Wong, R Scott Hinks, Ronald S Tikofsky, and James S Hyde. Time course epi of human brain function during task activation. *Magnetic Resonance in Medicine*, 25(2):390–397, 1992.

Jeffrey D Banfield and Adrian E Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

James C Bezdek and Nikhil R Pal. Some new indexes of cluster validity. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 28(3):301–315, 1998.

Christophe Biernacki and Gérard Govaert. Using the classification likelihood to choose the number of clusters. *Computing Science and Statistics*, pages 451–457, 1997.

Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(7):719–725, 2000.

David A Binder. Bayesian cluster analysis. *Biometrika*, 65(1):31–38, 1978.

Michael GB Blum. Choosing the summary statistics and the acceptance rate in approximate bayesian computation. In *Proceedings of COMPSTAT'2010*, pages 47–56. Springer, 2010.

Hans H Bock. Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis*, 23(1):5–28, 1996.

Matthew Brett, Ingrid S Johnsrude, and Adrian M Owen. The problem of functional localization in the human brain. *Nature Reviews Neuroscience*, 3(3):243–249, 2002.

Jonathan G Campbell, Chris Fraley, Fionn Murtagh, and Adrian E Raftery. Linear flaw detection in woven textiles using model-based clustering. *Pattern Recognition Letters*, 18(14):1539–1548, 1997.

Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. *Pattern recognition*, 28(5):781–793, 1995.

Chibiao Chen, Eric Durand, Florence Forbes, and Olivier François. Bayesian clustering algorithms ascertaining spatial population structure: a new computer program and a comparison study. *Molecular Ecology Notes*, 7(5):747–756, 2007.

Hanfeng Chen, Jiahua Chen, and John D Kalbfleisch. A modified likelihood ratio test for homogeneity in finite mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):19–29, 2001.

Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

D Louis Collins, Peter Neelin, Terrence M Peters, and Alan C Evans. Automatic 3d intersubject registration of mr volumetric data in standardized talairach space. *Journal of computer assisted tomography*, 18(2):192–205, 1994.

Peter Congdon. *Bayesian statistical modelling*, volume 704. John Wiley & Sons, 2007.

Jukka Corander, Patrik Waldmann, and Mikko J Sillanpää. Bayesian analysis of genetic differentiation between populations. *Genetics*, 163(1):367–374, 2003.

Jukka Corander, Patrik Waldmann, Pekka Marttinen, and Mikko J Sillanpää. Baps 2: enhanced possibilities for the analysis of genetic population structure. *Bioinformatics*, 20(15):2363–2369, 2004.

Jukka Corrander, Pekka Marttinen, and Samu Mäntyniemi. A bayesian method for identification of stock mixtures from molecular marker data. *Fishery Bulletin*, 104(4):550–558, 2006.

DR Cox. Some statistical model related with series of events. *Journal of the Royal Statistical Society Series B*, (17):129–164, 1995.

DB Dahl. Model-based clustering for expression data via a dirichlet process mixture model. *Bayesian inference for gene expression and proteomics*, pages 201–218, 2006.

Abhijit Dasgupta and Adrian E Raftery. Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93(441):294–302, 1998.

D. L. Davies and D. W. Bouldin. Şa cluster separation measure. *IEEE Trans. Patt. Anal. Machine Intell.*, PAMI-1:224–227, 1979.

Kevin J Dawson and Khalid Belkhir. A bayesian approach to the identification of panmictic populations and the assignment of individuals. *Genetical research*, 78 (01):59–77, 2001.

AP Dempster, NM Laird, and Donald B Rubin. Maximum likelihood estimation via the em algorithm. *J. of the Stat. Roy. Soc. B*, 39(1):1–38, 1977.

Joseph T Devlin, Richard P Russell, Matt H Davis, Cathy J Price, James Wilson, Helen E Moss, Paul M Matthews, and Lorraine K Tyler. Susceptibility-induced loss of signal: comparing pet and fmri on a semantic task. *Neuroimage*, 11(6): 589–600, 2000.

Robert M Dorazio, Bhramar Mukherjee, Li Zhang, Malay Ghosh, Howard L Jelks, and Frank Jordan. Modeling unobserved sources of heterogeneity in animal abundance using a dirichlet process prior. *Biometrics*, 64(2):635–644, 2008.

Hani Doss. Estimation of bayes factors for nonparametric bayes problems via radon-nikodym derivatives. Technical report, Technical Report, University of Florida, Department of Statistics, 2008.

Hani Doss. Hyperparameter and model selection for nonparametric bayes problems via radon-nikodym derivatives. *Statistica Sinica*, 22:1–26, 2012.

Richard O Duda and Peter E Hart. Pattern recognition and scene analysis, 1973.

Simon B Eickhoff, Angela R Laird, Christian Grefkes, Ling E Wang, Karl Zilles, and Peter T Fox. Coordinate-based activation likelihood estimation meta-analysis of neuroimaging data: A random-effects approach based on empirical estimates of spatial uncertainty. *Human brain mapping*, 30(9):2907–2926, 2009.

Simon B Eickhoff, Danilo Bzdok, Angela R Laird, Florian Kurth, and Peter T Fox. Activation likelihood estimation meta-analysis revisited. *Neuroimage*, 59(3):2349–2361, 2012.

Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, 90(430):577–588, 1995.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

AC Evans, DL Collins, and B Milner. An mri-based stereotactic atlas from 250 young normal subjects. In *Soc. neurosci. abstr*, volume 18, page 408, 1992.

Alan C Evans, D Louis Collins, SR Mills, ED Brown, RL Kelly, and Terry M Peters. 3d statistical neuroanatomical models from 305 mri volumes. In *Nuclear Science Symposium and Medical Imaging Conference, 1993., 1993 IEEE Conference Record.*, pages 1813–1817. IEEE, 1993.

Daniel Falush, Matthew Stephens, and Jonathan K Pritchard. Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, 164(4):1567–1587, 2003.

Anja Feldmann and Ward Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. In *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, volume 3, pages 1096–1104. IEEE, 1997.

Ching-Mei Feng, Shalini Narayana, Jack L Lancaster, Paul A Jerabek, Thomas L Arnow, Fang Zhu, Li Hai Tan, Peter T Fox, and Jia-Hong Gao. Cbf changes during brain activation: fmri vs. pet. *Neuroimage*, 22(1):443–446, 2004.

Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

Mario AT Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24 (3):381–396, 2002.

Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1): 176–190, 2008.

Peter T Fox, Jack L Lancaster, Lawrence M Parsons, Jin-Hu Xiong, and Frank Zamarripa. Functional volumes modeling: Theory and preliminary assessment. *Human Brain Mapping*, 5(4):306–311, 1997.

Olivier François, Sophie Ancelet, and Gilles Guillot. Bayesian clustering using hidden markov random fields in spatial population genetics. *Genetics*, 174(2):805–816, 2006.

Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

A Gelman, JB Carlin, HS Stern, and DB Rubin. Bayesian data analysis, 1995. *Chapman&Hall, London*.

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

Charles J Geyer. Practical markov chain monte carlo. *Statistical Science*, pages 473–483, 1992.

Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

Peter J Green and David I Hastie. Reversible jump mcmc. *Genetics*, 155(3):1391–1403, 2009.

Gilles Guillot, Arnaud Estoup, Frédéric Mortier, and Jean François Cosson. A spatial statistical model for landscape genetics. *Genetics*, 170(3):1261–1280, 2005.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Jian Kang, Timothy D Johnson, Thomas E Nichols, and Tor D Wager. Meta analysis of functional neuroimaging data via bayesian spatial point processes. *Journal of the American Statistical Association*, 106(493):124–134, 2011.

Robert E Kass and Larry Wasserman. A reference bayesian test for nested hypotheses and its relationship to the schwarz criterion. *Journal of the American Statistical Association*, 90(431):928–934, 1995.

Marc Kéry, J Andrew Royle, and Hans Schmid. Modeling avian abundance from replicated counts using binomial mixture models. *Ecological applications*, 15(4): 1450–1461, 2005.

Hedy Kober, Lisa Feldman Barrett, Josh Joseph, Eliza Bliss-Moreau, Kristen Lindquist, and Tor D Wager. Functional grouping and cortical–subcortical interactions in emotion: A meta-analysis of neuroimaging studies. *Neuroimage*, 42(2): 998–1031, 2008.

Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

Minjung Kyung, Jeff Gill, and George Casella. Estimation in dirichlet random effects models. *The Annals of Statistics*, 38(2):979–1009, 2010.

