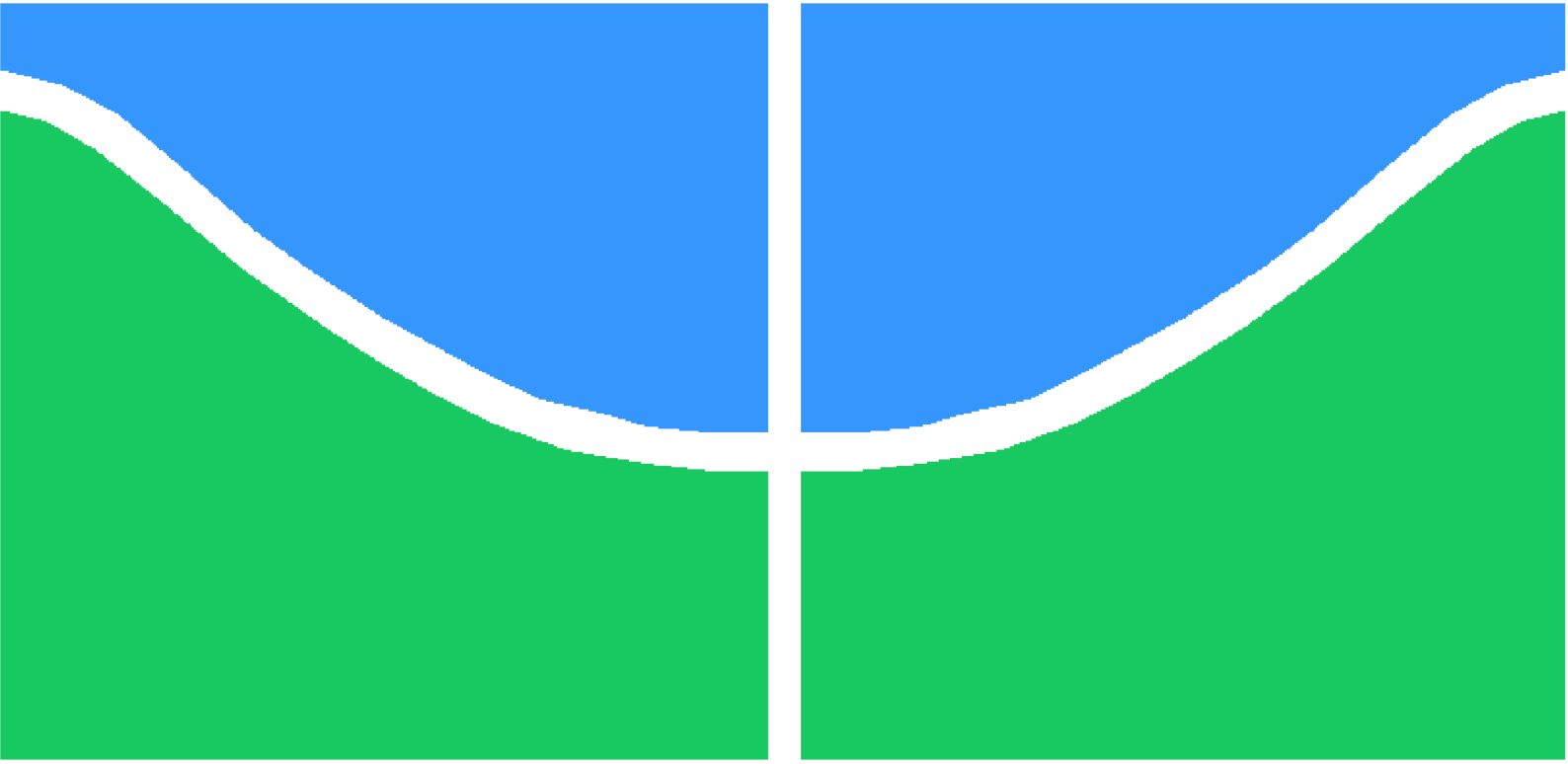


Universidade de Brasília - UnB

Instituto de Ciências Exatas - IE

Departamento de Estatística - EST

Uma estatística scan espacial bayesiana para dados com excesso de zeros

Autor: Lucas Barbosa Fernandes

Orientador: Prof. Dr. André Luiz Fernandes Cançado

Brasília, DF

2015

Lucas Barbosa Fernandes

Uma estatística scan espacial bayesiana para dados com excesso de zeros

Dissertação submetida ao programa de Pós-Graduação em Estatísticada Universidade de Brasília, como requisito parcial para obtenção do Título de Mestre em Estatística.

Universidade de Brasília - UnB

Instituto de Ciências Exatas - IE

Departamento de Estatística - EST

Orientador: Prof. Dr. André Luiz Fernandes Cançado

Brasília, DF

2015

Dedico o trabalho à minha afilhada Alice.

Agradecimentos

Agradeço primeiramente a Deus pelo dom da vida.

Aos meus pais e irmã, por todo apoio e compreensão.

Ao meu orientador André, por todo o auxílio e trabalho despendidos para a conclusão desta dissertação.

Aos meus colegas e chefes da Conab, pela compreensão e auxílio nesses anos de mestrado.

Aos meus amigos de graduação e mestrado, pelo auxílio nas horas de estudo.

Aos meus amigos de música, minha válvula de escape. Com certeza não chegaria aqui sem ela.

Aos meus amigos e compadres Fabrício e Adriana, pelo presente de poder ser padrinho da pequena Alice.

Ao amigo Samuel Dornelas (in memoriam), que nos deixou em fevereiro desse ano. Pessoa essencial na graduação e parceiro do trabalho final de curso.

"Do not fear mistakes.
There are none."
(Miles Davis)

Resumo

A análise e detecção de conglomerados (ou *clusters*) espaciais se mostra de grande utilidade para subsidiar decisões em áreas de saúde e segurança, por exemplo. O método Scan Circular de Kulldorff, um dos mais difundidos para detecção de conglomerados espaciais, recebeu extensões que permitem um melhor desempenho na presença de um grande número de zeros, além de uma abordagem Bayesiana, que possui vantagens computacionais e em termos de incorporação de informações à *priori*. Este trabalho apresenta adaptações dos trabalhos de Kulldorff (1997), Cançado et al. (2014) e Neill et al. (2006), com as estatísticas Scan Binomial, Scan ZIB, Scan ZIB-EM e Scan Beta-Binomial, e propõe as estatísticas Scan ZIBB e Scan ZIBB-Gibbs, que utilizam a abordagem bayesiana em dados com excesso de zeros. Os métodos são comparados com dados simulados e aplicados ao estudo de casos de Febre Hemorrágica do Dengue (FHD) no estado do Rio de Janeiro (2011). São obtidos resultados positivos para os métodos propostos.

Palavras-chaves: Estatística. Análise espacial - estatística. Estatística Bayesiana. Inflação de Zeros. Detecção de Cluster.

Abstract

The analysis and detection of spacial cluster are useful for support decisions on many areas, like health and public security. Kulldorff's Circular Scan method, one of the most known and used, received extensions for better performance on problems that include a great presence of zeros and a bayesian approach, which presents computational advantages and allows the incorporation of prior information. This work presents a review and an adaptation of the works of Kulldorff (1997), Cançado et al. (2014) and Neill et al. (2006) (Scan Binomial, Scan ZIB, Scan ZIB-EM and Scan Beta-Binomial statistics) and proposes the Scan ZIBB and Scan ZIBB-Gibbs statistics, using the Bayesian approach for zero-inflated data. The methods are compared with simulated data and applied to the study of cases of Dengue Hemorrhagic Fever (FHD) in the state of Rio de Janeiro (2011). The proposed methods exhibit good results.

Key-words: Statistics. Spacial Analysis - statistics. Bayesian statistics. Cluster Detection. Zero-Inflated Binomial.

Lista de ilustrações

Figura 1 – Exemplo: Construção de zonas por janelas circulares 20

Figura 2 – Localização do cluster no cenario A 39
Figura 3 – Localização do cluster no cenario B 40
Figura 4 – Localização do cluster no cenario C 40
Figura 5 – Localização do cluster no cenario D 41
Figura 6 – Mapa de quartis de controles de FHD (pacientes curados) - RJ (2011) . 46
Figura 7 – Mapa de quartis de casos de óbitos por FHD - RJ (2011) 46
Figura 8 – Clusters Detectados 47
Figura 9 – Mapa de probabilidades- Beta-Binomial 48
Figura 10 –Mapa de probabilidades - ZIBB-Gibbs 48

Lista de tabelas

Tabela 1 – Riscos relativos dentro do cluster, número de casos esperados sob H_0 (m_0) e número de casos esperados sob H_1 (m_1) 42
Tabela 2 – Fator de Bayes - Interpretação 43
Tabela 3 – Resultados das Simulações 44
Tabela 4 – Resumo para os clusters de óbitos por FHD detectados 47

Sumário

Introdução **11**

I Metodologia **14**

1 Revisão Metodológica **15**
1.1 Introdução 15
1.2 Scan Circular de Kulldorff 15
1.2.1 Modelo Binomial 16
1.2.2 Modelo Poisson 17
1.2.3 Encontrando o cluster mais verossímil 18
1.2.3.1 Matriz de distâncias 19
1.2.3.2 Encontrando candidatos a Clusters 19
1.2.4 Verificação da significância do Cluster 21
1.3 Scan para dados com excesso de zeros 21
1.3.1 Modelo ZIB 22
1.3.1.1 Procedimento para δ_i conhecido 22
1.3.1.2 Algoritmo EM para δ_i desconhecidos 24
1.3.2 Encontrando o cluster mais verossímil 25
1.3.3 Verificação da significância do Cluster 25
1.4 Abordagem Bayesiana 26
1.4.1 Scan Beta-Binomial 27
1.4.1.1 Algoritmo Scan-Bayesiano 28
1.4.2 Escolha das *prioris* 29

2 ZIB Bayesiano **31**
2.1 Introdução 31
2.2 Metodologia 31
2.2.1 Dados completos 31
2.2.2 Dados Incompletos 32
2.2.2.1 Amostrador de Gibbs para δ_i desconhecidos 33

II Resultados **35**

3 Resultados **36**
3.1 Implementação computacional 36

3.2 Cenários 38
3.2.1 Simulações 39
3.2.2 Análise de Desempenho 42
3.3 Aplicações em Dados Reais 45
3.3.1 Dados 45
3.3.2 Resultados e Discussão 45

4 Considerações finais 50
4.1 Trabalhos Futuros 50

Referências 52

Anexos 56

ANEXO A Programações 57
A.1 Scan Binomial 57
A.2 Scan ZIB 60
A.3 Scan ZIB-EM 65
A.4 Scan Beta-Binomial 69
A.5 Scan ZIBB 73
A.6 Scan ZIBB-Gibbs 77

Introdução

A Estatística Espacial é o ramo da Estatística que estuda e analisa padrões de dados distribuídos no espaço, com métodos exploratórios e inferenciais específicos. Segundo Druck et al. (2004):

> *"Compreender a distribuição espacial de dados oriundos de fenômenos ocorridos no espaço constitui hoje um grande desafio para a elucidação de questões centrais em diversas áreas do conhecimento, seja em saúde, em ambiente, em geologia, em agronomia, entre tantas outras."*

Três tipos de dados são analisados pela Estatística Espacial:

- Dados pontuais: Geralmente são representados no formato (x_i, y_i), indicando as coordenadas geográficas das ocorrências do evento de interesse. Por exemplo, casos de uma determinada doença ou crime.
- Dados de área, ou agregados. São obtidos quando não estão disponíveis as coordenadas de cada ocorrência do evento, mas apenas o número total de ocorrências em cada região - geralmente uma unidade administrativa - do mapa em estudo. Por exemplo, o número total de casos de uma doença ou crime em cada município de um estado.
- Dados contínuos ou de superfície, obtidos ao se realizar medições em determinadas localizações do mapa. Cada elemento do conjunto de dados é formado por uma tripla (x_i, y_i, z_i) que corresponde à coordenada geográfica aliada à medição feita naquela localização. Por exemplo, medição de temperatura ou umidade em determinadas localizações.

As análises de dados pontuais e de área envolvem, principalmente, questões relativas à distribuição geográfica dos eventos, ao passo que na análise de dados de superfície o pesquisador geralmente deseja prever, baseado nas medições feitas, o valor da medição em pontos que não tenham sido amostrados.

No caso específico de dados pontuais, é desejável saber se a distribuição dos dados no espaço se dá de forma aleatória ou segue algum padrão. Uma área ou região que possui um alta incidência de casos ou um número significativamente maior que o esperado se caracteriza como um cluster ou conglomerado.

Ao longo dos anos, foram desenvolvidos diversos métodos para detecção de conglomerados, como o *Geographical Analysis Machine* (GAM) de Openshaw et al. (1987), o método de Besag e Newell (1991) e o Scan Circular de Kulldorff (1997). Os dois primeiros eram métodos exploratórios, enquanto o Scan trouxe a possibilidade de se realizar inferências.

Por esse motivo, além de sua fácil implementação, a estatística Scan Circular de (Kulldorff, 1997) se tornou uma das mais utilizadas e foram propostas extensões para o modelo normal (Jung et al., 2010), exponencial (Lawson e Kleinman, 2005), para análises multivariadas (Kulldorff et al., 2007) e dados ordinais (Jung et al., 2007).

Foram publicadas, também, diversas aplicações da estatística Scan Circular, como em detecção de clusters de doenças respiratórias infecciosas (Bakker et al., 2004; Elias et al., 2006), doenças sexualmente transmissíveis (Jennings et al., 2005; Cuadros et al., 2013), vigilância sindrômica (Kleinman et al., 2005; Besculides et al., 2005), doenças do fígado (Ala et al., 2006), diabetes (Green et al., 2003), criminologia (Minamisava et al., 2009), entre outras.

Entre outras extensões de destaque, o artigo de Neill et al. (2006) apresentou uma abordagem Bayesiana da estatística Scan Espacial. As vantagens da utilização desta abordagem passa pela possibilidade de utilizar informações *a priori*, a utilização de verossimilhanças marginais, que torna o modelo mais flexível e menos propenso a *overfitting* (sobreajuste), além da melhoria no tempo de processamento, já que, ao contrário da abordagem frequentista, não é necessário realizar reamostragem, o que em alguns casos pode ser computacionalmente intensivo. Nesse trabalho é apresentada a versão da estatística Scan Beta-Binomial.

Já Cançado et al. (2014) propuseram uma extensão da estatística Scan para problemas em que os dados apresentam excesso de zeros. Nesses casos, a abordagem tradicional, utilizando um processo de Poisson, pode produzir inferências viesadas. Para resolver tal questão, foram propostas as estatísticas Scan ZIP e Scan ZIP-EM.

A proposta desse trabalho é apresentar uma adaptação dessas duas extensões da estatística Scan: um método de detecção de conglomerados para dados com excesso de zeros com abordagem Bayesiana, gerando as estatísticas Scan ZIBB e Scan ZIBB-Gibbs.

Os métodos foram implementados em linguagem R e comparados utilizando dados simulados e reais.

Objetivos

Objetivo Geral

Apresentar uma abordagem Bayesiana para a estatística Scan para dados com excesso de zeros.

Objetivos Específicos

- Apresentar e revisar as estatísticas Scan Binomial, Scan ZIB, Scan ZIB-EM e Scan Beta-Binomial.
- Propor as estatísticas Scan ZIBB e Scan ZIBB-Gibbs
- Implementar os métodos em linguagem R.
- Aplicar e comparar os métodos com dados simulados e dados reais.

Parte I

Metodologia

1 Revisão Metodológica

1.1 Introdução

Esse capítulo visa descrever em detalhes a estatística Scan, proposta por Kulldorff (1997), para realizar a detecção e inferência de conglomerados espaciais, bem como a extensão desta proposta para dados com excesso de zeros, conforme proposto por Cançado et al. (2014), e a abordagem bayesiana, conforme Neill et al. (2006).

1.2 Scan Circular de Kulldorff

Considere um mapa dividido em m regiões, em que cada região i possui uma *população em risco* n_i e um *número de casos* x_i, que representa o número de ocorrências de um evento de interesse (ex.: casos de uma doença, número de crimes etc.). Sejam $N = \sum_{i=1}^{m} n_i$ e $C = \sum_{i=1}^{m} x_i$ que representam, respectivamente, a população total em risco e o número total de casos.

Seja uma zona z, i.e, um subconjunto de regiões conexas do mapa, e Z o conjunto das mesmas. Um cluster é definido como uma zona específica onde a probabilidade θ_z de um indivíduo ser um caso é maior que nas demais.

Considere $x_z = \sum_{i \in z} x_i$, o número de casos na zona z, $x_{\bar{z}} = \sum_{i \notin z} x_i$, o número de casos fora de z, e $n_z = \sum_{i \in z} n_i$ e $n_{\bar{z}} = \sum_{i \notin z} n_i$, os tamanhos populacionais correspondentes.

Sendo assim, a estatística *Scan* de Kulldorff é definida a partir de um teste de razão de verossimilhança, associado às seguintes hipóteses:

- H_0: $\theta_z = \theta_0$.
- H_1: existe uma zona z tal que $\theta_z > \theta_0$,

onde θ_z é a probabilidade de ocorrer um caso na zona z e θ_0 a probabilidade de ocorrer um caso fora dela. A zona z será considerada um cluster caso o teste rejeite a hipótese nula.

O número de casos x_i de uma região i pode ser modelada a partir das distribuições Binomial e Poisson, sendo que a forma da estatística do teste de razão de verossimilhança será diferente para os dois casos, como veremos a seguir.

1.2.1 Modelo Binomial

Uma forma comum de modelar o número de casos x_i é assumindo a distribuição Binomial:

$$x_i \sim Bin(n_i, \theta).$$

Sob H_0, temos $\theta = \theta_0$. Sendo assim, a verossimilhança assume a forma:

$$\mathcal{L}_0(\mathbf{x}, \theta_0) = \left[\prod_{i=1}^{m} \binom{n_i}{x_i}\right] \theta_0^C (1-\theta_0)^{N-C}. \tag{1.1}$$

Para a log-verossimilhança, tem-se que:

$$\begin{aligned} \ell_0(\mathbf{x}, \theta_0) = \log(\mathcal{L}_0(\mathbf{x}, \theta_0)) = \sum_i \log \binom{n_i}{x_i} \\ + C \log \theta_0 + (N-C)\log(1-\theta_0). \end{aligned} \tag{1.2}$$

Derivando em relação a θ_0 e igualando a zero, temos

$$\frac{\partial \ell_0}{\partial \theta_0} = \frac{C}{\theta_0} - \frac{N-C}{1-\theta_0} = 0. \tag{1.3}$$

Logo, tem-se $\hat{\theta}_0 = \frac{C}{N}$.

Sob H_1, $\begin{cases} \theta_i = \theta_z & \text{, se } i \in z \\ \theta_i = \theta_0 & \text{, se } i \notin z. \end{cases}$

Nesse caso, a verossimilhança será escrita como:

$$\begin{aligned} \mathcal{L}(\mathbf{z}, \mathbf{x}, \theta_0, \theta_z) = \left[\prod_{i \in z} \binom{n_i}{x_i}\right] \theta_z^{\sum_{i\in z} x_i} (1-\theta_z)^{\sum_{i\in z} n_i - \sum_{i\in z} x_i} \\ \times \left[\prod_{i \notin z} \binom{n_i}{x_i}\right] \theta_0^{\sum_{i\notin z} x_i} (1-\theta_0)^{\sum_{i\notin z} n_i - \sum_{i\notin z} x_i}. \end{aligned} \tag{1.4}$$

Aplicando o logaritmo:

$$\begin{aligned} \ell(\mathbf{z}, \mathbf{x}, \theta_0, \theta_z) = \log(\mathcal{L}) = \sum_{i\in z} \log \binom{n_i}{x_i} + x_z \log(\theta_z) + (n_z - x_z)\log(1-\theta_z) \\ + \sum_{i\notin z} \log \binom{n_i}{x_i} + x_{\bar{z}} \log(\theta_0) + (n_{\bar{z}} - x_{\bar{z}})\log(1-\theta_0). \end{aligned} \tag{1.5}$$

Maximizando-se $\ell(\mathbf{z}, \mathbf{x}, \theta_0, \theta_z)$, tem-se:

$$\frac{\partial \ell}{\partial \theta_0} = 0 \Rightarrow \hat{\theta_0} = \frac{x_{\bar{z}}}{n_{\bar{z}}}, \tag{1.6}$$

$$\frac{\partial \ell}{\partial \theta_z} = 0 \Rightarrow \hat{\theta_z} = \frac{x_z}{n_z}. \tag{1.7}$$

A razão de verossimilhança para o Modelo Binomial é definida como:

$$\begin{aligned} \lambda = & \left[\prod_{i \in z} \binom{n_i}{x_i}\right] \theta_z^{\sum_{i \in z} x_i} (1-\theta_z)^{\sum_{i \in z} n_i - \sum_{i \in z} x_i} \\ & \times \left[\prod_{i \notin z} \binom{n_i}{x_i}\right] \theta_0^{\sum_{i \notin z} x_i} (1-\theta_0)^{\sum_{i \notin z} n_i - \sum_{i \notin z} x_i} \\ & \times \frac{1}{\prod_{i=1}^{m} \binom{n_i}{x_i} \theta^{\sum_i x_i} (1-\theta)^{\sum_i n_i - \sum_i x_i}} = \\ & = \frac{{\theta_z}^{x_z} (1-\theta_z)^{n_z - x_z} {\theta_0}^{x_{\bar{z}}} (1-\theta_0)^{n_{\bar{z}} - x_{\bar{z}}}}{\theta^C (1-\theta)^{N-C}}. \end{aligned} \tag{1.8}$$

1.2.2 Modelo Poisson

Outra forma usual para se desenvolver a estatística *Scan* assume $x_i \sim Poisson(n_i \theta_i)$. Da mesma forma que no modelo Binomial, sob H_0, tem-se que $\theta_i = \theta_0$ e:

$$\begin{aligned} \mathcal{L}_0(\mathbf{x}, \theta_0) = \prod_{i=1}^{m} \frac{e^{-n_i \theta_0} (n_i \theta_0)^{x_i}}{x_i!} &= \frac{e^{-\sum_i n_i \theta_0} \times \prod_i n_i^{xi} \times \theta_0^{\sum_i x_i}}{\prod_i (x_i!)} \\ &= \frac{e^{-N\theta_0} \times \prod_i n_i^{x_i} \times \theta_0^C}{\prod_i (x_i!)}. \end{aligned} \tag{1.9}$$

A log-verossimilhança é dada por:

$$\begin{aligned} \ell_0(\mathbf{x}, \theta_0) = \log(\mathcal{L}_0(\mathbf{x}, \theta_0)) = & - N\theta_0 + \sum_i x_i \log n_i \\ & + C \log \theta_0 - \sum_i \log(x_i!). \end{aligned} \tag{1.10}$$

Derivando em relação a θ_0 e igualando a zero, se obtém:

$$\frac{\partial \ell_0}{\partial \theta_0} = -N + \frac{C}{\theta_0} = 0. \tag{1.11}$$

Resolvendo-se a equação, resulta-se em $\hat{\theta}_0 = \frac{C}{N}$, como no caso Binomial.

Sob H_1, $\begin{cases} \theta_i = \theta_z & \text{, se } i \in z, \\ \theta_i = \theta_0 & \text{, se } i \notin z. \end{cases}$

Nesse caso, a verossimilhança será escrita como:

$$\begin{aligned}\mathcal{L}(\mathbf{z},\mathbf{x},\theta_0,\theta_z) &= \prod_{i\in z}\frac{e^{-n_i\theta_z}(n_i\theta_z)^{x_i}}{x_i!}\times\prod_{i\notin z}\frac{e^{-n_i\theta_0}(n_i\theta_0)^{x_i}}{x_i!}\\ &= \frac{e^{-\sum_{i\in z}n_i\theta_z}\times\prod_{i\in z}n_i^{x_i}\times\theta_z^{\sum_{i\in z}x_i}}{\prod_{i\in z}(x_i!)}\\ &\times\frac{e^{-\sum_{i\notin z}n_i\theta_0}\times\prod_{i\notin z}n_i^{x_i}\times\theta_0^{\sum_{i\notin z}x_i}}{\prod_{i\notin z}(x_i!)}\end{aligned} \tag{1.12}$$

Aplicando o logaritmo:

$$\begin{aligned}\ell(\mathbf{z},\mathbf{x},\theta_0,\theta_z) = \log(\mathcal{L}) = &-n_z\theta_z+\sum_{i\in z}x_i\log n_i + x_z\log\theta_z - \sum_{i\in z}\log(x_i!)\\ &-n_{\bar{z}}\theta_0+\sum_{i\notin z}x_i\log n_i + x_{\bar{z}}\log\theta_0 - \sum_{i\notin z}\log(x_i!).\end{aligned} \tag{1.13}$$

Maximizando em relação a θ_0 e θ_z:

$$\frac{\partial\ell}{\partial\theta_0}=0\Rightarrow\hat{\theta_0}=\frac{x_{\bar{z}}}{n_{\bar{z}}}, \tag{1.14}$$

$$\frac{\partial\ell}{\partial\theta_z}=0\Rightarrow\hat{\theta_z}=\frac{x_z}{n_z}. \tag{1.15}$$

Similar ao modelo binomial, tem-se que:

$$\begin{aligned}\lambda = &\frac{e^{-\sum_{i\in z}n_i\theta_z}\times\prod_{i\in z}n_i^{x_i}\times\theta_z^{\sum_{i\in z}x_i}}{\prod_{i\in z}(x_i!)}\times\frac{e^{-\sum_{i\notin z}n_i\theta_0}\times\prod_{i\notin z}n_i^{x_i}\times\theta_0^{\sum_{i\notin z}x_i}}{\prod_{i\notin z}(x_i!)}\\ &\times\left(\frac{e^{-\sum_i n_i\theta}\times\prod_i n_i^{xi}\times\theta^{\sum_i x_i}}{\prod_i(x_i!)}\right)^{-1}=\frac{e^{-(n_z\theta_z+n_{\bar{z}}\theta_0-n\theta)}\theta_z^{x_z}\theta_0^{x_{\bar{z}}}}{\theta^C}.\end{aligned} \tag{1.16}$$

1.2.3 Encontrando o cluster mais verossímil

Para os modelos Binomial e Poisson, tem-se, para uma zona z:

$$\lambda_z=\frac{\sup_{\theta_z>\theta_0}\mathcal{L}(\mathbf{z},\mathbf{x},\theta_0,\theta_z)}{\sup_{\theta_z=\theta_0}\mathcal{L}(\mathbf{z},\mathbf{x},\theta_0,\theta_z)}=\left(\frac{x_z/n_z}{C/N}\right)^{x_z}\times\left(\frac{x_{\bar{z}}/n_{\bar{z}}}{C/N}\right)^{x_{\hat{z}}}\times I(x_z/n_z>x_{\bar{z}}/n_{\bar{z}}) \tag{1.17}$$

A estatística Scan é definida por:

$$T=\sup_z\lambda_z=\frac{\sup_{\theta_z>\theta_0}\mathcal{L}(\mathbf{z},\mathbf{x},\theta_0,\theta_z)}{\sup_{\theta_z=\theta_0}\mathcal{L}(\mathbf{z},\mathbf{x},\theta_0,\theta_z)}. \tag{1.18}$$

isto é, o cluster mais verossímil será a zona com maior valor de λ_z dentre todas as zonas candidatas a cluster.

Para encontrar o cluster mais verossímil, define-se o conjunto de zonas candidatas Z e, para elemento $z \in Z$, calcula-se λ_z. Um forma simples e eficiente que é comumente utilizada para definir-se um conjunto Z razoável é através de janelas circulares com diferentes centros e raios. A obtenção dessas janelas e das respectivas zonas candidatas é descrita nas subseções (1.2.3.1) e (1.2.3.2).