Angela R Laird, P Mickle Fox, Cathy J Price, David C Glahn, Angela M Uecker, Jack L Lancaster, Peter E Turkeltaub, Peter Kochunov, and Peter T Fox. Ale meta-analysis: Controlling the false discovery rate and performing statistical contrasts. *Human brain mapping*, 25(1):155–164, 2005.

Angela R Laird, Jennifer L Robinson, Kathryn M McMillan, Diana Tordesillas-Gutiérrez, Sarah T Moran, Sabina M Gonzales, Kimberly L Ray, Crystal Franklin, David C Glahn, Peter T Fox, et al. Comparison of the disparity between talairach and mni coordinates in functional neuroimaging data: validation of the lancaster transform. *Neuroimage*, 51(2):677–683, 2010.

Jack L Lancaster, Diana Tordesillas-Gutiérrez, Michael Martinez, Felipe Salinas, Alan Evans, Karl Zilles, John C Mazziotta, and Peter T Fox. Bias between mni and talairach coordinates analyzed using the icbm-152 brain template. *Human brain mapping*, 28(11):1194–1205, 2007.

Emily K Latch, Guha Dharmarajan, Jeffrey C Glaubitz, and Olin E Rhodes Jr. Relative performance of bayesian clustering software for inferring population substructure and individual assignment at low levels of population differentiation. *Conservation Genetics*, 7(2):295–302, 2006.

Brian G Leroux and Martin L Puterman. Maximum-penalized-likelihood estimation for independent and markov-dependent mixture models. *Biometrics*, pages 545–558, 1992.

Jun S Liu. Nonparametric hierarchical bayes via sequential imputations. *The Annals of Statistics*, 24(3):911–930, 1996.

Stuart Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

Nikos K Logothetis. What we can do and what we cannot do with fmri. *Nature*, 453 (7197):869–878, 2008.

Steven N MacEachern and Peter Müller. Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):223–238, 1998.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. California, USA, 1967.

Jean-Michel Marin, Kerrie Mengersen, and Christian P Robert. Bayesian modelling and inference on mixtures of distributions. *Handbook of statistics*, 25:459–507, 2005.

Jon D McAuliffe, David M Blei, and Michael I Jordan. Nonparametric empirical bayes for the dirichlet process mixture model. *Statistics and Computing*, 16(1): 5–14, 2006.

Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

Geoffrey J McLachlan and Kaye E Basford. Mixture models. inference and applications to clustering. *Statistics: Textbooks and Monographs, New York: Dekker*, 1, 1988.

GJ McLachlan and PN Jones. Fitting mixture models to grouped and truncated data via the em algorithm. *Biometrics*, pages 571–578, 1988.

GJ McLachlan and D Peel. *Finite mixture models*. John Wiley & Sons, 2000.

Christine E McLaren, Gary M Brittenham, and Victor Hasselblad. Analysis of the volume of red blood cells: application of the expectation-maximization algorithm to grouped data from the doubly-truncated lognormal distribution. *Biometrics*, pages 143–158, 1986.

Xiao-Li Meng and Donald B Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

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, 1953.

Glenn W Milligan and Martha C Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

Glenn W Milligan and Martha C Cooper. Methodology review: Clustering methods. *Applied psychological measurement*, 11(4):329–354, 1987.

Jesper Moller and Rasmus Plenge Waagepetersen. *Statistical inference and simulation for spatial point processes*. CRC Press, 2004.

Fionn Murtagh and Adrian E Raftery. Fitting straight lines to point patterns. *Pattern recognition*, 17(5):479–483, 1984.

Malay Naskar and Kalyan Das. Inference in dirichlet process mixed generalized linear models by using monte carlo em. *Australian & New Zealand Journal of Statistics*, 46(4):685–701, 2004.

Malay Naskar and Kalyan Das. Semiparametric analysis of two-level bivariate binary data. *Biometrics*, 62(4):1004–1013, 2006.

Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

Seiji Ogawa, TM Lee, AR Kay, and DW Tank. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences*, 87(24):9868–9872, 1990.

Jeffrey G Ojemann, Randy L Buckner, Erbil Akbudak, Abraham Z Snyder, John M Ollinger, Robert C McKinstry, Bruce R Rosen, Steve E Petersen, Marcus E Raichle, and Thomas E Conturo. Functional mri studies of word-stem completion: reliability across laboratories and comparison to blood flow imaging with pet. *Human Brain Mapping*, 6(4):203–215, 1998.

David Pollard. A central limit theorem for k-means clustering. *The Annals of Probability*, pages 919–926, 1982.

Chris Preston. Spatial birth and death processes. In *ADVANCES IN APPLIED PROBABILITY*, volume 7, pages 465–466. APPLIED PROBABILITY TRUST THE UNIVERSITY, SCHOOL MATHEMATICS STATISTICS, SHEFFIELD, ENGLAND S3 7RH, 1975.

Chris Preston. Spatial birth-and-death processes. *Bull. Inst. Interational Statistics*, 46:371–391, 1976.

Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

Jonathan K Pritchard, William Wen, and Daniel Falush. Documentation for structure software: version 2. 2003.

Adrian E Raftery. Bayesian model selection in social research. *Sociological methodology*, 25:111–164, 1995.

Adrian E Raftery. Bayes factors and bic. *Sociological Methods & Research*, 27(3): 411–417, 1999.

Carl Edward Rasmussen. The infinite gaussian mixture model. In *NIPS*, volume 12, pages 554–560, 1999.

Jorma Rissanen. *Stochastic complexity in statistical inquiry theory*. World Scientific Publishing Co., Inc., 1989.

J Andrew Royle. N-mixture models for estimating population size from spatially replicated counts. *Biometrics*, 60(1):108–115, 2004.

Gholamreza Salimi-Khorshidi, Stephen M Smith, John R Keltner, Tor D Wager, and Thomas E Nichols. Meta-analysis of neuroimaging data: a comparison of image-based and coordinate-based pooling of studies. *Neuroimage*, 45(3):810–823, 2009.

Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

AJ Scott and Michael J Symons. Clustering methods based on likelihood ratio criteria. *Biometrics*, pages 387–397, 1971.

John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

Stephen M Smith. Overview of fmri analysis. *British journal of radiology*, 77(suppl 2):S167–S175, 2004.

David J Spiegelhalter, Nicola G Best, Bradley P Carlin, and Angelika Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.

Matthew Stephens. Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. *Annals of Statistics*, pages 40–74, 2000.

J Talairach and Tournoux. *Co-planar Stereotaxic Atlas of the Human Brain: 3-Dimensional Proportional System - an Approach to Cerebral Imaging*. Thieme Medical Publishers, 1988.

Bertrand Thirion, Philippe Pinel, Sébastien Mériaux, Alexis Roche, Stanislas Dehaene, and Jean-Baptiste Poline. Analysis of a large fmri cohort: Statistical and methodological issues for group analyses. *Neuroimage*, 35(1):105–120, 2007.

D Michael Titterington, Adrian FM Smith, Udi E Makov, et al. *Statistical analysis of finite mixture distributions*, volume 7. Wiley New York, 1985.

Peter E Turkeltaub, Guinevere F Eden, Karen M Jones, and Thomas A Zeffiro. Meta-analysis of the functional neuroanatomy of single-word reading: method and validation. *Neuroimage*, 16(3):765–780, 2002.

Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *Neural Networks, IEEE Transactions on*, 11(3):586–600, 2000.

Tor D Wager, K Luan Phan, Israel Liberzon, and Stephan F Taylor. Valence, gender, and lateralization of functional brain anatomy in emotion: a meta-analysis of findings from neuroimaging. *Neuroimage*, 19(3):513–531, 2003.

Tor D Wager, John Jonides, and Susan Reading. Neuroimaging studies of shifting attention: a meta-analysis. *Neuroimage*, 22(4):1679–1693, 2004.

Tor D Wager, Martin Lindquist, and Lauren Kaplan. Meta-analysis of functional neuroimaging data: current and future directions. *Social cognitive and affective neuroscience*, 2(2):150–158, 2007.

Stephen Walker and Paul Damien. Sampling methods for bayesian nonparametric inference involving stochastic processes. In *Practical Nonparametric and Semiparametric Bayesian Statistics*, pages 243–254. Springer, 1998.

Samuel K Wasser, Andrew M Shedlock, Kenine Comstock, Elaine A Ostrander, Benezeth Mutayoba, and Matthew Stephens. Assigning african elephant dna to geographic region of origin: applications to the ivory trade. *Proceedings of the National Academy of Sciences of the United States of America*, 101(41):14847–14852, 2004.

M West. M uller, p., and escobar, md (1994)\ hierarchical priors and mixture models, with application in regression and density estimation. *Aspects of Uncertainty*, pages 363–386.

Choong-Wan Woo, Anjali Krishnan, and Tor D Wager. Cluster-extent based thresholding in fmri analyses: Pitfalls and recommendations. *NeuroImage*, 91:412–419, 2014.

CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

Mohamed Zaït and Hammou Messatfa. A comparative study of clustering methods. *Future Generation Computer Systems*, 13(2):149–159, 1997.

Lianjun Zhang, Jeffrey H Gove, Chuangmin Liu, and William B Leak. A finite mixture of two weibull distributions for modeling the diameter distributions of rotated-sigmoid, uneven-aged stands. *Canadian journal of forest research*, 31(9): 1654–1659, 2001.

Mingyuan Zhou and Lawrence Carin. Negative binomial process count and mixture modeling. 2013.

# Appendix A

# R code for spatial Poisson point process coding with real data

```
%\begin{lstlisting}
library(MASS)
library(pscl)
library(cluster)
library(msm)
library(tmvtnorm)
library(plyr)
data<-read.csv('C:\\Users\\Meredith\\Documents\\USC\\Dissertation\\Data\\
meta061106.JianKangcopy.csv',header=T)
colnames(data)
#> colnames(data)
# [1] "Study"             "headstable"        "Scan"
# [4] "ROI"               "CoordSys"          "Subjects"
# [7] "Women"             "Gender"            "Mode"
#[10] "FixedRandom"       "Stimuli"           "Method_"
#[13] "Emotion"           "Valence"           "EERMean"
#[16] "EERSD"             "EERRS"             "Subtraction"
#[19] "Contrast"          "Target"            "Ref"
#[22] "Other"             "Intrascan_cog"     "x"
#[25] "y"                 "z"                 "zscore"
#[28] "tscore"            "p_threshold"       "pcorrection"
#[31] "Region"            "Notes"             "SubjectiveWeights"
#[34] "Affect_focus"      "Cog_load"          "object_eval"
#[37] "feel_arousal"      "phys_arous"
###Extracting Columns needed###
data1<-data[,c("x","y","z","Study","Contrast")]
colnames(data1)<-c("X","Y","Z","pub","sub_study")
data1[,1:3]<-data1[,1:3]/10
n<-nrow(data1)
nn<-length(unique(data1$sub_study))
cluster <- as.numeric(c(rep(1,n)))
a_ij <- c(rep(6.5,n))
phi_X <- c(rep(mean(data1[,1], na.rm=FALSE),n))
phi_Y <- c(rep(mean(data1[,2], na.rm=FALSE),n))
phi_Z <- c(rep(mean(data1[,3], na.rm=FALSE),n))
X_0 <- median(data1[,1]-phi_X, na.rm=FALSE)
Y_0 <- median(data1[,2]-phi_Y, na.rm=FALSE)
Z_0 <- median(data1[,3]-phi_Z, na.rm=FALSE)
c_0 <- as.numeric(t(c(X_0,Y_0,Z_0)))
phi_Px <- c(rep(0,n))
phi_Py <- c(rep(0,n))
phi_Pz <- c(rep(0,n))
study <- as.numeric(c(rep(1,n)))
study.indicator<-NULL
```

```
for(ss in unique(data1$sub_study)){
study.indicator<-c(study.indicator,match(ss,data1$sub_study))}
sub.study.indicator<-c(rep(0,n))
sub.study.indicator[study.indicator]<-1
data <- data.frame(data1,study,cluster,phi_X,phi_Y,phi_Z,a_ij,phi_Px,
phi_Py,phi_Pz,sub.study.indicator)
colnames(data)<-c("X","Y","Z","pub","sub_study","study","cluster",
"phi_X","phi_Y","phi_Z","a_ij","phi_Px","phi_Py","phi_Pz","indicator")
##Functions##
#Log-Likehihood - All Subjects#
likelihood_inter <- function(a){
        vec<-cbind((data$X-data$phi_Px -data$phi_X),
        (data$Y-data$phi_Py -data$phi_Y),(data$Z-data$phi_Pz -data$phi_Z))
        distance5 <- sqrt(diag(vec%*%t(vec)))
        kernel5 <- data$a_ij*exp(-(distance5*distance5)/rho)
        all_sum_ind5 <- -(sum(exp(kernel5))/n) + sum(kernel5)
return(all_sum_ind5)
}
#Log-Likelihood - Individuals#
likelihood_I <- function(a,b,c,d){
        vec<-c((data$X[i]-data$phi_Px[i]-a),(data$Y[i]-data$phi_Py[i]-b),
        (data$Z[i]-data$phi_Pz[i]-c))
        distance1 <- sqrt(t(vec)%*% vec)
        kernel1<-d*exp(-(distance1*distance1)/rho)
        individual1 <- -exp(kernel1)/n + kernel1
return(individual1)
}
#Log-Likehihood - cluster effect - updated theta in cluster e#
likelihood_A <- function(a,b,c,d,e){
        ind<-which(data$cluster==e)
        vec<-cbind((data$X[ind]-data$phi_Px[ind]-a),
        (data$Y[ind]-data$phi_Py[ind]-b),(data$Z[ind]-data$phi_Pz[ind]-c))
        distance0 <- sqrt(diag(vec %*% t(vec)))
        kernel0 <- d*exp(-(distance0*distance0)/rho)
        individ.cluster<- -sum(exp(kernel0))/n  + sum(kernel0)
return(individ.cluster)
}
#Log-Likelihood - Study Effect#
likelihood_P.new <- function(a,b,c){
        ind<-which(data$sub_study==j)
        vec<-cbind((data$X[ind]-a-data$phi_X[ind]),
        (data$Y[ind]-b-data$phi_Y[ind]),
        (data$Z[ind]-c-data$phi_Z[ind]))
        distance1 <- sqrt(diag(vec %*% t(vec)))
        study_dist1 <- data$a_ij[ind]*exp(-(distance1*distance1)/rho)
        study.individ<- -sum(exp(study_dist1))/n + sum(study_dist1)
return(study.individ)
}
#Log-Likelihood - Study cluster - updated p in cluster e#
likelihood_P.all <- function(a,b,c,d){
        ind<-which(data$study==d)
        vec<-cbind((data$X[ind]-a-data$phi_X[ind]),
        (data$Y[ind]-b-data$phi_Y[ind]),
        (data$Z[ind]-c-data$phi_Z[ind]))
        distance1 <- sqrt(diag(vec %*% t(vec)))
        study_dist1 <- data$a_ij[ind]*exp(-(distance1*distance1)/rho)
        study.cluster<- -sum(exp(study_dist1))/n + sum(study_dist1)
return(study.cluster)
}
##Parameters for Priors##
h1 <- 0.5
```

```
m1 <- 0.5
I <- diag(3)
alpha_0 <- 0
sigma_a <- 25
m <- 7
mm <- 3
a.hyper<-1.1
b.hyper<-0.1
'%ni%' <- Negate('%in%')
mu_mu<-c(0,0,0)
sigma_mu<-0.04
q<-2
l<-1
#####Algorithm#####
burn.in<-4000
working<-3000 + burn.in
final<-1000
nloops <-working+final
DIC<-loglike<-NULL
alpha_1<-0.5
alpha_2<-1.5
rho <- 1
limit<-0.15*abs(apply(data[,c("X","Y","Z")],2,max)-
apply(data[,c("X","Y","Z")],2,min))
mu_conditional<-matrix(c(0,0,0),nrow=nloops+1,ncol=3)
sigma_conditional<-rep(1,nloops+1)
sigma2_conditional<-rep(1,nloops+1)
cluster_matrix <- matrix(0,ncol=n, nrow=n)
median.obs<-c(median(data$X),median(data$Y),median(data$Z))
lower<-c(min(data$X),min(data$Y),min(data$Z))
upper<-c(max(data$X),max(data$Y),max(data$Z))
ls.cluster<-1000000000000000
Dev<-NULL
off = TRUE
combo<-data.frame(matrix(NA,nrow=m,ncol=5))
colnames(combo)<-c("phi_X","phi_Y","phi_Z","a_ij","Freq")