1.2.3.1 Matriz de distâncias

Para cada uma das m regiões do mapa, considere um centroide com cordenadas (x_i, y_i), $i = 1, ..., m$. A distância entre duas regiões é dada pela distância entre seus centroides. Então, para duas regiões i e j quaisquer, a distância Euclidiana é dada por:

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{1.19}$$

A matriz quadrada de distâncias entre os centroides é simétrica com m linhas e m colunas, em que cada elemento da matriz representa a distância entre duas regiões. Sendo assim, a matriz tem a forma:

$$D = \begin{vmatrix} 0 & d_{1,2} & \dots & d_{1,j} & \dots & d_{1,m} \\ d_{2,1} & 0 & \dots & d_{2,j} & \dots & d_{2,m} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{i,1} & d_{i,2} & \dots & 0 & \dots & d_{2,m} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{m,1} & d_{m,2} & \dots & d_{m,j} & \dots & 0 \end{vmatrix} \tag{1.20}$$

1.2.3.2 Encontrando candidatos a Clusters

Este processo inicia-se pela construção de zonas, caracterizadas como a aglomeração de uma ou mais regiões próximas. Comumente, são utilizadas janelas circulares para realizar este procedimento.

Por exemplo, iniciando-se pela região 1, tem-se o seguinte vetor coluna correspondente às distâncias para as outras regiões:

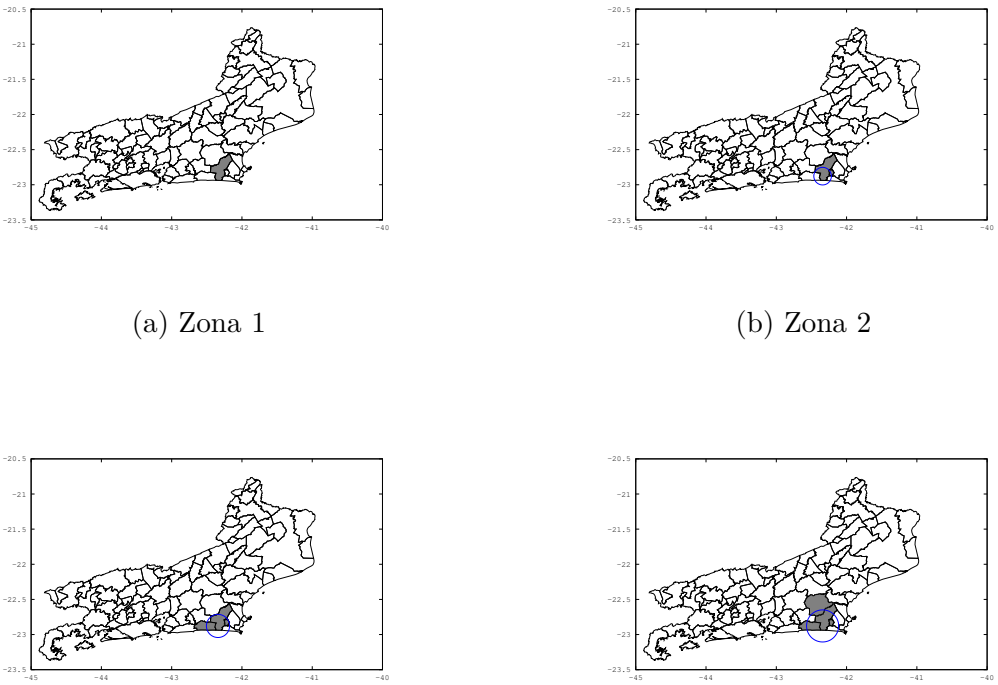
$$\begin{bmatrix} 0 \\ d_{2,1} \\ d_{3,1} \\ \vdots \\ d_{m,1} \end{bmatrix} \tag{1.21}$$

Seja $d_{(j),i}$ a distância da j-ésima região mais próxima da região i, em que $d_{(n),i} > d_{(n-1),i} > \cdots > d_{(3),i} > d_{(2),i}$. O vetor coluna anterior ordenado em ordem crescente de distâncias tem a forma:

$$\begin{bmatrix} 0 \\ d_{(2),1} \\ d_{(3),1} \\ \vdots \\ d_{(m),1} \end{bmatrix} \tag{1.22}$$

A primeira zona selecionada será formada somente pela região 1, ou seja, $z_1 = \{1\}$. Calcula-se, então, o valor λ_{z_1}, conforme (1.17).

A segunda zona é formada por $\{1\}$ e também pela região mais próxima, isto é, a região correspondente à distância $d_{(2),1}$. Esta zona será representada por $z_2 = \{1, (2)\}$. Com os dados da zona z_2, calcula-se λ_{z_2}. O processo iterativo se repete agregando, em cada passo, uma região segundo a distância, até atingir um tamanho máximo de população previamente definido. A figura 1 exemplifica a construção de zonas por meio de janelas circulares, utilizando os municípios do estado do Rio de Janeiro



(a) Zona 1

(b) Zona 2

(c) Zona 3

(d) Zona 4

Figura 1: Exemplo: Construção de zonas por janelas circulares

O cluster mais verossímil será a zona correspondente ao maior valor de λ_z, obedecendo o limite de população máxima, e esta será a estatística T procurada.

1.2.4 Verificação da significância do Cluster

Após a execução do algoritmo para a detecção do cluster mais verossímil, é necessário testar sua significância. Como a distribuição exata de T é desconhecida, é necessário executar uma simulação de Monte Carlo para obter-se a distribuição empírica.

Esse procedimento baseia-se na geração de réplicas do mapa original, distribuindo o número de casos C, $C \in \mathbb{N}^*$, aleatoriamente sob H_0. Seja J o número de réplicas a serem executadas (geralmente milhares). A simulação segue os seguintes passos:

1. Distribuir C casos aleatoriamente de acordo com uma distribuição Multinomial com parâmetros C e n_i/N.
2. Calcule T conforme (1.18) com o novo banco de dados gerado. Armazene esse valor.
3. Repita os passos 2 a 5 um número J de vezes obtendo a distribuição empírica de T, sob H_0.
4. Rejeite, com nível de significância de 5%, a hipótese H_0 de ausência de clusters se $T > P_{95}$, onde T é o valor da estatística de teste obtido para os dados reais, conforme descrito na seção (1.2.3.2), e P_{95} é o 95º percentil da distribuição empírica de T sob H_0.

Esse procedimento pode ser computacionalmente intensivo quando o número de casos é muito grande e/ou o mapa estudado possui muitas regiões.

1.3 Scan para dados com excesso de zeros

Dois tipos de contagem nula podem ocorrer nos dados:

- Zeros amostrais, quando provenientes de uma distribuição Poisson ou Binomial cuja contagem/resposta naquele caso é nula.
- Zeros estruturais, quando, segundo Agresti (2002), é teoricamente impossível de se ter observações.

Agresti (2002) afirma:

> *"Um zero amostral é uma observação com valor 0, e consideramos como uma das observações de contagem. No entanto, um zero estrutural não é uma observação".*

Exemplos de zero estrutural: número de flores quando uma planta já está morta, número de casos de câncer de mama em um município onde não há mamógrafo e etc.

O excesso de zeros é notável em estudos envolvendo doenças raras. Nesse contexto, Gómez-Rubio e López-Quílez (2010) argumentam que a estatística Scan não é adequada para tais problemas onde o número de casos é baixo e o excesso de zeros pode causar estimativas viesadas.

O artigo de Cançado et al. (2014) propõe extensões da estatística Scan, denominadas Scan-ZIP e Scan-ZIP-EM, de forma a modelar a ocorrência de dados com excesso de zeros, obtendo estimadores menos viesados na detecção de clusters. No trabalho, foi utilizada a distribuição Poisson, que resulta na estatística Scan-ZIP. Nesta dissertação será apresentada uma adaptação do método Scan-ZIP para a distribuição Binomial, que denominamos Scan-ZIB.

1.3.1 Modelo ZIB

O modelo ZIB para a estatística Scan é uma adaptação do trabalho de Cançado et al. (2014) (que utiliza a distribuição Poisson) para a distribuição Binomial.

Seja δ_i um indicador de zero estrutural para uma determinada região i. δ_i assume valor 0 para regiões sem zero estrutural e valor 1 com probabilidade p em regiões com zero estrutural. Ou seja, p é a probabilidade da ocorrência de zero estrutural em cada zona. Logo:

$$\delta_i \sim Bernoulli(p)$$

1.3.1.1 Procedimento para δ_i conhecido

Nesse caso, δ_i é uma variável aleatória observável, isto é, após o processo amostral, o valor de δ_i é conhecido para cada zona z.

A distribuição conjunta de x_i e δ_i, onde x_i é o número de casos na região i, é dada por:

$$P(x_i, \delta_i) = p^{\delta_i}\left[(1-p)\binom{n_i}{x_i}\theta_i^{x_i}(1-\theta_i)^{n_i-x_i}\right]^{1-\delta_i} \tag{1.23}$$

onde $x_i = 0, ..., n_i$ e $\delta_i = 0, 1$

Sob $H_0 : \theta_i = \theta$. A verossimilhança da distribuição conjunta será:

$$\begin{aligned}\mathcal{L}_0(\delta, \mathbf{x}) &= \prod_{i=1}^{m} p^{\delta_i}\Big[(1-p)\binom{n_i}{x_i}\theta^{x_i}(1-\theta)^{n_i-x_i}\Big]^{1-\delta_i} \\ &= \prod_{i=1}^{m}\binom{n_i}{x_i} p^{\sum_i \delta_i}(1-p)^{\sum_i (1-\delta_i)}\theta^{\sum_i x_i(1-\delta_i)}(1-\theta)^{\sum_i n_i(1-\delta_i)-\sum_i x_i(1-\delta_i)}\end{aligned} \tag{1.24}$$

Aplicando o logaritmo:

$$\begin{aligned}\log(\mathcal{L}_0) = \ell_o = &\sum_i \binom{n_i}{x_i} + \sum_i \delta_i \log p + \sum_i (1-\delta_i)\log(1-p) \\ &+ \sum_i x_i(1-\delta_i)\log\theta + \Big[\sum_i n_i(1-\delta_i) - \sum_i x_i(1-\delta_i)\Big]\log(1-\theta)\end{aligned} \tag{1.25}$$

Para obter-se o estimador de máxima verossimilhança para θ:

$$\begin{aligned}\frac{\partial \ell_0}{\partial \theta} &= \frac{\sum_i x_i(1-\delta_i)}{\theta} - \frac{\sum_i n_i(1-\delta_i) - \sum_i x_i(1-\delta_i)}{1-\theta} = 0 \\ &\Rightarrow \hat{\theta} = \frac{\sum_i x_i(1-\delta_i)}{\sum_i n_i(1-\delta_i)}.\end{aligned} \tag{1.26}$$

De maneira similar, para encontrar o estimador de p:

$$\frac{\partial \ell_0}{\partial p} = 0 \Rightarrow \hat{p} = \frac{\sum_i \delta_i}{m}. \tag{1.27}$$

Sob H_1, são obtidos os seguintes estimadores:

$$\hat{\theta_z} = \frac{\sum_{i\in z} x_i(1-\delta_i)}{\sum_{i\in z} n_i(1-\delta_i)}, \tag{1.28}$$

e

$$\hat{\theta_0} = \frac{\sum_{i\notin z} x_i(1-\delta_i)}{\sum_{i\notin z} n_i(1-\delta_i)}. \tag{1.29}$$

Dessa maneira, a razão de verossimilhança para o modelo ZIB é definida como:

$$\begin{aligned}\lambda = &\theta_z^{\sum_{i\in z} x_i(1-\delta_i)}(1-\theta_z)^{\sum_{i\in z} n_i(1-\delta_i)-\sum_{i\in z} x_i(1-\delta_i)} \\ &\times \theta_0^{\sum_{i\notin z} x_i(1-\delta_i)}(1-\theta_0)^{\sum_{i\notin z} n_i(1-\delta_i)-\sum_{i\notin z} x_i(1-\delta_i)} \\ &\times \Big[\theta^{\sum_i x_i(1-\delta_i)}(1-\theta)^{\sum_i n_i(1-\delta_i)-\sum_i x_i(1-\delta_i)}\Big]^{-1}.\end{aligned} \tag{1.30}$$

1.3.1.2 Algoritmo EM para δ_i desconhecidos

Cançado et al. (2014) apresentam uma solução para os problemas em que δ_i, $i = 1, ..., n$, são desconhecidos. É sugerido um procedimento de estimação via algoritmo EM, gerando a estatística Scan ZIP-EM, no caso Poisson. O algoritmo EM, segundo Dempster et al. (1977), é um método iterativo para computo de estimadores de máxima verossimilhança quando as observações podem ser interpretadas como dados incompletos.