rownames(combo) <- c("Aux1","Aux2","Aux3","Aux4","Aux5","Aux6","Aux7")
combo2<-matrix(NA,nrow=m,ncol=3)
colnames(combo2)<-c("phi_X","phi_Y","phi_Z")
combo3<-matrix(NA,nrow=m,ncol=4)
colnames(combo3)<-c("phi_Px","phi_Py","phi_Pz","Freq")
rownames(combo) <- rownames(combo2)<-rownames(combo3)<-c("Aux1","Aux2",
                                    "Aux3","Aux4","Aux5","Aux6","Aux7")
########Algorithm#######
for (x in 1:nloops){
###Individual Clustering###
data.individual<-as.matrix(cbind(data$X-data$phi_Px,data$Y-data$phi_Py,
data$Z-data$phi_Pz))
cluster.current<-data.frame(cbind(do.call("rbind",
as.list(by(data[,c("phi_X","phi_Y","phi_Z","a_ij")],
data$cluster,tail,n=1))),table(data$cluster)))[,-5]
for (i in 1:n){
#Updating cluster probabilities
if(any(cluster.current$Freq==0)==TRUE){
cluster.current=cluster.current[-which(cluster.current$Freq==0),]
}
remove<-which(rownames(cluster.current) == data$cluster[i])
cluster.current[remove,5]<-cluster.current[remove,5]-1
phi_A <- matrix(rtmvnorm(1,median.obs, sigma2_conditional[x]*I,
lower=lower,upper=upper), ncol=3,nrow=1)
    combo[c(1,2,4),1]<-data$phi_X[i]
    combo[c(3,5,6,7),1]<-phi_A[1,1]
        combo[c(1,3,5),2]<-data$phi_Y[i]
```

```
        combo[c(2,4,6,7),2]<-phi_A[1,2]
        combo[c(2,3,6),3]<-data$phi_Z[i]
        combo[c(1,4,5,7),3]<-phi_A[1,3]
        combo[,4]<- data$a_ij[i]
        combo[,5]<-(alpha_2/m)
cluster.aux<-data.frame(rbind(cluster.current,combo))
max.cluster<-eval(1+max(data$cluster))
         matrix.temp<-as.matrix(matrix(rep(data.individual[i,],
      nrow(cluster.aux)),nrow=nrow(cluster.aux),byrow=T)
      -cluster.aux[,-c(4:5)])
        distance1<-sqrt(diag(matrix.temp%*%t(matrix.temp)))
        kernel1<-cluster.aux[,4]*exp(-{distance1*distance1}/rho)
        prob<-log(cluster.aux[,5])+{-exp(kernel1)/n + kernel1}
        max<-max(prob)
        new_cluster <- rmultinom(1, 1, exp(prob-max))
        inds = which(new_cluster == max(new_cluster), arr.ind=TRUE)
        cluster_name = rownames(new_cluster)[inds[,1]]
        rnames = rownames(new_cluster)[inds[,1]]
    ind<-which(rownames(cluster.aux) == rnames)
                data$phi_X[i] <- cluster.aux[ind,1]
                data$phi_Y[i] <- cluster.aux[ind,2]
                data$phi_Z[i] <- cluster.aux[ind,3]
                data$a_ij[i] <- cluster.aux[ind,4]
if(ind %ni% (nrow(cluster.aux)-6):nrow(cluster.aux)){
    data$cluster[i] <- as.numeric(rnames);
    cluster.current[ind,5]<-cluster.current[ind,5]+1
}else{
    rownames(cluster.aux)[ind]<-data$cluster[i]<-max.cluster;
    cluster.aux[ind,5]<-1;
    cluster.current<-rbind(cluster.current,cluster.aux[ind,]);
    rownames(cluster.current)<-c(rownames(cluster.current)[
            -nrow(cluster.current)],data$cluster[i])
}
} #end of individual cluster loop
        ###Updating theta for each cluster###
for(d in unique(data$cluster)){
   cluster.X <- data$phi_X[data$cluster==d]
   cluster.Y <- data$phi_Y[data$cluster==d]
   cluster.Z <- data$phi_Z[data$cluster==d]
   cluster.a <- data$a_ij[data$cluster==d]
   theta_MH<- as.numeric(cluster.current[{rownames(cluster.current)==d},
    1:3])
   cand.sd2 <- 0.005
   can.theta <- rtmvnorm(1,theta_MH, cand.sd2*I,lower=lower,
    upper=upper)
        combo2[c(1,2,4),1]<-theta_MH[1]
        combo2[c(3,5,6,7),1]<-can.theta[1]
        combo2[c(1,3,5),2]<-theta_MH[2]
        combo2[c(2,4,6,7),2]<-can.theta[2]
        combo2[c(2,3,6),3]<-theta_MH[3]
        combo2[c(1,4,5,7),3]<-can.theta[3]
        pcan.max<-apply(combo2,1,function(y) log(dtmvnorm(c(y[1],y[2],y[3]),
        c_0,sigma2_conditional[x]*I,lower=lower,upper=upper))
        +likelihood_A(y[1],y[2],y[3],cluster.a,d))
        pcan.ind<-which(pcan.max == max(pcan.max))
        if(length(pcan.ind) > 1 && pcan.max[pcan.ind][1] == 'Inf'){
        pcan.like<-sample(pcan.ind,1)
        pcan.ind<-pcan.like
    }
pcur2 <-likelihood_A(theta_MH[1],theta_MH[2],theta_MH[3],cluster.a,d)+
    log(dtmvnorm(c(theta_MH[1],theta_MH[2],theta_MH[3]), c_0,
```

```
        sigma2_conditional[x]*I,lower=lower,upper=upper))
        jcan2<-log(dtmvnorm(combo2[pcan.ind,],theta_MH, cand.sd2*I,
        lower=lower,upper=upper))
        jcur2<-log(dtmvnorm(theta_MH,combo2[pcan.ind,], cand.sd2*I,
        lower=lower,upper=upper))
        min<-pcan.max[pcan.ind]-pcur2+jcur2-jcan2
        if(log(runif(1))<=min){data$phi_X[data$cluster==d]<-combo2[pcan.ind,
    1]
           data$phi_Y[data$cluster==d]<-combo2[pcan.ind,2]
           data$phi_Z[data$cluster==d]<-combo2[pcan.ind,3]
        }else{data$phi_X[data$cluster==d]<-theta_MH[1]
        data$phi_Y[data$cluster==d]<-theta_MH[2]
             data$phi_Z[data$cluster==d]<-theta_MH[3]
        }
}#end up updating theta
###Updating A_ij after each loop###
MH.iter2<-100
c <- 0
for(d in unique(data$cluster)){
        c <- c+1
        a_ij_MH2<- 0
        cluster.X <- data$phi_X[data$cluster==d]
        cluster.Y <- data$phi_Y[data$cluster==d]
        cluster.Z <- data$phi_Z[data$cluster==d]
   for (p in 1:MH.iter2){
        cand.sd2 <- 2
        can2 <- rnorm(1,a_ij_MH2[p], cand.sd2)
   pcan2<-likelihood_A(cluster.X[1],cluster.Y[1],cluster.Z[1],can2,d)
         +log(dnorm(can2, 0, sigma_a))
   pcur2<-likelihood_A(cluster.X[1],cluster.Y[1],cluster.Z[1],a_ij_MH2[p],d)
         +log(dnorm(a_ij_MH2[p], 0, sigma_a))
   compare<-(pcan2-pcur2)
             if (log(runif(1)) <= compare){
           a_ij_MH2 <- c(a_ij_MH2,can2)
             }else{
           a_ij_MH2 <- c(a_ij_MH2,a_ij_MH2[p])
        }
        }
data$a_ij[data$cluster == d] <- mean(a_ij_MH2[MH.iter2-50:MH.iter2])
}#end updating a_ij
###Sigma_2 full conditional posterior###
vec<- cbind((data$phi_X-median.obs[1]),(data$phi_Y-median.obs[2]),
(data$phi_Z-median.obs[3]))
distance3 <-sum(diag(vec%*%t(vec)))+1
sigma2_conditional[x+1]<-rigamma(1,(n+1)/2 , 0.5*distance3)
###Study Clustering###
cluster.current2<-data.frame(cbind(do.call("rbind",
as.list(by(data[,c("phi_Px","phi_Py","phi_Pz")],data$study,tail,n=1))),
table(data$study[data$indicator==1])))[,-4]
for (j in unique(data$sub_study)){
if(any(cluster.current2$Freq==0)==TRUE){
    cluster.current2=cluster.current2[-which(cluster.current2$Freq==0),]
}
remove<-which(rownames(cluster.current2)==unique(data$study[
data$sub_study==j]))
cluster.current2[remove,4]<-cluster.current2[remove,4]-1
phi_A2 <- rtmvnorm(1,c(mu_conditional[x,]), sigma_conditional[x]*diag(3),
    lower=-limit,upper=limit)
    combo3[c(1,2,4),1]<-data$phi_Px[data$sub_study==j&data$indicator==1]
    combo3[c(3,5,6,7),1]<-phi_A2[1,1]
```

```
    combo3[c(1,3,5),2]<-data$phi_Py[data$sub_study==j&data$indicator==1]
    combo3[c(2,4,6,7),2]<-phi_A2[1,2]
    combo3[c(2,3,6),3]<-data$phi_Pz[data$sub_study==j&data$indicator==1]
    combo3[c(1,4,5,7),3]<-phi_A2[1,3]
        combo3[,4]<-(alpha_1/mm)
cluster.aux<-data.frame(rbind(cluster.current2,combo3))
max.study<-eval(1+max(data$study))
        prob2<-apply(cluster.aux,1,function(x)
            log(x[4])+likelihood_P.new(x[1],x[2],x[3]))
                max<-max(prob2)
                new_cluster <- rmultinom(1, 1, exp(prob2-max))
                rownames(new_cluster) = names(prob2)
                inds = which(new_cluster == max(new_cluster), arr.ind=TRUE)
                rnames = rownames(new_cluster)[inds[,1]]
                ind<-which(rownames(cluster.aux) == rnames)
                data$phi_Px[data$sub_study==j] <- cluster.aux[ind,1]
                data$phi_Py[data$sub_study==j] <- cluster.aux[ind,2]
                data$phi_Pz[data$sub_study==j] <- cluster.aux[ind,3]
        if(ind %ni% (nrow(cluster.aux)-6):nrow(cluster.aux)){
        data$study[data$sub_study==j] <- as.numeric(rnames);
        cluster.current2[ind,4]<-cluster.current2[ind,4]+1
    }else{
       rownames(cluster.aux)[ind]<-data$study[data$sub_study==j]<-max.study;
       cluster.aux[ind,4]<-1;
       cluster.current2<-rbind(cluster.current2,cluster.aux[ind,])
    }
} #end of loop for study
###Updating p_i for each cluster###
for(d in unique(data$study)){
        cluster.p <- data[data$study==d,c("phi_Px","phi_Py","phi_Pz")][1,]
        theta_MH2<- as.numeric(cluster.p)
        cand.sd2 <- 0.005
        can.theta2 <- rtmvnorm(1,theta_MH2, cand.sd2*I, lower=-limit
    ,upper=limit)
        combo2[c(1,2,4),1]<-theta_MH2[1]
        combo2[c(3,5,6,7),1]<-can.theta2[1]
        combo2[c(1,3,5),2]<-theta_MH2[2]
        combo2[c(2,4,6,7),2]<-can.theta2[2]
        combo2[c(2,3,6),3]<-theta_MH2[3]
        combo2[c(1,4,5,7),3]<-can.theta2[3]
        pcan.max<-apply(combo,1,function(y) log(dtmvnorm(c(y[1],y[2],y[3]),
        mu_conditional[x,],sigma_conditional[x]*I,lower=-limit,upper=limit))
        +likelihood_P.all(y[1],y[2],y[3],d))
        pcan.ind<-which(pcan.max == max(pcan.max))
        if(length(pcan.ind) > 1 && pcan.max[pcan.ind][1] == 'Inf'||
        pcan.max[pcan.ind][1] == '-Inf'){pcan.like<-sample(pcan.ind,1)
                pcan.ind<-pcan.like}
    pcur2 <-likelihood_P.all(theta_MH2[1],theta_MH2[2],theta_MH2[3],d)
        +log(dtmvnorm(c(theta_MH2[1],theta_MH2[2],theta_MH2[3]),
        mu_conditional[x,],
        sigma_conditional[x]*I,lower=-limit,upper=limit))
        jcan2<-log(dtmvnorm(combo2[pcan.ind,],theta_MH2, cand.sd2*I,
        lower=-limit,upper=limit))
        jcur2<-log(dtmvnorm(theta_MH2,combo2[pcan.ind,], cand.sd2*I,
        lower=-limit,upper=limit))
        min<-pcan.max[pcan.ind]-pcur2+jcur2-jcan2
   if(log(runif(1))<=min){data$phi_Px[data$study==d]<-combo2[pcan.ind,1]
           data$phi_Py[data$study==d]<-combo2[pcan.ind,2]
           data$phi_Pz[data$study==d]<-combo2[pcan.ind,3]
        }else{
```

```
            data$phi_Px[data$study==d]<-theta_MH2[1]
            data$phi_Py[data$study==d]<-theta_MH2[2]
        data$phi_Pz[data$study==d]<-theta_MH2[3]
    }
}#end updating p_i
###Mu Full Conditional Posterior - Study effect###
study.temp<-data[data$indicator == 1,c("phi_Px","phi_Py","phi_Pz")]
mean.study <- colSums(study.temp)/nn
mu_hyper<-rtmvnorm(1,1/(nn*(1/sigma_conditional[x])
    +(1/sigma_mu))*(nn*(1/sigma_conditional[x])*mean.study
    +(1/sigma_mu)*mu_mu),solve((nn*(1/sigma_conditional[x])
    +(1/sigma_mu))*I),lower=-limit,upper=limit)
mu_conditional[x+1,] <- mu_hyper
###Sigma Full Conditional Posterior - Study effect###
vec<-as.matrix(study.temp - mu_conditional[x+1,])
distance4 <- sum(diag(vec %*% t(vec)))
sigma_conditional[x+1] <- rigamma(1,((nn/2) + q) , 0.5*distance4 + l)
###least squares clustering###
#Updating clustering indicator Matrix#
if(x >= burn.in & x < working){
        for(ind in unique(data$cluster)){
                matrix_indicator<-which(data$cluster == ind)
                cluster_matrix[matrix_indicator,matrix_indicator] <-
            cluster_matrix[matrix_indicator,matrix_indicator] + 1
    }
}
#Calculating the average probability matrix at the#
#last Wth or working iteration#
if(x == working){
        cluster_average<-cluster_matrix/{x-burn.in}
        best.distance=10000000
}
#F or final iterations#
if(x >= working){
        cluster_matrix.temp<-matrix(0,nrow=n,ncol=n)
        for(ind in unique(data$cluster)){
        matrix_indicator<-which(data$cluster == ind)
        cluster_matrix.temp[matrix_indicator,matrix_indicator]<-
        cluster_matrix.temp[matrix_indicator,matrix_indicator]+1
}
ls.distance<-sum({cluster_matrix.temp-cluster_average}
        *{cluster_matrix.temp-cluster_average})
if(ls.distance <= best.distance){
        best.distance<-ls.distance
        cluster.data<-data
        updated.run<-x
}
}
#DIC Calculations#
if(x > burn.in){Dev<- c(Dev,-2*likelihood_inter(0))}
cat(alpha_1," ",alpha_2," ",x," ","\n")
} #end of program
data<-cluster.data
#DIC<-2*(median(Dev))-Dev[(length(Dev))]
#DIC2<-2*(mean(Dev))-Dev[(length(Dev))]
DIC<-2*(median(Dev))-Dev[(updated.run-burn.in)]
DIC2<-2*(mean(Dev))-Dev[(updated.run-burn.in)]
loglike<-likelihood_inter(0)
```