Essa abordagem pode ser utilizada de maneira similar, também, para o caso no modelo ZIB, resultando na estatística Scan-ZIB-EM. O procedimento é repetido para cada candidato a cluster, isto é, para cada zona z é necessário estimar os δ_i, conforme explicitado a seguir:

- Passo E: estima-se δ_i pela esperança condicional de δ_i dado X_i. Como $(\delta_i|X_i, p, \theta) \sim Bernoulli(\zeta_i)$,

$$\begin{aligned}
\zeta_i &= E(\delta_i|X_i) = P(\delta_i = 1|X_i) \\
&= \frac{P(X_i = x_i|\delta_i = 1)P(\delta_i = 1)}{P(X_i = x_i|\delta_i = 1)P(\delta_i = 1) + P(X_i = x_i|\delta_i = 0)P(\delta_i = 0)} I(x_i = 0) \\
&= \frac{p}{p + (1-p)\binom{n_i}{x_i}\theta_i^{x_i}(1-\theta_i)^{n_i - x_i}} I(x_i = 0) \qquad (1.31) \\
&= \begin{cases} \frac{p}{p+(1-p)(1-\theta_i)^{n_i}} & , \text{se } x_i = 0 \\ 0 & , \text{se } x_i = 1, 2... \end{cases}
\end{aligned}$$

  Na m-ésima iteração do algoritmo EM,

$$\delta_i{}^{(m)} = \frac{\hat{p}^{(m-1)}}{\hat{p}^{(m-1)} + (1 - \hat{p}^{(m-1)})\left(1 - \hat{\theta}_i{}^{(m-1)}\right)^{n_i}}. \qquad (1.32)$$

- Passo M: Dado o vetor $\hat{\delta}^{(m)} = \left(\hat{\delta}_1^{(m)}, ..., \hat{\delta}_m^{(m)}\right)$, na iteração $m+1$, os estimadores de p, θ_z e θ_0 são obtidos de acordo com (1.27), (1.28) e (1.29), utilizando os $\hat{\delta}_i$ estimados no passo anterior. Logo, sob H_1:

$$\hat{\delta}_i{}^{(m+1)} = \begin{cases} \frac{\hat{p}^{(m)}}{\hat{p}^{(m)} + (1-\hat{p}^{(m)})\left(1-\hat{\theta_z}^{(m)}\right)^{n_i}} & , \text{se } x_i = 0 \text{ e } i \in z \\ \frac{\hat{p}^{(m)}}{\hat{p}^{(m)} + (1-\hat{p}^{(m)})\left(1-\hat{\theta_0}^{(m)}\right)^{n_i}} & , \text{se } x_i = 0 \text{ e } i \notin z \\ 0 & , \text{se } x_i = 1, 2, ... \end{cases} \qquad (1.33)$$

Os passos E e M são repetidos até a convergência. Para a inicialização do algoritmo EM, pode-se utilizar, dentre outras alternativas:

$$\delta_i^0 = \begin{cases} 0.5 & \text{, se } x_i = 0 \\ 0 & \text{, se } x_i > 0. \end{cases} \tag{1.34}$$

Sendo assim, o algoritmo irá estimar δ_i tanto para as regiões dentro da zona z quanto para fora dela.

1.3.2 Encontrando o cluster mais verossímil

De maneira similar ao modelo Binomial, tem-se, para uma zona z:

$$\begin{aligned} \lambda_{ZIB} &= \frac{\sup_{z\in\mathcal{Z},\theta_z>\theta_0} \mathcal{L}(\mathbf{p},\mathbf{z},\mathbf{x},\theta_0,\theta_z)}{\sup_{z\in\mathcal{Z},\theta_z=\theta_0} \mathcal{L}(\mathbf{p},\mathbf{z},\mathbf{x},\theta_0,\theta_z)} \\ &= \sup_{z\in\mathcal{Z}} \frac{\left[\frac{\sum_{i\in z} x_i(1-\delta_i)}{\sum_{i\in z} n_i(1-\delta_i)}\right]^{\sum_{i\in z} x_i(1-\delta_i)} \left[\frac{\sum_{i\notin z} x_i(1-\delta_i)}{\sum_{i\notin z} n_i(1-\delta_i)}\right]^{\sum_{i\notin z} x_i(1-\delta_i)}}{\left[\frac{\sum_i x_i(1-\delta_i)}{\sum_i n_i(1-\delta_i)}\right]^{\sum_i x_i(1-\delta_i)}} \\ &\times I\left(\frac{\sum_{i\in z} x_i(1-\delta_i)}{\sum_{i\in z} n_i(1-\delta_i)} > \frac{\sum_{i\notin z} x_i(1-\delta_i)}{\sum_{i\notin z} n_i(1-\delta_i)}\right), \end{aligned} \tag{1.35}$$

se existir pelo menos uma zona z de tal forma que $\frac{\sum_{i\in z} x_i(1-\delta_i)}{\sum_{i\in z} n_i(1-\delta_i)} > \frac{\sum_{i\notin z} x_i(1-\delta_i)}{\sum_{i\notin z} n_i(1-\delta_i)}$, e $\lambda_{ZIB} = 1$, caso contrário.

A estatística Scan-ZIB-EM (λ_{ZIB-EM}) é calculada da mesma maneira, utilizando os δ_i estimados pelo algoritmo EM (1.3.1.2).

Assim como no caso Binomial, é necessário um algoritmo para buscar os candidatos a clusters. Os procedimentos são idênticos aos da seção (1.2.3.2), utilizando as estatísticas λ_{ZIB} e λ_{ZIB-EM}.

1.3.3 Verificação da significância do Cluster

As estatísticas Scan-ZIB e Scan-ZIB-EM não possuem distribuições conhecidas. Dessa maneira, são necessárias simulações de Monte Carlo a fim de se verificar a significância dos clusters detectados.

Para o caso em que δ_i são conhecidos, pode-se simplesmente remover as regiões com zero estrutural ($\delta_i = 1$) e proceder da mesma forma que em (1.2.4), utilizando a estatística Scan-Binomial tradicional.

Já para casos onde não se conhece os δ_i, de acordo com Cançado et al. (2014), como não se sabe o número de regiões com zero estrutural e nem suas localizações, as réplicas de Monte Carlo são construídas a partir de uma técnica similar ao *bootstrap* paramétrico.

Considere $\hat{p} = \frac{\sum_i \hat{\delta}_i}{m}$ a estimativa da probabilidade de zero estrutural obtida para os dados reais, onde $\hat{\delta}_i$ é um elemento do vetor $\hat{\delta}$ estimado via algoritmo EM para o cluster mais verossímil. A simulação é feita da seguinte maneira:

1. Atribua, com probabilidade $\hat{p}$, zero estrutural para cada região do mapa.
2. Distribuia de forma aleatória os $C = \sum_i x_i$ casos nas regiões que não tiveram zero estrutural atribuído no passo anterior.
3. Encontre o cluster mais verossímil utilizando a estatística Scan-ZIB para esse banco de dados.
4. Repita os passos 1 a 3 um número J de vezes e construa a distribuição empírica de λ_{ZIB-EM}.
5. Rejeite, com nível de significância de 5%, a hipótese H_0 de ausência de clusters se $\lambda_{ZIB-EM} > P_{95}$, onde P_{95} é o 95º percentil da distribuição empírica de λ_{ZIB-EM}.

1.4 Abordagem Bayesiana

O trabalho de Neill et al. (2006) apresenta uma abordagem Bayesiana para a estatística Scan tradicional. Os autores apresentam três desvantagens do modelo frequentista:

> *"Primeiramente, é difícil utilizar qualquer informação a priori que podemos ter, como, por exemplo, a opinião sobre a gravidade de um possível surto e o seu impacto na taxa da doença. A acurácia dessa técnica é altamente dependente da exatidão das estimativas de máxima verossimilhança dos parâmetros. Como resultado, o modelo está propenso ao sobreajuste e pode perder poder de detecção por má especificação do modelo. Finalmente, o cômputo da estatística scan em sua abordagem frequentista é computacionalmente oneroso e pode se tornar inviável para grandes conjuntos de dados."*

Wakefield e Kim (2013) comentam, também, sobre possíveis dificuldades dos métodos frequentistas para se avaliar a significância de clusters secundários. Por exemplo, em Kulldorff (1997) e Jemal et al. (2002), os *p-valores* para os clusters secundários são calculados a partir da comparação da razão de verossimilhança dos mesmos com as simulações sob H_0 para o cluster principal, enquanto o correto seria realizar simulações sob H_0 para cada cluster.

Zhang et al. (2010) propuseram uma alternativa para melhorar o procedimento de avaliação de significância para múltiplos clusters. Por exemplo, os dados do cluster principal (significativo) são removidos do banco de dados original e repete-se o algoritmo Scan para o banco de dados resultante. Caso encontre um cluster secundário significativo, os dados do mesmo são também removidos e verifica-se a presença de um terceiro cluster. Os procedimentos são repetidos até que não se encontre mais clusters significativos.

Apesar da melhoria na avalição das significâncias, o procedimento apresenta, além do custo computacional intenso, a impossibilidade de se comparar os *p-valores* gerados, já que os mesmos são gerados a partir de amostras de tamanhos diferentes.

Nesse contexto, os métodos bayesianos apresentam a vantagem de gerar probabilidades à *posteriori* para cada candidato a cluster, possibilitando uma interpretação mais direta e facilitando as comparações entre os clusters.

Neill et al. (2006) apresenta uma extensão natural para a estatística Scan em sua versão Poisson, com a distribuição conjugada Gamma-Poisson, utilizada em estudos com dados de contagem para epidemias. Para o caso Binomial, apresentado aqui, a extensão natural é a conjugada Beta-Binomial, segundo Ehlers (2011).

Conforme sugerido no trabalho de Shen e Cooper (2010), a distribuição Beta pode ser utilizada na representação de incertezas ou variações aleatórias de uma taxa ou proporção. Mais especificamente, a distribuição Beta-Binomial é utilizada na descrição de incertezas do parâmetro de probabilidade binomial.

1.4.1 Scan Beta-Binomial

Seja $(x_i|\theta_i) \sim Bin(n_i, \theta_i)$, $\theta \sim Beta(\alpha, \beta)$, $C = \sum_i x_i$ e $N = \sum_i n_i$. Sob H_0, $\theta_i = \theta$ e a distribuição à *posteriori* de θ é dada por:

$$\begin{aligned} P(\theta|X) &= \frac{\theta^{\sum_i x_i}(1-\theta)^{\sum_i n_i - \sum_i x_i} \times \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)}}{\int_0^1 \theta^{\sum_i x_i}(1-\theta)^{\sum_i n_i - \sum_i x_i} \times \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)} d\theta} \\ &= \frac{\theta^{\sum_i x_i+\alpha-1}(1-\theta)^{\sum_i n_i - \sum_i x_i+\beta-1}}{\int_0^1 \theta^{\sum_i x_i+\alpha-1}(1-\theta)^{\sum_i n_i - \sum_i x_i+\beta-1} d\theta} \end{aligned} \tag{1.36}$$

onde $B(\alpha, \beta)$ é definida como a função Beta, tal que $B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$. Logo:

$$(\theta|X) \sim Beta\Big(C + \alpha, N - C + \beta\Big) \tag{1.37}$$

Sob H_1, as *posterioris* são dadas por:

$$(\theta_z|X) \sim Beta\Big(\sum_{i\in z} x_i + \alpha_z; \sum_{i\in z} n_i - \sum_{i\in z} x_i + \beta_z\Big) \tag{1.38}$$

e

$$(\theta_0|X) \sim Beta\Big(\sum_{i\notin z} x_i + \alpha_0; \sum_{i\notin z} n_i - \sum_{i\notin z} x_i + \beta_0\Big). \tag{1.39}$$

A verossimilhança marginal $P(X|H_0)$ possui a forma:

$$P(X|H_0) = \int P(X,\theta|H_0)d\theta = \int P(X|\theta,H_0)P(\theta)d\theta. \tag{1.40}$$

Dessa maneira,

$$\begin{aligned} P(X|H_0) &\propto \int_0^1 \theta^{\sum_i x_i}(1-\theta)^{\sum_i n_i - \sum_i x_i + \beta - 1} \times \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)}d\theta \\ &= \frac{1}{B(\alpha,\beta)}\int_0^1 \theta^{\sum_i x_i + \alpha - 1}(1-\theta)^{\sum_i n_i - \sum_i x_i + \beta - 1}d\theta \\ &= \frac{B\Big(C+\alpha; N-C+\beta\Big)}{B(\alpha,\beta)}. \end{aligned} \tag{1.41}$$

Seja H_z a hipótese alternativa de que cada zona $z \in Z$ é um cluster. A verossimilhança marginal $P(X|H_z)$ obtida para cada zona será:

$$\begin{aligned} P(X|H_z) \propto & \frac{B(\sum_{i\in z} x_i + \alpha_z; \sum_{i\in z} n_i - \sum_{i\in z} x_i + \beta_z)}{B(\alpha_z;\beta_z)} \\ & \times \frac{B(\sum_{i\notin z} x_i + \alpha_0; \sum_{i\notin z} n_i - \sum_{i\notin z} x_i + \beta_0)}{B(\alpha_0;\beta_0)}. \end{aligned} \tag{1.42}$$

1.4.1.1 Algoritmo Scan-Bayesiano

A estatística Scan em sua abordagem Bayesiana pode ser calculada da seguinte maneira:

1. Calcular o escore $P(X|H_z)P(H_z)$ para cada zona Z.
2. Calcular $P(X|H_0)P(H_0)$ e adicionar à soma dos escores calculados anteriormente para todas as zonas. Com isso, é obtida a probabilidade $P(X)$.
3. Obter as probabilidades à *posteriori* $P(H_z|X) = \frac{P(X|H_z)P(H_z)}{P(X)}$ para cada zona.

Pode-se definir:

$$\lambda_{bayes} = \sup_Z P(H_z|X). \tag{1.43}$$

Isto é, o cluster procurado será a zona que maximiza o escore da probabilidade à *posteriori* $P(H_z|X)$. Importante notar que, para o caso Bayesiano, não é necessário a simulação de Monte Carlo para a verificação da significância do Cluster, já que a estatística é uma probabilidade. A obtenção da probabilidade $P(H_z)$ é discutida na seção 1.4.2.

1.4.2 Escolha das *prioris*

De acordo com Neill et al. (2006), a maior dificuldade da abordagem Bayesiana é a estimação dos parâmetros α_z, β_z, α_0 e β_0 em relação a cada zona examinada, a probabilidade à *priori* das zonas $P(H_z)$, além das *prioris* globais α, β e $P(H_0)$.

No caso da distribuição Beta-Binomial, os parâmetros possuem interpretações que auxiliam na indicação das *prioris*. Considere um cenário ideal onde existam dados de um tempo passado j do mesmo mapa de estudo, onde é sabido que não existiam clusters.

Seja x_i^j e n_i^j, respectivamente, o número de casos e a população da região i no tempo j. Pode-se utilizar como *prioris* globais $\alpha = \sum_i x_i^j$, $\beta = \sum_i n_i^j - \sum_i x_i^j$ e, para cada zona, $\alpha_z = \sum_{i \in z} x_i^j$, $\beta_z = \sum_{i \in z} n_i^j - \sum_{i \in z} x_i^j$, $\alpha_0 = \sum_{i \notin z} x_i^j$ e $\beta_0 = \sum_{i \notin z} n_i^j - \sum_{i \notin z} x_i^j$.

Nesse caso, pode-se interpretar o parâmetro α como o número de casos na zona estudada e β como o número de controles (diferença entre população e casos).