# Appendix B

## R code for multivariate normal mixture model

```
%\begin{lstlisting}
library(MASS)
library(pscl)
library(cluster)
library(stats)
library(mvtnorm)
library(MCMCpack)
library(msm)
library(tmvtnorm)
data<-read.csv('C:\\Users\\Meredith\\Documents\\USC\\Dissertation\\Data\\
meta061106.JianKangcopy.csv',header=T)
n<-500
nn<-50
set.seed(1)
#Cluster 1 - mu1.c = 1#
c1s1<-matrix((mvrnorm(75, c(1,1,1), diag(3)* .002)),75,3)
c1s2<-matrix((mvrnorm(75, c(1,1,1), diag(3)* .002)),75,3)
#Cluster 2 - mu2.c = 20#
c2s1<-matrix((mvrnorm(75, c(2,2,2), diag(3)* .002)),75,3)
c2s2<-matrix((mvrnorm(75, c(2,2,2), diag(3)* .002)),75,3)
#Cluster 3 - mu3.c = 50#
c3s1<-matrix((mvrnorm(100, c(4,4,4), diag(3)* .002)),100,3)
c3s2<-matrix((mvrnorm(100, c(4,4,4), diag(3)* .002)),100,3)
a<-rep(1:25, each=10)
b<- rep(26:50, each = 10)
s1 <- sample(a, 250, replace = FALSE, prob = NULL)
s2 <- sample(b, 250, replace = FALSE, prob = NULL)
study_1 <- rbind(c1s1,c2s1,c3s1)+0.4
study_2 <- rbind(c1s2,c2s2,c3s2)+0.1
#Combining to creat complete simulated dataset#
data1<-rbind(study_1, study_2) #removed rescaling data down by 1000
means<-colMeans(data1)
phi_Px <- data1[,1]*.05
phi_Py <- data1[,2]*.05
phi_Pz <- data1[,3]*.05
phi_X <- data1[,1]-phi_Px
phi_Y <- data1[,2]-phi_Py
phi_Z <- data1[,3]-phi_Pz
study.ori <- as.numeric(c(rep(1,250),rep(2,250)))
study <-  c(s1,s2)
#study <-  as.numeric(c(rep(1,n)))
sub_study <- c(s1,s2)
study.indicator<-NULL
for(ss in 1:50){
phi_Px[sub_study==ss] <- mean(phi_Px[sub_study==ss])
phi_Py[sub_study==ss] <- mean(phi_Py[sub_study==ss])
phi_Pz[sub_study==ss] <- mean(phi_Pz[sub_study==ss])
study.indicator<-c(study.indicator,match(ss,sub_study))}
```

```
sub.study.indicator<-c(rep(0,n))
sub.study.indicator[study.indicator]<-1
original <- c(rep(1,75),rep(2,75),rep(3,100),rep(1,75),rep(2,75),rep(3,100))
cluster <- as.numeric(c(rep(1,n)))
mu_X <- rep(0,n)
mu_Y <- rep(0,n)
mu_Z <- rep(0,n)
X_0 <- median(data1[,1], na.rm=FALSE)
Y_0 <- median(data1[,2], na.rm=FALSE)
Z_0 <- median(data1[,3], na.rm=FALSE)
c_0 <- t(c(X_0,Y_0,Z_0))
sig_X <- rep(1,n)
sig_Y <- rep(1,n)
sig_Z <- rep(1,n)
cluster_e<-1:500
pi <-c(rep(1,n))
obs<-1:n
data.full <- data.frame(data1,study.ori,study, sub_study,original,
cluster,phi_Px, phi_Py, phi_Pz, sig_X,sig_Y,sig_Z,mu_X,mu_Y,mu_Z,
sub.study.indicator, pi,obs,phi_X,phi_Y,phi_Z,cluster_e)
colnames(data.full)<-c("X","Y","Z","study_ori","study","sub_study",
"original","cluster","phi_Px","phi_Py","phi_Pz","sig_X", "sig_Y","sig_Z",
"mu_X","mu_Y","mu_Z","indicator","pi","obs","phi_X","phi_Y","phi_Z",
"cluster_e")
data<-data.full
##############Functions Needed##############
###Total Model###
###Compartment Death Probabilities###
comp.prob <- function(b){
death.comp<-NULL
param.temp<-data.frame(cbind(do.call("rbind", as.list(by(
data[,c("cluster","pi","mu_X","mu_Y","mu_Z","sig_X","sig_Y","sig_Z",
"phi_X","phi_Y","phi_Z")],
data$cluster,tail,n=1))),table(data$cluster)))
clus<-length(unique(data$cluster))
probs<-matrix(0,nrow=nrow(data),ncol=clus)
for(k in 1:clus){
   probs[,k]<-apply(as.matrix(data[,c("phi_X","phi_Y","phi_Z","cluster",
"cluster_e")]),1,function (x)
        (alpha1/(alpha1+as.numeric(param.temp[k,13])))*
        dmvnorm(x[1:3],as.numeric(param.temp[k,3:5]),
(as.numeric(param.temp[k,6:8])*diag(3)))
        +(as.numeric(length(data$cluster_e[data$cluster==param.temp[k,1]&
data$cluster_e == x[5]]))/(alpha1+as.numeric(param.temp[k,13]))))
}
        for(miss in 1:clus){
                full.temp<-probs%*%param.temp[,2]
                partial.pi<-(param.temp[,2]/(1-param.temp[miss,2]))
                partial.pi[miss]<-0
                partial.temp<-probs%*%partial.pi
                temp<-prod(partial.temp/full.temp)
                death.comp<-c(death.comp,(b/lambda)*temp)
        }
        return(death.comp)
}
###Birth/Death Decision###
bd.decision <- function(b,d){
if(any(d=='Inf')==TRUE){prob.dec<-c(0,1)
}else{prob.dec<-c(b/(b+sum(d)), sum(d)/(b+sum(d)))}
new_cluster<-rmultinom(1, 1, prob.dec)
rownames(new_cluster) = c('birth','death')
```

```
inds = which(new_cluster == max(new_cluster), arr.ind=TRUE)
rnames = rownames(new_cluster)[inds[,1]]
return(rnames)
}
###Compartment Death Decision###
comp.decision <- function(d){
if(any(d=='Inf')==TRUE){prob.death<-rep(0,length(d));
prob.death[which(d=='Inf')]<-1
}else{prob.death<-c(d/sum(d))}
new_cluster<-rmultinom(1, 1, prob.death)
rownames(new_cluster) = c(sort(unique(data$cluster)))
inds = which(new_cluster == max(new_cluster), arr.ind=TRUE)
rnames = rownames(new_cluster)[inds[,1]]
return(rnames)
}
#phi likelihood update#
likelihood_phi<-function(a,b,c,d){
temp.data<-data[which(data$cluster_e == d),]
ni<-length(which(data$cluster == temp.data[1,8]))
like<-apply(temp.data,1,function (x) dmvnorm(x[1:3],x[9:11]
+c(a,b,c),x[12:14]*I))
prior<-(alpha1/(alpha1+ni))*dmvnorm(c(a,b,c),
as.numeric(temp.data[15:17][1,]),as.numeric(temp.data[12:14][1,])*I)
+(nrow(temp.data)/(alpha1+ni))
output=list(like=like,prior=prior)
output
}
#Study likelihood#
study.like<-function(a,b,c,d){
temp.study<-data[data$sub_study==a,c("X","Y","Z","phi_X","phi_Y","phi_Z")]
study.out<-apply(temp.study,1,function(x) dmvnorm(c(x[1:3]),
c(x[4:6]+c(b,c,d)),sig_S[ii]*diag(3)))
return(prod(study.out))
}
#Least Squares Clustering#
l.s<-function(a,b){
sub.matrix<- ((a-b)*(a-b))
row.matrix<-rowSums(sub.matrix)
sum(row.matrix)
}
########Inital Rates/Hyperpriors########
n<-nrow(data)  #number of foci
nn<-length(unique(data$sub_study))  #number of sub-studies
kappa<-matrix(c(1/((max(data$X)-min(data$X))^2),0, 0, 0,
1/((max(data$Y)-min(data$Y))^2),0,0,0,1/((max(data$Z)-min(data$Z))^2)),
byrow=T,ncol=3)
xi<-c(median(data$X),median(data$Y),median(data$Z))
lambda.b<-0.0002 #some small constant(constant)
lambda<-lambda.b
m<-7        #auxiliary parameters for Neal's algorithm###
mm<-3
n<-nrow(data)  #number of individual
nn<-length(unique(data$sub_study))  #number of sub-studies
I=diag(3)
alpha<-0.5        #for study effect
alpha1<-2        #for individual cluster
'%ni%' <- Negate('%in%')
aa<-0.5        ###hyperpriors for sigma for S, observed values###
bb<-0.5     ####################################################
sig_S<- 5 #initial value for sigma for S, all observed values
#########Individual Base Distribution Parameters############
g<-20        ####hyperpriors for sigma for base distributions###
h<-0.5  ####################################################
```

```
#############Study Effect Parameters#################
mu_mu<-c(0,0,0)        ###hyperprior parameters for mu for study effect###
#sigma_mu<-0.04        ######################################################
sigma_mu<-0.1        ######################################################
q<-3          ###hyperprior parameters for sigma for study effect###
l<-0.5  ######################################################
mu_conditional<-matrix(c(0,0,0),nrow=1,ncol=3)
sigma_conditional<-0.1
limit<-0.15*abs(apply(data[,c("X","Y","Z")],2,max)-
apply(data[,c("X","Y","Z")],2,min))
burn.in<-300
working<-100 + burn.in
final<-100
#burn.in<-800
#working<-400 + burn.in
#final<-100
nloops<-working+final
Dev<-log.like<-rep(NA,nloops+1)
DIC<-full.like<-NULL
cluster_matrix <- matrix(0,ncol=n, nrow=n)
ls.cluster<-1000000000000000
off=FALSE
time=1
combo<-data.frame(matrix(NA,nrow=m,ncol=4))
colnames(combo)<-c("phi_X","phi_Y","phi_Z","Freq")
rownames(combo) <- c("Aux1","Aux2","Aux3","Aux4","Aux5","Aux6","Aux7")
combo3<-matrix(NA,nrow=m,ncol=4)
colnames(combo3)<-c("phi_Px","phi_Py","phi_Pz","Freq")
rownames(combo) <- c("Aux1","Aux2","Aux3","Aux4","Aux5","Aux6","Aux7")
rownames(combo3)<-c("Aux1","Aux2","Aux3","Aux4","Aux5","Aux6","Aux7")
for (ii in 1:nloops){
        if(length(unique(data$cluster))== 1){decision = 'birth'
        d.rate=0}else{
        d.rate<-comp.prob(lambda.b)
        decision<-bd.decision(lambda.b,d.rate)
        }
        jump<-rexp(1,rate=(lambda.b+sum(d.rate)))
        if(jump < 1){jump <- 1}else{jump<-ceiling(jump)}
        time<-ii+1
#If Birth, sample new pi,mu, and sigma#
if(decision == 'birth'){
        k<-length(unique(data$cluster))
        y1<-rgamma(1,1,1)
        y2<-rgamma(1,k,1)
        new.pi<- y1/(y1+y2)
        data$pi<-data$pi*(1-new.pi)
    new.mu<-rtmvnorm(1,mean=xi,sigma=solve(kappa),lower=c(min(data$X),
     min(data$Y),min(data$Z)),upper=c(max(data$X),max(data$Y),max(data$Z)))
    new.sig<-rigamma(3,g,h)
        new.phi<-rmvnorm(1,new.mu,new.sig*I)
}
#If Death, pick which cluster is removed#
if(decision == 'death'){
        cluster.dead<-as.numeric(comp.decision(d.rate))
        old.pi<-data$pi[which(data$cluster == cluster.dead)][1]
        data$pi<-data$pi/(1-old.pi)
}
#Updaing individual clustering labeling, Z#
param.temp<-cbind(do.call("rbind", as.list(by(data[,
c("cluster","pi","mu_X","mu_Y","mu_Z","sig_X","sig_Y","sig_Z")],
data$cluster,tail,n=1))))
if(decision == 'death'){param.temp<-param.temp[-which(
param.temp$cluster == cluster.dead),]}
```

```
if(decision == 'birth'){param.temp<-rbind(param.temp,
c(max(data$cluster)+1,new.pi,new.mu,new.sig,new.phi))}
for(j in 1:n){
        if(nrow(param.temp) > 1){
        individ.prob<-apply(param.temp,1,function (x) as.numeric(x[2])*
        ((alpha1/(alpha1+nrow(data[data$cluster ==x[1],])))*
        dmvnorm(c(data$phi_X[j],data$phi_Y[j],data$phi_Z[j]),
        c(as.numeric(x[3]),as.numeric(x[4]),as.numeric(x[5])),
        (c(as.numeric(x[6]),as.numeric(x[7]),as.numeric(x[8]))*diag(3)))
        +(1/(alpha1+nrow(data[data$cluster ==x[1],])))*
        (length(data$cluster_e[which(data$cluster == x[1] &
    data$cluster_e == data$cluster_e[j])])))
                }else{individ.prob<-1}
                zj<-rmultinom(1,1,individ.prob)
                zj.row<-which(zj == max(zj), arr.ind=TRUE)[1]
                data$cluster[j]<-param.temp$cluster[zj.row]
                data$pi[j]<-param.temp$pi[zj.row]
                data$mu_X[j]<-param.temp$mu_X[zj.row]
                data$mu_Y[j]<-param.temp$mu_Y[zj.row]
                data$mu_Z[j]<-param.temp$mu_Z[zj.row]
                data$sig_X[j]<-param.temp$sig_X[zj.row]
                data$sig_Y[j]<-param.temp$sig_Y[zj.row]
                data$sig_Z[j]<-param.temp$sig_Z[zj.row]
        }
        #Updating each cluster's mixing proportion, pi#
        total.clusters<-sort(unique(data$cluster))
        if(length(total.clusters)>= 2){
                clus.loc<-0
                alpha.test<-c(apply(table(data$cluster),1,max))
                pi_conditional<-rdirichlet(1, alpha.test+1)
        }else{clus.loc<-0
                pi_conditional<-1}
        loc<-0
        for(u in total.clusters){
                loc<-loc+1
                data$pi[which(data$cluster == u)]<-pi_conditional[loc]
        }
################
for(ee in unique(data$cluster)){
  nni<-which(data$cluster == ee)
  ni<-length(nni)
  mu.temp<-c(data$mu_X[nni][1],data$mu_Y[nni][1],data$mu_Z[nni][1])
  sigma.temp<-c(data$sig_X[nni][1],data$sig_Y[nni][1],data$sig_Z[nni][1])
  cluster.current<-data.frame(cbind(do.call("rbind",
as.list(by(data[data$cluster==ee,c("phi_X","phi_Y","phi_Z")],
data$cluster_e[data$cluster==ee],tail,n=1))),
table(data$cluster_e[data$cluster==ee])))[,-4]
for(e in nni){
        if(any(cluster.current$Freq==0)==TRUE){cluster.current=
cluster.current[-which(cluster.current$Freq==0),]}
        remove<-which(rownames(cluster.current)==unique(data$cluster_e[e]))
        cluster.current[remove,4]<-cluster.current[remove,4]-1
        phi_A <-rmvnorm(1,mu.temp,sigma.temp*I)
                combo[c(1,2,4),1]<-data$phi_X[e]
                combo[c(3,5,6,7),1]<-phi_A[1,1]
                combo[c(1,3,5),2]<-data$phi_Y[e]
                combo[c(2,4,6,7),2]<-phi_A[1,2]
                combo[c(2,3,6),3]<-data$phi_Z[e]
                combo[c(1,4,5,7),3]<-phi_A[1,3]
                combo[,4]<-(alpha1/m)
```

```
        cluster.aux<-data.frame(rbind(cluster.current,combo))
        max.cluster<-eval(1+max(data$cluster_e))
        prob<-apply(cluster.aux,1,function(x) x[4]*
        dmvnorm(c(data$X[e],data$Y[e],data$Z[e]),
        c(data$phi_Px[e],data$phi_Py[e],data$phi_Pz[e])
        +c(x[1],x[2],x[3]),sig_S[ii]*I))
        new_cluster <- rmultinom(1, 1, {(prob)/sum(prob)})
        rownames(new_cluster) = names(prob)
        inds = which(new_cluster == max(new_cluster), arr.ind=TRUE)
        rnames = rownames(new_cluster)[inds[,1]]
        ind<-which(rownames(cluster.aux) == rnames)
        data$phi_X[e] <- cluster.aux[ind,1]
        data$phi_Y[e] <- cluster.aux[ind,2]
        data$phi_Z[e] <- cluster.aux[ind,3]
        if(ind %ni% (nrow(cluster.aux)-6):nrow(cluster.aux)){
    data$cluster_e[e] <- as.numeric(rnames);
    cluster.current[ind,4]<-cluster.current[ind,4]+1
        }else{
    rownames(cluster.aux)[ind]<-data$cluster_e[e]<-max.cluster;
    cluster.aux[ind,4]<-1;
    cluster.current<-rbind(cluster.current,cluster.aux[ind,])
    }
} #end of single cluster
#Updating phi_rj within each cluster_e after assignment#
        for(gg in unique(data$cluster_e[nni])){
        phi.temp<-data[which(data$cluster_e == gg & data$cluster==ee),]
                mean.obs<-apply(phi.temp[,c("X","Y","Z")] -
            phi.temp[,c("phi_Px","phi_Py","phi_Pz")],2,mean)
                new.phi<-rmvnorm(1,as.numeric(solve(nrow(phi.temp)*
            solve(sig_S[ii]*I)+solve(sigma.temp*I))%*%
            (nrow(phi.temp)*solve(sig_S[ii]*I)%*%mean.obs
            + solve(sigma.temp*I)%*%mu.temp)),
            solve(nrow(phi.temp)*solve(sig_S[ii]*I)+solve(sigma.temp*I))*I)
        data$phi_X[data$cluster_e == gg & data$cluster==ee]<-new.phi[1]
        data$phi_Y[data$cluster_e == gg & data$cluster==ee]<-new.phi[2]
        data$phi_Z[data$cluster_e == gg & data$cluster==ee]<-new.phi[3]
    } #end of phi_rj update
    #Updating mu and sigma within each cluster#
    #mu conditional#
    mu.temp.conditional<-rmvnorm(1,solve(solve(kappa)+ni*