É necessário parcimônia na escolha do objeto de estudo devido às limitações ligadas ao cômputo da função Beta, que pode apresentar problemas numéricos quando os valores parâmetros α e β são elevados. Dessa forma, a metodologia é mais apropriada para estudos com número reduzido de casos e controles. Recomenda-se, por exemplo, aplicações em estudos de doenças raras. Recomenda-se, ainda, que os controles sejam contagens de casos de outro fenômeno - também raro - mas sabidamente controlado.

Por exemplo, em um estudo para detecção de epidemia de uma nova doença, x seria o número de casos da doença de interesse e n o número de casos de outra doença relacionada, conhecida e controlada. Dessa forma, os valores dos parâmetros serão reduzidos, evitando possíveis problemas numéricos no cálculo da função Beta.

Para a definição das probabilidades à *priori* $P(H_z)$ e $P(H_0)$, de acordo com Neill et al. (2006), é necessário especificar P_1, a probabilidade de que ocorra um surto em uma zona qualquer do mapa. No caso de utilização de *prioris* não informativas, para um dado tempo j, é assumido que a probabilidade de um surto ocorrer é igual em qualquer zona z. Dessa forma, pode-se assumir $P(H_0) = 1 - P_1$ e $P(H_z) = P_1/N_z$, onde N_z é o número de zonas candidatas a cluster. Nota-se que esta escolha assume $P(H_z)$ igual para qualquer zona $z \in Z$.

A probabilidade P_1 pode ser de conhecimento de especialistas, obtida de dados históricos ou ainda utilizada para calibrar a sensibilidade do algoritmo. O modelo também poderia utilizar probabilidade não uniforme. Assim como P_1, as probabilidades $P(H_z)$ também podem ser obtidas de especialistas ou de dados históricos, inclusive podendo ser diferentes para cada zona $z \in Z$.

Nos casos em que não se tem dados de cenários passados, pode-se utilizar *prioris* não informativas como, por exemplo, $\alpha = 1$ e $\beta = 1$. Essa especificação, conhecida por

Bayes-Laplace, foi apresentada em Berger (1985) como uma das quatro melhores opções para casos em que não se tem informações à *priori*. O artigo de Tuyl et al. (2009) apresenta um profundo estudo em que é mostrado que as distribuições preditivas a *posteriori* para as distribuições Binomial e Multinomial sugerem o uso de *prioris* uniformes, sendo a Bayes-Laplace a opção natural para representar a ignorância a *priori*.

Na aplicação para detecção de conglomerados, em cada zona os parâmetros seriam proporcionais ao número de casos e controles, isto é, $\alpha_z = \alpha \sum_{i \in z} x_i/C$, $\beta_z = \beta \sum_{i \in z} n_i/N$, $\alpha_0 = \alpha \sum_{i \notin z} n_i/C$ e $\beta_0 = \beta \sum_{i \notin z} n_i/N$.

No caso específico da especificação de Bayes-Laplace, pode-se simplificar as expressões: $\alpha_z = \sum_{i \in z} x_i/C$, $\beta_z = \sum_{i \in z} n_i/N$, $\alpha_0 = \sum_{i \notin z} n_i/C$ e $\beta_0 = \sum_{i \notin z} n_i/N$.

2 ZIB Bayesiano

2.1 Introdução

O modelo proposto ZIB Bayesiano é inspirado nos trabalhos apresentados por Neill et al. (2006) e Cançado et al. (2014), de forma a estender a estatística Scan para dados com excesso de zeros utilizando a abordagem bayesiana.

2.2 Metodologia

2.2.1 Dados completos

Como no modelo ZIB, exposto na seção (1.3.1), considere δ_i um indicador de zero estrutural para uma determinada região i, em que $\delta_i \sim Bernoulli(p)$ e p é a probabilidade que cada região com contagem de casos nula seja um zero estrutural. Seja $(X_i|\theta_i, p) \sim ZIB(n_i, \theta_i, p)$ e $\theta_i \sim Beta(\alpha_i, \beta_i)$. Nesse caso, assume-se δ_i conhecido para cada zona $z \in Z$.

Sob H_0, tem-se que $\theta_i = \theta$, $\alpha_i = \alpha$ e $\beta_i = \beta$. Então:

$$\begin{aligned} P(\theta|X) &= \frac{\theta^{\sum_i x_i(1-\delta_i)}(1-\theta)^{\sum_i n_i(1-\delta_i)-\sum_i x_i(1-\delta_i)} \times \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)}}{\int_0^1 \theta^{\sum_i x_i(1-\delta_i)}(1-\theta)^{\sum_i n_i(1-\delta_i)-\sum_i x_i(1-\delta_i)} \times \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)} d\theta} \times \int_p P(\theta, p|X)dp = \\ &= \frac{\theta^{\sum_i x_i(1-\delta_i)+\alpha-1}(1-\theta)^{\sum_i n_i(1-\delta_i)-\sum_i x_i(1-\delta_i)+\beta-1}}{\int_0^1 \theta^{\sum_i x_i(1-\delta_i)+\alpha-1}(1-\theta)^{\sum_i n_i(1-\delta_i)-\sum_i x_i(1-\delta_i)+\beta-1} d\theta}. \end{aligned} \tag{2.1}$$

Logo $(\theta|X) \sim Beta\Big(\sum_i x_i(1-\delta_i) + \alpha; \sum_i (n_i - x_i)(1-\delta_i) + \beta\Big)$.

Sob H_1, as *posterioris* são dadas por:

$$(\theta_z|X) \sim Beta\Big(\sum_{i\in z} x_i(1-\delta_i) + \alpha_z; \sum_{i\in z} (n_i - x_i)(1-\delta_i) + \beta_z\Big) \tag{2.2}$$

e

$$(\theta_0|X) \sim Beta\Big(\sum_{i\notin z} x_i(1-\delta_i) + \alpha_0; \sum_{i\notin z} (n_i - x_i)(1-\delta_i) + \beta_0\Big). \tag{2.3}$$

A verossimilhança marginal $P(X|H_0)$ pode ser obtida como:

$$\begin{aligned} P(X|H_0) &= \int_0^1 \theta^{\sum_i x_i(1-\delta_i)}(1-\theta)^{\sum_i n_i(1-\delta_i)-\sum_i x_i(1-\delta_i)} \times \frac{\theta^{\alpha-1}(1-\theta)^{\beta-i}}{B(\alpha,\beta)}d\theta \\ &= \frac{1}{B(\alpha,\beta)}\int_0^1 \theta^{\sum_i x_i(1-\delta_i)+\alpha-1}(1-\theta)^{\sum_i n_i(1-\delta_i)-\sum_i x_i(1-\delta_i)+\beta-1}d\theta \\ &= \frac{B\Big(\sum_i x_i(1-\delta_i)+\alpha; \sum_i (n_i-x_i)(1-\delta_i)+\beta\Big)}{B(\alpha,\beta)}. \end{aligned} \tag{2.4}$$

De maneira similar, sob H_z, calcula-se para cada zona $z \in Z$:

$$\begin{aligned} P(X|H_z) \propto & \frac{B\Big(\sum_{i\in z} x_i(1-\delta_i)+\alpha_z; \sum_{i\in z}(n_i-x_i)(1-\delta_i)+\beta_z\Big)}{B(\alpha_z,\beta_z)} \\ & \times \frac{B\Big(\sum_{i\notin z} x_i(1-\delta_i)+\alpha_0; \sum_{i\notin z}(n_i-x_i)(1-\delta_i)+\beta_0\Big)}{B(\alpha_0,\beta_0)}. \end{aligned} \tag{2.5}$$

O algoritmo para o cálculo da estatística Scan ZIBB (ZIB Bayesiano) será similar ao apresentado em (1.4.1.1), com $T_{ZIBB} = \sup_Z P(H_z|X)$.

2.2.2 Dados Incompletos

Da mesma forma que na versão frequentista (ZIB-EM), é necessário um método para o caso em que δ_i é uma variável latente (desconhecida). Neste caso, ela deverá ser estimada através de um procedimento Bayesiano, conhecido como Amostrador de Gibbs.

O Amostrador de Gibbs é um dos algoritmos mais simples de MCMC (*Markov Chain Monte Carlo*), introduzido primeiramente no contexto de processamento de imagens por Geman e Geman (1984). O artigo de Tanner e Wong (1987) discutiu o uso do Amostrador de Gibbs para casos de dados faltantes, por meio de "aumento de dados".

Primeiramente, seja a distribuição condicional completa à *posteriori* de δ_i definida como $P(\delta_i|\theta_0,\theta_1,p,x_i)$. Considere também $\delta_i \sim Bernoulli(p)$, a distribuição à *priori* de δ_i. Condicionalmente a x_i, θ_i e p, tem-se que $(\delta_i|p,x_i,\theta_0,\theta_z) \sim Bernoulli(\psi_i)$, onde:

$$\begin{aligned} \psi_i = P(\delta_i = 1|p,x_i,\theta_i) &= \frac{P(x_i|\delta_i=1)P(\delta_i=1)}{P(x_i|\delta_i=1)P(\delta_i=1)+P(x_i|\delta_i=0)P(\delta_i=0)} \\ &= \begin{cases} \frac{P(x_i|\delta_i=1)P(\delta_i=1)}{P(x_i|\delta_i=1)P(\delta_i=1)+P(x_i|\delta_i=0)P(\delta_i=0)} & , x_i = 0; \\ 0 & , x_i = 1,2,... \end{cases} \end{aligned} \tag{2.6}$$

Logo:

$$\psi_i = \begin{cases} \frac{p}{p+(1-\theta_z)^{n_i}(1-p)} & , x_i = 0 \text{ e } i \in z; \\ \frac{p}{p+(1-\theta_0)^{n_i}(1-p)} & , x_i = 0 \text{ e } i \notin z; \\ 0 & , x_i = 1, 2, ... \end{cases} \tag{2.7}$$

Estima-se δ_i pela média à *posteriori*, isto é:

$$\hat{\delta}_i = E(\delta_i | p, x_i, \theta_0, \theta_1) = \hat{\psi}_i \tag{2.8}$$

Para se iniciar o Amostrador de Gibbs, é necessário obter, ainda, as condicionais completas de θ_0, θ_z e p. No caso de p, já que $\delta_i \sim Bernoulli(p)$:

$$(p|\theta_0, \theta_z, \boldsymbol{\delta}, \mathbf{x}) \sim Beta\Big(\alpha_p + \sum_i \delta_i, \beta_p + \sum_i (1 - \delta_i)\Big) \tag{2.9}$$

Já para θ_0 e θ_z, de maneira similar ao caso com dados completos:

$$(\theta_0|\theta_z, \boldsymbol{\delta}, p, \mathbf{x}) \sim Beta\Big(\sum_{i \notin z} x_i(1 - \delta_i) + \alpha_0; \sum_{i \notin z} (n_i - x_i)(1 - \delta_i) + \beta_0\Big) \tag{2.10}$$

e

$$(\theta_z|\theta_0, \boldsymbol{\delta}, p, \mathbf{x}) \sim Beta\Big(\sum_{i \in z} x_i(1 - \delta_i) + \alpha_z; \sum_{i \in z} (n_i - x_i)(1 - \delta_i) + \beta_z\Big) \tag{2.11}$$

Da mesma forma que no Algoritmo EM utilizado para estimar os δ_i no caso frequentista (1.3.1.2), o algoritmo do Amostrador de Gibbs deve ser repetido para cada candidato a cluster, isto é, para cada zona z é necessário estimar os δ_i.

2.2.2.1 Amostrador de Gibbs para δ_i desconhecidos

Seja w o número de iterações de "aquecimento" e m o número de iterações que serão de fato utilizadas para se estimar os δ_i. Para cada zona z candidata a cluster, o Amostrador de Gibbs segue os seguintes passos:

1. Iniciar o vetor $\boldsymbol{\delta}$, onde:

$$\delta_i^0 = \begin{cases} 0.5 & , \text{se } x_i = 0 \\ 0 & , \text{se } x_i > 0 \end{cases} \tag{2.12}$$

2. Para cada iteração $j = 1, ..., w, ..., w + m$:

a) Gere valores de $(p|\theta_0, \theta_z, \boldsymbol{\delta}^{j-1}, \mathbf{x})$, $(\theta_0|\theta_z, \boldsymbol{\delta}^{j-1}, p, \mathbf{x})$ e $(\theta_z|\theta_0, \boldsymbol{\delta}^{j-1}, p, \mathbf{x})$ com base em nas distribuições completas definidas em (2.9), (2.10) e (2.11), respectivamente, onde δ^{j-1} é o vetor δ calculado na iteração anterior.

b) Calcular o vetor δ^j, definindo-se:

$$\delta_i^j = \begin{cases} \frac{p^j}{p^j+(1-\theta_z^j)^{n_i}(1-p^j)} & , x_i = 0 \text{ e } i \in z; \\ \frac{p^j}{p^j+(1-\theta_0^j)^{n_i}(1-p^j)} & , x_i = 0 \text{ e } i \notin z; \\ 0 & , x_i = 1, 2, ... \end{cases} \tag{2.13}$$

onde δ_i^j, θ_z^j, θ_0^j e p^j são, respectivamente, os valores obtidos de δ_i, θ_z, θ_0 e p na j-ésima iteração.

3. Desconsidere as primeiras w amostras de δ_i, θ_z, θ_0 e p, restando, ainda, m amostras de cada variável.

4. Tome as médias amostrais de cada variável. No caso do vetor estimado $\hat{\delta}$, os valores de $\hat{\delta}_i$ serão a média dos δ_i^j, considerando apenas as m amostras finais.

Esse procedimento gera a estatística Scan-ZIBB-Gibbs. O algoritmo para seu cálculo será idêntico ao Scan-ZIBB, com os $\hat{\delta}_i$ estimados pelo Amostrador de Gibbs.