    solve(sigma.temp*I))%*%(solve(kappa)%*%xi+ni*solve(sigma.temp*I)%*%
    c(mean(data$phi_X[nni]),mean(data$phi_Y[nni]),mean(data$phi_Z[nni]))),
    solve(solve(kappa)+ni*solve(sigma.temp*I)))
        data$mu_X[nni]<-mu.temp.conditional[1]
        data$mu_Y[nni]<-mu.temp.conditional[2]
        data$mu_Z[nni]<-mu.temp.conditional[3]
        #sigma conditional#
        distance3<-sum((sqrt((data$phi_X[nni]-mu.temp[1])*
    (data$phi_X[nni]-mu.temp[1])+(data$phi_Y[nni]-mu.temp[2])*
    (data$phi_Y[nni]-mu.temp[2])+(data$phi_Z[nni]-mu.temp[3])*
    (data$phi_Z[nni]-mu.temp[3])))^2)
        beta = 0.5*distance3
        sigma.temp.conditional <- rigamma(3,(ni/2)+ g, beta + h)
        data$sig_X[nni]<-sigma.temp.conditional[1]
        data$sig_Y[nni]<-sigma.temp.conditional[2]
        data$sig_Z[nni]<-sigma.temp.conditional[3]
} #end of all clusters
#Updating Study Effect p_r#
###Study Clustering###
cluster.current2<-data.frame(cbind(do.call("rbind",
```

```
as.list(by(data[,c("phi_Px","phi_Py","phi_Pz")],data$study,tail,n=1))),
table(data$study[data$indicator==1])))[,-4]
for (r in unique(data$sub_study)){
if(any(cluster.current2$Freq==0)==TRUE){
    cluster.current2=cluster.current2[-which(cluster.current2$Freq==0),]
}
remove<-which(rownames(cluster.current2) ==
unique(data$study[data$sub_study==r]))
cluster.current2[remove,4]<-cluster.current2[remove,4]-1
phi_A2 <- rtmvnorm(1,c(mu_conditional[ii,]), sigma_conditional[ii]*diag(3),
    lower=-limit,upper=limit)
    combo3[c(1,2,4),1]<-data$phi_Px[data$sub_study==r&data$indicator==1]
    combo3[c(3,5,6,7),1]<-phi_A2[1,1]
    combo3[c(1,3,5),2]<-data$phi_Py[data$sub_study==r&data$indicator==1]
    combo3[c(2,4,6,7),2]<-phi_A2[1,2]
    combo3[c(2,3,6),3]<-data$phi_Pz[data$sub_study==r&data$indicator==1]
    combo3[c(1,4,5,7),3]<-phi_A2[1,3]
    combo3[,4]<-(alpha/m)
        cluster.aux<-data.frame(rbind(cluster.current2,combo3))
        max.study<-eval(1+max(data$study))
        prob2<-apply(cluster.aux,1,function(x) x[4]*
    study.like(r,x[1],x[2],x[3]))
        new_cluster <- rmultinom(1, 1, {(prob2)/sum(prob2)})
        rownames(new_cluster) = names(prob2)
        inds = which(new_cluster == max(new_cluster), arr.ind=TRUE)
        rnames = rownames(new_cluster)[inds[,1]]
        ind<-which(rownames(cluster.aux) == rnames)
                data$phi_Px[data$sub_study==j] <- cluster.aux[ind,1]
                data$phi_Py[data$sub_study==j] <- cluster.aux[ind,2]
                data$phi_Pz[data$sub_study==j] <- cluster.aux[ind,3]
        if(ind %ni% (nrow(cluster.aux)-6):nrow(cluster.aux)){
    data$study[data$sub_study==r] <- as.numeric(rnames);
    cluster.current2[ind,4]<-cluster.current2[ind,4]+1
        }else{
    rownames(cluster.aux)[ind]<-data$study[data$sub_study==r]<-max.study;
    cluster.aux[ind,4]<-1;
    cluster.current2<-rbind(cluster.current2,cluster.aux[ind,])
    }
} #end of loop for study
#Updating each study's parameter, pr#
for(d in unique(data$study)){
        temp<-data[data$study == d,]
        n.study<-length(unique(temp$sub_study))
        mean.temp<-c(mean(temp$X-temp$phi_X),mean(temp$Y-temp$phi_Y),
        mean(temp$Z-temp$phi_Z))
    data[data$study == d, c("phi_Px","phi_Py","phi_Pz")]<-
    matrix(rep(rtmvnorm(1,as.numeric(solve(n.study*solve(sig_S[ii]*I)
    +solve(sigma_conditional[ii]*I))%*%(n.study*solve(sig_S[ii]*I)%*%
    mean.temp+solve(sigma_conditional[ii]*I)%*%mu_conditional[ii,])),
    solve(n.study*solve(sig_S[ii]*I)+solve(sigma_conditional[ii]*I)),
    lower=-limit,upper=limit),each=nrow(temp)),nrow=nrow(temp),ncol=3)
}
        #Mu Full Conditional Posterior
        study.temp<-data[data$indicator == 1,c("phi_Px","phi_Py","phi_Pz")]
        mean.study <- apply(study.temp,2,mean)
        mu_hyper<-rmvnorm(1,solve(nn*solve(sigma_conditional[ii]*I)
    +solve(sigma_mu*I))%*%(nn*solve(sigma_conditional[ii]*I)%*%mean.study
    +solve(sigma_mu*I)%*%mu_mu),solve(nn*solve(sigma_conditional[ii]*I)
    +solve(sigma_mu*I)))
        mu_conditional <- rbind(mu_conditional,mu_hyper)
```

```
   #Sigma Full Conditional Posterior#
   distance4 <- sum(diag(as.matrix(study.temp - mu_conditional[ii+1,])
   %*% I %*% as.matrix(t(study.temp - mu_conditional[ii+1,]))))
   beta2 = 0.5*distance4
   sigma_conditional<-c(sigma_conditional,rigamma(1,((nn/2)+q),beta2+l))
        #Updating Sigma for overall data#
        distance5 <- 0.5*(sum(diag(as.matrix(data[,c("X","Y","Z")]
        -data[,c("phi_Px","phi_Py","phi_Pz")]
        -data[,c("phi_X","phi_Y","phi_Z")])%*%
         I%*%t(as.matrix(data[,c("X","Y","Z")]
         -data[,c("phi_Px","phi_Py","phi_Pz")]-
         data[,c("phi_X","phi_Y","phi_Z")])))))
        sig_S<-c(sig_S,rigamma(1,((n/2)+aa),distance5+bb))
###least squares clustering###
#Updating clustering indicator Matrix#
if(ii >= burn.in & ii < working){
        for(ind in unique(data$cluster)){
        matrix_indicator<-which(data$cluster == ind)
        cluster_matrix[matrix_indicator,matrix_indicator] <-
        cluster_matrix[matrix_indicator,matrix_indicator] + 1}
}
#Calculating the average probability matrix at the #
#last Wth or working iteration#
if(ii == working){
        cluster_average<-cluster_matrix/{ii-burn.in}
        best.distance<-10000000
}
#F or final iterations#
if(ii >= working){
        cluster_matrix.temp<-matrix(0,nrow=n,ncol=n)
                for(ind in unique(data$cluster)){
                matrix_indicator<-which(data$cluster == ind)
                cluster_matrix.temp[matrix_indicator,matrix_indicator] <-
            cluster_matrix.temp[matrix_indicator,matrix_indicator] + 1
                }
        ls.distance<-sum((cluster_matrix.temp-cluster_average)*
    (cluster_matrix.temp-cluster_average))
                if(ls.distance <= best.distance){
                best.distance<-ls.distance
                cluster.data<-data
                updated.run<-ii
                }
        #Loglikelihoods#
        param.temp<-data.frame(cbind(do.call("rbind",
    as.list(by(data[,c("cluster","pi","mu_X","mu_Y","mu_Z","sig_X",
    "sig_Y","sig_Z","phi_X","phi_Y","phi_Z")],data$cluster,tail,n=1))),
    table(data$cluster)))
        clus<-length(unique(data$cluster))
        probs<-matrix(0,nrow=nrow(data),ncol=clus)
        for(k in 1:clus){
        probs[,k]<-apply(as.matrix(data[,c("phi_X","phi_Y","phi_Z",
    "cluster","cluster_e")]),1,function (x)
        {alpha1/{alpha1+as.numeric(param.temp[k,13])}}*
        dmvnorm(x[1:3],as.numeric(param.temp[k,3:5]),
    (as.numeric(param.temp[k,6:8])*diag(3)))
        +(as.numeric(length(data$cluster_e[data$cluster==param.temp[k,1]&
    data$cluster_e == x[5]]))/(alpha1+as.numeric(param.temp[k,13]))))
                }
        full.like<-(probs%*%param.temp[,2])
        log.like[ii]<-sum(log(full.like))
}
```

```
#DIC Calculations#
if(ii > burn.in){Dev[ii]<--2*log.like[ii]}
} #end of loops
DIC<-mean(na.omit(Dev)) + {var(na.omit(Dev))/2}
```