Parte II

Resultados

3 Resultados

3.1 Implementação computacional

A fim de comparar os métodos apresentados neste trabalho, os mesmos foram implementados em linguagem R, utilizando técnicas computacionais descritas em Chambers (2010). Considere novamente $N = \sum_{i=1}^{m} n_i$ e $C = \sum_{i=1}^{m} x_i$ como, respectivamente, a população total e o número total de caso no mapa estudado. Para cada método, o algoritmo foi implementado da seguinte forma:

1. Cálculo da matriz de distâncias, conforme (1.2.3.1), e construção das zonas com base na distância entre os centroides das regiões do mapa.

2. Identificação das zonas candidatas a cluster. Serão consideradas candidatas aquelas zonas para os quais $\sum_{i \in z} x_i > \frac{\sum_{i \in z} n_i C}{N}$ e $\sum_{i \in z} n_i < n_{max}$, isto é, o número observado de casos é maior que o esperado e a população da zona é menor que um valor previamente estipulado. Nesse caso, foi utilizado $n_{max} = 0.25N$.

Exemplo: Algoritmo Scan ZIBB - Passos 1 e 2

```
a<-cbind(pop,cases,x,y)
  coord<-cbind(x,y)
  n<-nrow(a)
  alpha<-1
  beta<-1

  N=sum(a[,1]) #Total population
  C=sum(a[,2]) #Total number of cases
  cmax=n
  nmax=0.25*N  #Maximum population allowed inside a cluster

  #Distance Matrix
  dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
  ss<-seq(1,n,1)
  d<-cbind(dist,ss)

  R<-matrix(NA,ncol=n,nrow=n)
  P<-matrix(NA,ncol=n,nrow=n)
  ex_matrix<-matrix(NA,ncol=n,nrow=n)
  matrix_a<-matrix(NA,ncol=n,nrow=n)
  for(j in 1:n){
    d<-d[order(d[,j]),]
    R[,j]<-d[,n+1]
    for (i in 1:n) {
      P[i,j]<-sum(a[R[1:i,j],1])
      ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
      if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
        matrix_a[i,j]<-1}
        else {  matrix_a[i,j]<-0}
```

```
          }
      } regions<-which(matrix_a==1,arr.ind=TRUE)
```

3. Cálculo da estatística razão de verossimilhança (para os métodos frequentistas Scan-Binomial, Scan-ZIB e Scan-ZIB-EM) ou a probabilidade à posteriori (para os métodos bayesianos Scan-Beta-Binomial, Scan-ZIBB e Scan-ZIBB-Gibbs), para as zonas candidatas a cluster.

4. Assumir a zona com maior valor da razão de verossimilhança ou probabilidade à posteriori como o cluster detectado.

Exemplo: Algoritmo Scan ZIBB - Passos 3 e 4

```
ad<-matrix(NA,ncol=2,nrow=n)
ad[,1]<-a[,1]*(1-delta)
ad[,2]<-a[,2]*(1-delta)
    nz=0
    xz=0
    llr_matrix<-matrix(0,ncol=n,nrow=n)

    xhz_matrix<-llr_matrix
        bfac<-matrix(NA,ncol=n,nrow=n)
    theta<-C/N
    p_1<-0.5
    ph0<-1-p_1
    phz<-p_1/nreg

    pxh0<-beta(sum(ad[,2])+alpha, sum(ad[,1])-sum(ad[,2])+beta)/
    beta(alpha,beta)

for(ii in 1:nrow(regions)){
  i<-regions[ii,1]
  j<-regions[ii,2]
  iz<-R[1:i,j]  #index inside
  oz<-R[-(1:i),j] #index outside

  az<-alpha*sum(a[iz,2])/C
  bz<-beta*(sum(a[iz,1])-sum(a[iz,2]))/(N-C)
  ao<-alpha*sum(a[oz,2])/C
  bo<-beta*(sum(a[oz,1])-sum(a[oz,2]))/(N-C)

  pxhz<-(beta(sum(ad[iz,2])+az, sum(ad[iz,1])-
  sum(ad[iz,2])+bz)/beta(az,bz))*
  (beta(sum(ad[oz,2])+ao, sum(ad[oz,1])-sum(ad[oz,2])+bo)/beta(ao,bo))
  bfac[i,j]<-pxhz/pxh0
xhz_matrix[i,j]<-pxhz*phz

}
                p_d<-sum(xhz_matrix,na.rm=TRUE)+pxh0*ph0
                llr_matrix<-xhz_matrix/p_d

        t<-max(na.omit(llr_matrix)) #Maximum Posterior Probability Value
        max<-which(llr_matrix==t, arr.ind=TRUE)
        bf<-bfac[max[1,1],max[1,2]] #Bayes Factor
        clustera<-sort(R[1:max[1,1],max[1,2]]) #Detected cluster
        ncluster<-sum(a[clustera,1]) #Population inside detected cluster
        xcluster<-sum(a[clustera,2]) #Observed Cases inside detected cluster
```

```
expected_cases<-ncluster*C/N #Expected Cases inside detected cluster
```

5. Verificação da significância do cluster encontrado, no caso dos métodos frequentistas.

Os métodos bayesianos automaticamente assumem *prioris* não informativas ($\alpha = \beta = 1$) caso as mesmas não sejam informadas.

Exemplo: Algoritmo Scan ZIBB - Verificação das Prioris

```
scan_zibb<-function(pop,cases,x,y,delta,alpha=NULL,beta=NULL){
  if(is.null(alpha) & is.null(beta)) {
  ...
  }
  else {...}
```

Para os métodos frequentistas, essa forma de implementação permite a fácil obtenção de clusters secundários, caso haja interesse. Já para os métodos Bayesianos, permite a confecção de mapas com escala de probabilidades. Os códigos implementados estão no Anexo A deste trabalho.

3.2 Cenários

Foram gerados quatro cenários artificiais (A, B, C e D) para a comparação dos métodos descritos neste trabalho. O mapa hipotético contém 121 regiões em formato de quadrados. Os cenários foram definidos de forma similar aos que foram descritos por Cançado et al. (2014), de modo a comparar os métodos em diferentes situações de tamanho, formato e número de regiões com zero estrutural dentro dos clusters.

Cada região possui $n_i = 15$. Nesse caso, interpreta-se n_i como um número de controles, podendo ser, por exemplo, o número de casos de uma doença conhecida e controlada. Sendo assim, $N = \sum_i n_i = 1815$, para todos os cenários.

Das 121 regiões, 11 foram selecionadas para terem zero estrutural, também fixadas para os quatro cenários. Ao encontro da proposta deste trabalho de estudar o comportamento dos métodos para problemas envolvendo um baixo número de casos, foi utilizado $C = \sum_i x_i = 170$, isto é, aproximadamente 9% do número de controles serão considerados como casos.

O cenário A possui um cluster em formato de um quadrado com três regiões em cada lado, abrangendo assim 9 regiões. As regiões em sua diagonal possuem zero estrutural. Elas funcionam como um divisor do cluster, isto é, se removidas separam o cluster em zonas desconexas. Esse cenário pode ser visualizado na figura 2.

O cenário B possui um cluster em formato de retângulo de tamanho 4×3, abrangendo 12 regiões. Três destas regiões possuem zero estrutural, porém não possuem um

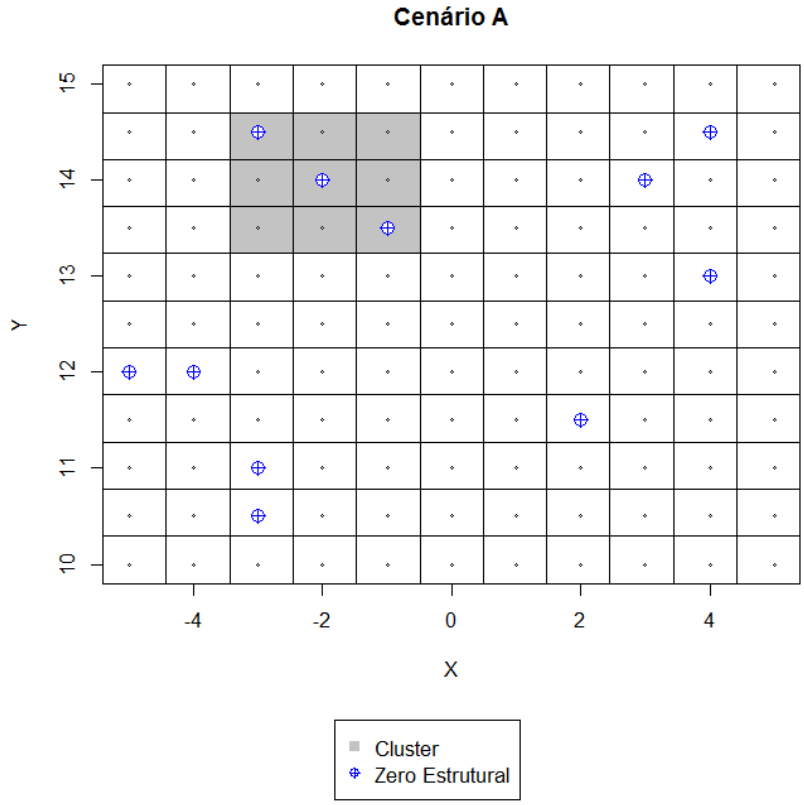


Figura 2: Localização do cluster no cenario A

padrão de distribuição espacial bem definido como no cenário A. Pode-se visualizar essa situação na figura 3.

O cenário C se caracteriza por possuir um cluster em formato de cruz, com 5 regiões. Apenas uma região, localizada no centro do cluster, possui zero estrutural, conforme visualizado na figura 4.

Por fim, no cenario D (figura 5), o cluster possui um formato de "L" invertido. Das 14 regiões do cluster, 4 possuem zero estrutural.

3.2.1 Simulações

Para cada cenário, foram avaliadas 1000 simulações em que os 170 casos foram distribuídos aleatoriamente utilizando uma distribuição multinomial com probabilidades proporcionais aos riscos relativos de cada região. Os riscos relativos são calculados de acordo com Kulldorff et al. (2003), de maneira que as regiões dentro do cluster possuem alto risco relativo, enquanto nas regiões fora dele o risco será baixo. Os riscos relativos para as regiões dentro do cluster devem ser altos o suficiente para que um teste Binomial simples rejeite com probabilidade de 99.9% a hipótese nula de ausência de cluster.

Seja n_z o número de controles dentro do cluster, N o total de controles em todas as regiões, C o total de casos e H_0 a hipótese nula de ausência de cluster. Condicional

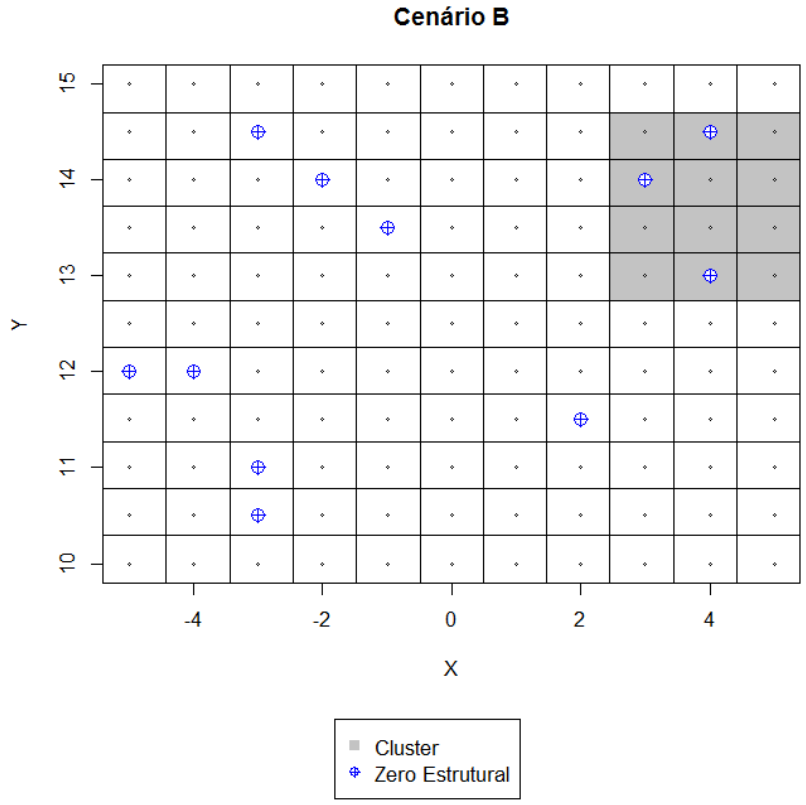


Figura 3: Localização do cluster no cenario B

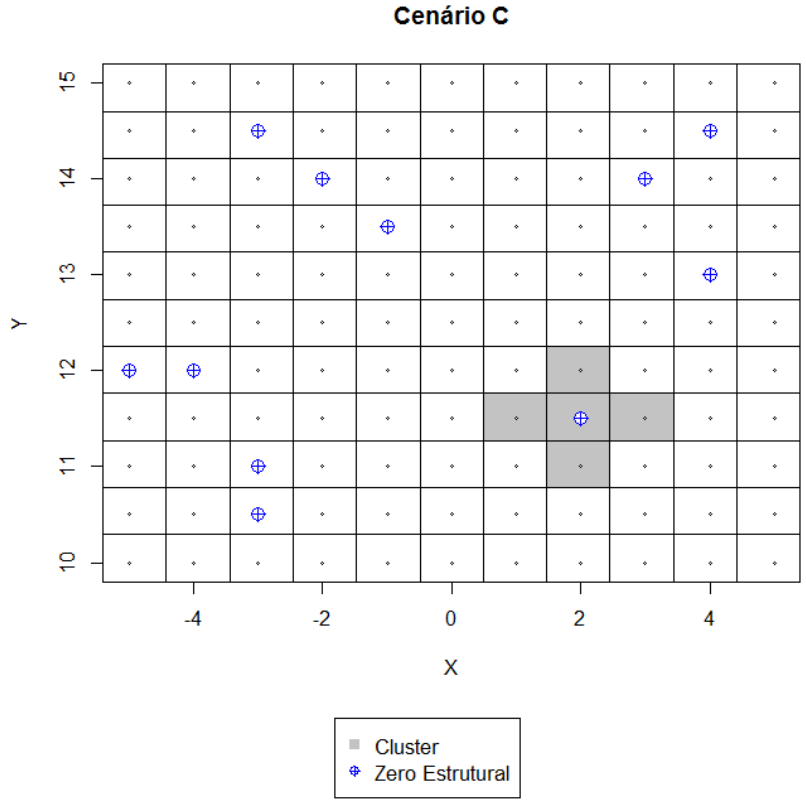


Figura 4: Localização do cluster no cenario C

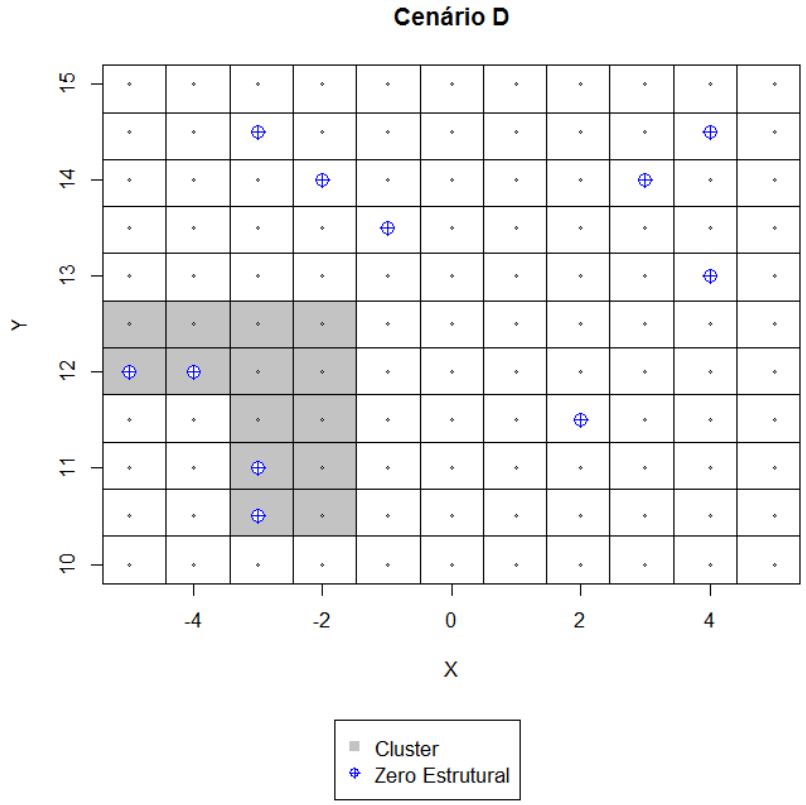


Figura 5: Localização do cluster no cenario D

a C, o número de casos sob H_0 para as regiões dentro do cluster segue uma distribuição Binomial com média $m_0 = Cn_z/N$ e variância $v_0 = m_0(N - n_z)/N$. Aproximando a distribuição Binomial pela Normal, tem-se que o número de casos k necessário para que o teste unilateral rejeite a hipótese nula com 5% de significância é tal que $(k - m_0)/\sqrt{v_0} = 1.645$.

Sob a hipótese alternativa, as regiões dentro do cluster possuem risco relativo r e os casos tem distribuição Binomial com média $m_1 = (Cn_z r)/(N - n_z + n_z r)$ e variância $v_1 = m_1(N - n_z)/(N - n_z + n_z r)$. Sendo assim, aproximando novamente pela normal, calcula-se o risco relativo r de tal forma que $(k - m_1)/\sqrt{v_1} = 3.09$. Essa escolha é feita para que a hipótese nula seja rejeitada com probabilidade de 99.9% quando realizado um teste Binomial simples. Para as regiões fora do cluster, o risco relativo é fixado em $r = 1$. Os riscos relativos para as regiões dentro do cluster (RR), número esperado de casos dentro do cluster para a hipótese nula (m_0) e alternativa (m_1) dos quatro cenários estão na tabela 4.

De acordo com Cançado et al. (2014), uma região com zero estrutural dentro do cluster pode ser interpretada como uma área de alto risco em que a contagem de casos não está disponível. Neste caso, apesar do alto risco relativo, a distribuição de casos dentro do cluster será realizada somente nas regiões que não possuem zero estrutural. Um caso

Tabela 1: Riscos relativos dentro do cluster, número de casos esperados sob H_0 (m_0) e número de casos esperados sob H_1 (m_1)

Cenario	RR	m_0	m_1
A	3.1653	12.645	34.472
B	2.8336	16.860	40.423
C	4.1385	7.0248	25.735
D	2.6856	19.669	44.203

atribuído para uma região com zero estrutural será rejeitado e outro caso será gerado. Este procedimento garante que o número de casos em regiões com zero estrutural seja sempre nulo.

3.2.2 Análise de Desempenho

Para efeitos comparativos, lança-se mão de uma série de medidas de desempenho para os métodos apresentados no trabalho. Para os métodos frequentistas (Scan-Binomial, Scan-ZIB e Scan-ZIB+EM), em cada uma das M simulações calcula-se a estatística de teste λ correspondente e compara-se a mesma com um valor crítico λ^*, obtido através das simulações de Monte Carlo, sob H_0, conforme visto em (1.2.4) e (1.3.3). Define-se então o poder como

$$Poder = \frac{\sum_{i=1}^{M} I(\lambda > \lambda^*)}{M} \tag{3.1}$$

onde $I(\lambda_i > \lambda^*) = 1$ caso $\lambda > \lambda^*$ e 0 caso contrário. Utilizou-se também as medidas de Sensibilidade e Valor Preditivo Positivo (VPP), conforme proposto por Kulldorff et al. (2009). Tais medidas são baseadas na comparação das populações (ou controles) do cluster detectado e do cluster verdadeiro.

$$Sensibilidade = \frac{Pop(ClusterDetectado \cap ClusterVerdadeiro)}{Pop(ClusterVerdadeiro)} \tag{3.2}$$

$$VPP = \frac{Pop(ClusterDetectado \cap ClusterVerdadeiro)}{Pop(ClusterDetectado)} \tag{3.3}$$

A Sensibilidade pode ser interpretada como o quanto do cluster verdadeiro é detectado, enquanto o VPP representa o quanto do cluster detectado pertence ao verdadeiro. Dessa forma, em casos de estudos envolvendo detecção de clusters de doenças, por exemplo, a utilização de um método com maior Sensibilidade é benéfico.

No caso dos métodos Bayesianos (Scan-Beta-Binomial, Scan-ZIBB e Scan-ZIBB-Gibbs), não é possível calcular o "Poder". Uma alternativa é utilizar o Fator de Bayes,

definido como a razão entre as probabilidades à posteriori da hipótese alternativa e da hipótese nula. Isto é:

$$BF = \frac{P(X|H_z)}{P(X|H_0)} \tag{3.4}$$

No caso deste estudo, o Fator de Bayes indica o quão mais provável que uma zona seja um cluster (H_z) em relação à ausência de clusters (H_0). Para efeitos de comparação, considerou-se somente o cluster mais provável, calculando o Fator de Bayes como:

$$BF = \frac{\sup_Z P(X|H_z)}{P(X|H_0)} \tag{3.5}$$

Jeffreys (1967) sugere que o Fator de Bayes seja interpretado na escala de $\log_{10}$. Fazendo a aproximação sugerida por Kass e Raftery (1995), é obtida uma escala de interpretação, conforme a tabela 2. Apesar de não haver consenso, para efeitos comparativos foi assumido que o cluster mais provável é "significativo" caso $BF > 100$.

Tabela 2: Fator de Bayes - Interpretação

$\log_{10}(BF)$	BF	Evidência contra H_0
0 a 1/2	1 a 3.2	Não significativa
1/2 a 1	3.2 a 10	Positiva
1 a 2	10 a 100	Forte
> 2	> 100	Decisiva

A tabela 3 mostra os resultados para cada método para as 1000 simulações dos quatro cenários estudados. Os valores de Sensibilidade e VPP são as médias das simulações que retornaram clusters significativos (para os métodos frequentistas) ou com $BF > 100$, no caso dos métodos Bayesianos.

Uma rápida análise indica uma deterioração da sensibilidade média para os métodos Scan Binomial e Beta-Binomial à medida que o número de regiões com zero estrutural dentro do cluster cresce.

Por exemplo, no cenário C, onde o cluster verdadeiro apresenta apenas uma região com zero estrutural, a sensibilidade média para o Scan Binomial foi de 0.910 e de 0.829 para o Beta-Binomial. Já no cenário D, com quatro regiões com zero estrutural no cluster, as sensibilidades médias para o Scan Binomial e Beta-Binomial foram respectivamente 0.479 e 0.453.

Em comparação aos métodos Scan ZIB e ZIBB, com zeros estruturais conhecidos, os métodos apresentaram Sensibilidade e VPP médio semelhantes. Porém, o custo computacional do método Bayesiano se destaca por ser, em média, aproximadamente 500 vezes

Tabela 3: Resultados das Simulações

Cenário	Método	Poder	$\%BF > 100$	Sensibilidade	VPP
A	Binomial	0.829	-	0.705	0.644
	ZIB	0.925	-	0.801	0.757
	ZIB-EM	0.917	-	0.805	0.713
	Beta-Binomial	-	0.859	0.606	0.720
	ZIBB	-	0.926	0.853	0.759
	ZIBB-Gibbs	-	0.920	0.827	0.630
B	Binomial	0.891	-	0.640	0.774
	ZIB	0.936	-	0.841	0.829
	ZIB-EM	0.905	-	0.842	0.796
	Beta-Binomial	-	0.887	0.617	0.802
	ZIBB	-	0.937	0.865	0.826
	ZIBB-Gibbs	-	0.879	0.844	0.711
C	Binomial	0.885	-	0.910	0.581
	ZIB	0.912	-	0.826	0.813
	ZIB-EM	0.905	-	0.868	0.694
	Beta-Binomial	-	0.887	0.829	0.652
	ZIBB	-	0.906	0.853	0.797
	ZIBB-Gibbs	-	0.895	0.849	0.508
D	Binomial	0.876	-	0.479	0.793
	ZIB	0.907	-	0.618	0.759
	ZIB-EM	0.858	-	0.621	0.717
	Beta-Binomial	-	0.878	0.453	0.817
	ZIBB	-	0.893	0.651	0.752
	ZIBB-Gibbs	-	0.873	0.715	0.539

mais rápido que o método frequentista, já que não é necessário realizar simulações sob H_0 para verificar a significância do cluster.

Os métodos Scan ZIB-EM e ZIBB-Gibbs, para zeros estruturais desconhecidos, tiveram performance semelhante em termos de Sensibilidade média nos 4 cenários. Em termos de VPP, o desempenho do ZIB-Gibbs ficou um pouco abaixo do ZIB-EM, porém ainda apresentou um resultado razoável.

Existe uma certa dificuldade em se comparar os dois métodos do ponto de vista do custo computacional, já que os dois necessitam de algoritmos iterativos para se estimar os δ_i e o ZIB-EM ainda necessita de simulações para verificar a significância. Entretanto, para um certo valor fixo j, o ZIB-Gibbs (utilizando j amostras de Gibbs em cada candidato a cluster onde foi necessário) foi aproximadamente 6 vezes mais rápido que o ZIB-EM (utilizando 7 iterações do EM onde necessário e j simulações para verificar sua significância).

3.3 Aplicações em Dados Reais

3.3.1 Dados

Para a aplicação da metodologia proposta, foram utilizados dados de óbitos por Febre Hemorrágica do Dengue (FHD) no estado do Rio de Janeiro em 2011. Os dados foram obtidos no Sistema de Informação de Agravos de Notificação - SINAN (`http://dtr2004.saude.gov.br/sinanweb/`).

Assim como na Dengue tradicional, o modo de transmissão da FHD é por meio de picada do mosquito *Aedes Aegypti*. A chance de desenvolvimento da FHD é maior em pacientes que já contrairam a dengue clássica ao menos uma vez. Exames específicos (como isolamento e sorologia) e inespecíficos (como hemograma, hemoconcentração e trombocitopenia) são utilizados para diagnosticar a doença. Segundo Dias et al. (2010):

> *"O extravasamento de plasma é a manifestação mais específica da FHD, já que está presente apenas nessa forma clínica da doença, e é também o que põe em risco a vida do paciente, pois quando ocorre de forma muito intensa pode levar ao choque circulatório, que é de rápida instalação e se não for prontamente tratado pode levar ao óbito em 12-24 horas."*

O estado do Rio de Janeiro possui 92 municípios e apresentou, em 2011, 914 casos de FHD. Destes, 855 pacientes foram curados em 26 municípios. Foram reportados 37 óbitos por FHD em 12 municípios. 21 casos não foram reportados devidamente e foram ignorados.

Nesta aplicação, os pacientes curados foram utilizados como controles e os óbitos como casos. Dessa forma, o estudo buscará um cluster em que a doença é mais letal, podendo indicar, por exemplo, municípios com piores condições dos hospitais e postos de saúde. Nas figuras 6 e 7 é possível vizualizar, respectivamente, os controles e casos distribuídos por faixas de quartis.

3.3.2 Resultados e Discussão

Foram aplicadas as metodologias Scan ZIB-EM, Scan-Beta-Binomial e Scan ZIB-Gibbs. A figura 8 apresenta os clusters detectados nos três métodos.

Para o Scan Beta-Binomial, que não considera o excesso de zeros, o cluster mais provável engloba 21 municípios e possui probabilidade à *posteriori* de 2.52%. O Fator de Bayes obtido para a região foi de 184.68, considerado decisivo de acordo com a tabela 2. O método Scan-ZIB-EM detectou um cluster não significativo de tamanho 44. Já o método Scan-ZIBB-Gibbs, o algoritmo detectou um cluster de tamanho 43, com probabilidade

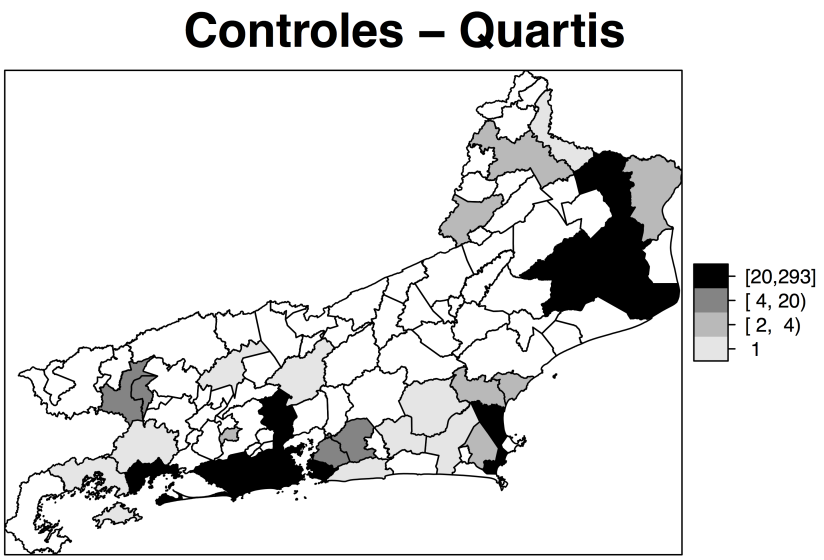


Figura 6: Mapa de quartis de controles de FHD (pacientes curados) - RJ (2011)

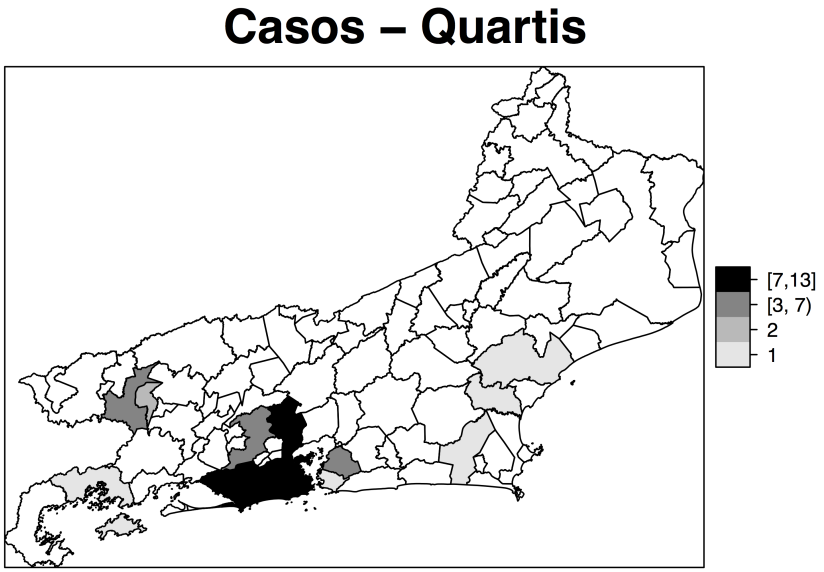


Figura 7: Mapa de quartis de casos de óbitos por FHD - RJ (2011)

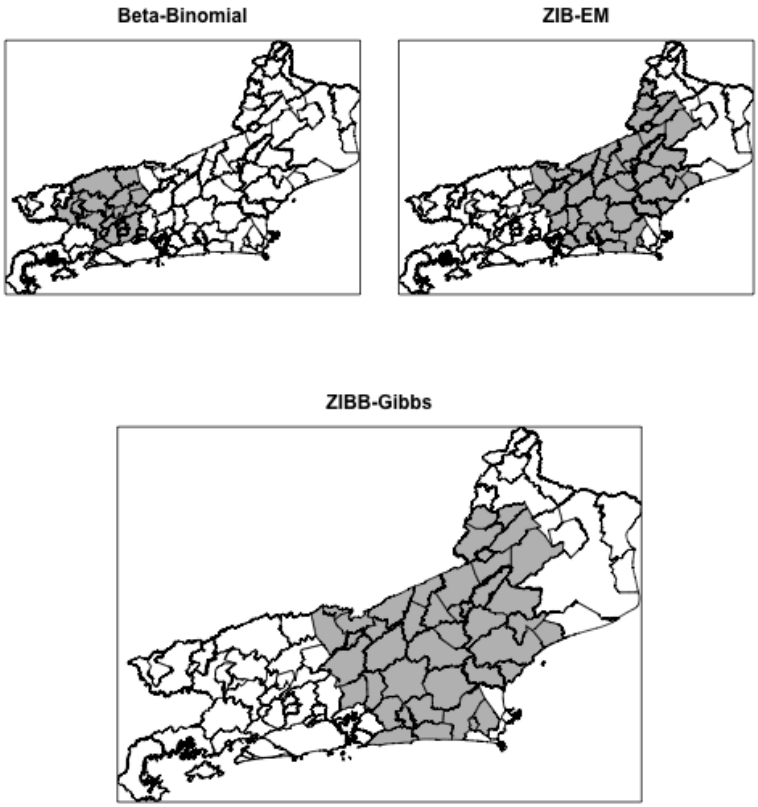


Figura 8: Clusters Detectados

a *posteriori* de 4.8% e Fator de Bayes 9950.676. A tabela 4 apresenta com detalhes as informações dos clusters detectados.

Percebe-se que as regiões presentes nos clusters detectados pelos métodos Scan-ZIB-EM e Scan-ZIBB-Gibbs são semelhantes, o que de certa forma é esperado, já que ambas consideram o excesso de zeros em seus cálculos.

Tabela 4: Resumo para os clusters de óbitos por FHD detectados

Método	Municípios	Controles	Casos Esperados	Casos Observados	Sigficativo	Fator de Bayes
Scan Beta-Binomial	21	32	1.4	8	-	184.68
Scan ZIB-EM	44	24	1.05	7	Não	-
Scan ZIBB-Gibbs	43	20	0.88	3	-	9950.67

Os métodos bayesianos posssibilitam a elaboração de mapas de gradiente que permitem a visualização das regiões presentes nos clusters mais prováveis. Por exemplo, nas figuras 9 e 10, são apresentadas as regiões presentes nos 10 clusters mais prováveis detectados nos métodos Scan Beta-Binomial e Scan ZIBB-Gibbs. Quanto mais escuro o tom de cinza, maior a probabilidade da região pertencer ao cluster.

Pelo fato de um grande número de municípios com zero casos integrarem o cluster

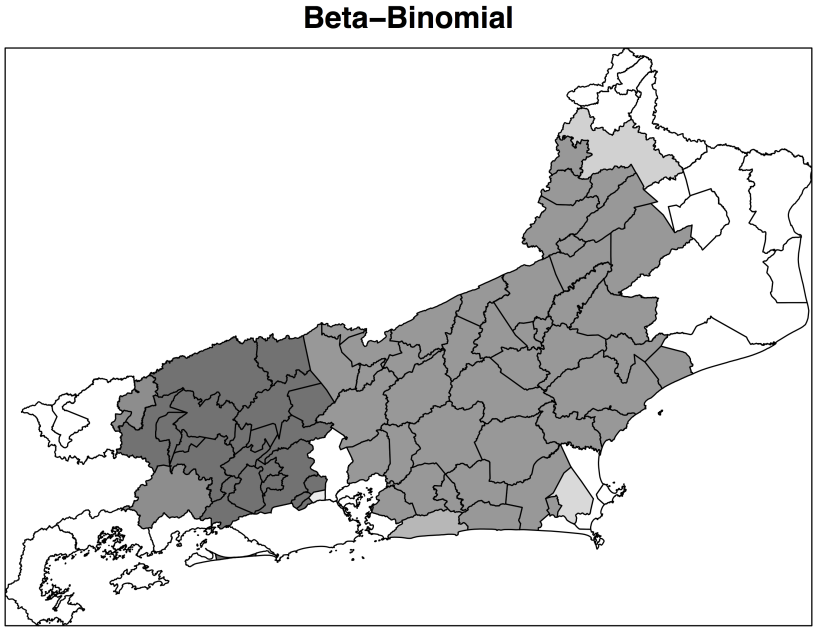


Figura 9: Mapa de probabilidades- Beta-Binomial

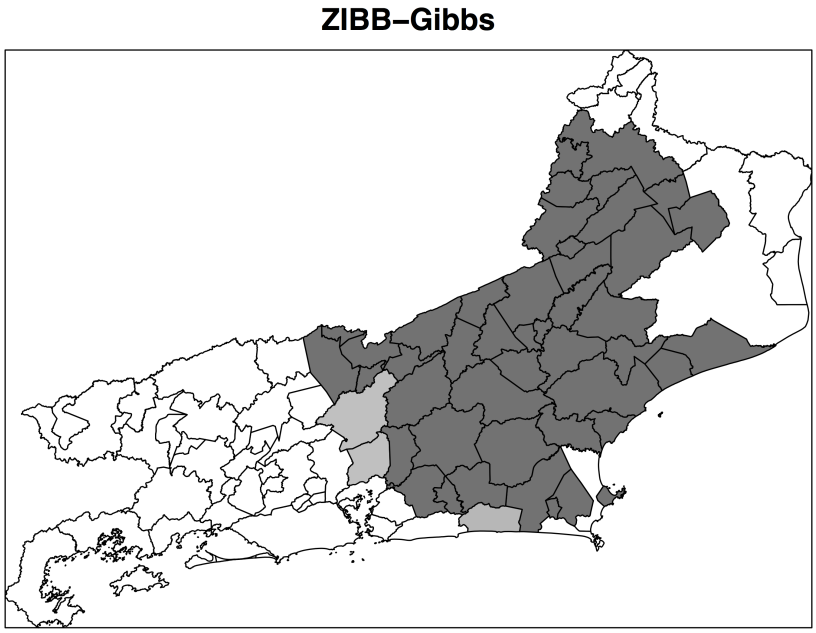


Figura 10: Mapa de probabilidades - ZIBB-Gibbs

mais provável e de não ser possível definir a presença de zeros estruturais, percebe-se que grande parte do estado pode ter apresentado mortes por dengue hemorrágica não reportadas.

4 Considerações finais

Neste trabalho foram propostos os métodos de detecção de conglomerados espaciais Scan ZIBB e Scan ZIBB-Gibbs. Tais métodos são propostos a partir da abordagem bayesiana de métodos frequentistas de detecção para casos com excesso de zeros.

Os métodos para dados com excesso de zeros, como o Scan ZIB, ZIB-EM, ZIBB e ZIBB-Gibbs tem por principal característica considerar a possibilidade da existência de zeros estruturais, isto é, regiões onde é impossível de se obter uma contagem de casos. Os métodos usualmente utilizados, como o Scan Binomial e Beta-Binomail, consideram apenas a presença de zeros amostrais, isto é, regiões com contagem nula de casos.

As simulações realizadas mostram que os métodos que não consideram a possível presença de zeros estruturais apresentam menor poder à medida que o número de regiões com contagem nula dentro do cluster verdadeiro aumenta.

No caso da comparação entre métodos frequentistas e bayesianos, pode-se observar que os métodos Scan ZIBB e Scan ZIBB-Gibbs possuem um custo computacional reduzido em relação aos relativos frequentistas (Scan ZIB e ZIBB-EM). Isso se dá pela ausência da necessidade de se realizar simulações para verificar a significância dos clusters encontrados.

Outro fator de relevância é a possibilidade de se elaborar mapas de gradiente, possibilitando a visualização de regiões com maior probabilidade de serem incluídas no cluster verdadeiro (desconhecido).

4.1 Trabalhos Futuros

Em todas as simulações e aplicações foram utilizadas *prioris* não informativas ($\alpha = \beta = 1$). Apesar dos bons resultados obtidos, um aprofundamento na descrição e especificação das *prioris* pode trazer melhorias na detecção dos clusters.

Uma das principais dificuldades encontradas no estudo foi a limitação numérica da função *Beta*, necessária para se realizar alguns cálculos do Scan ZIBB e ZIBB-Gibbs. Dessa forma, um estudo aprofundado para métodos de aproximações e possíveis alternativas de distribuições pode reduzir essa limitação, ampliando as possibilidades de uso dos métodos propostos. Por exemplo, pode-se utilizar a distribuição *Kumaraswamy* (Kumaraswamy (1980)), porém o desenvolvimento matemático e o estudo das distribuições nos métodos Bayesianos tende a se tornar complexo matematicamente.

Como os métodos apresentados no trabalho estão todos implementados em linguagem *R*, um próximo passo seria a união desses algoritmos em um pacote único e de fácil

utilização para usuários finais.

Referências

AGRESTI, A. *Categorical Data Analysis*. Segunda. [S.l.]: Wiley, 2002. Citado na página 21.

ALA, A.; STANCA, C. M.; BU-GHANIM, M.; AHMADO, I.; BRANCH, A. D.; SCHIANO, T. D.; ODIN, J. A.; BACH, N. Increased prevalence of primary biliary cirrhosis near superfund toxic waste sites. *Hepatology*, Wiley Online Library, v. 43, n. 3, p. 525–531, 2006. Citado na página 12.

BAKKER, M. I.; HATTA, M.; KWENANG, A.; FABER, W. R.; BEERS, S. M. van; KLATSER, P. R.; OSKAM, L. Population survey to determine risk factors for mycobacterium leprae transmission and infection. *International Journal of Epidemiology*, IEA, v. 33, n. 6, p. 1329–1336, 2004. Citado na página 12.

BERGER, J. O. *Statistical decision theory and Bayesian analysis*. [S.l.]: Springer Science & Business Media, 1985. Citado na página 30.

BESAG, J.; NEWELL, J. The detection of clusters in rare diseases. *Journal of the Royal Statistical Society*, v. 154, n. 1, p. 143–155, 1991. Citado na página 12.

BESCULIDES, M.; HEFFERNAN, R.; MOSTASHARI, F.; WEISS, D. Evaluation of school absenteeism data for early outbreak detection, new york city. *BMC public health*, BioMed Central Ltd, v. 5, n. 1, p. 105, 2005. Citado na página 12.

CANÇADO, A. L. F.; SILVA, C. Q. da; SILVA, M. F. da. A spatial scan statistic for zero-inflated poisson process. *Environmental and Ecological Statistics*, Springer US, v. 21, n. 4, p. 627–650, 2014. ISSN 1352-8505. Disponível em: <http://dx.doi.org/10-.1007/s10651-013-0272-1>. Citado 10 vezes nas páginas 5, 6, 12, 15, 22, 24, 26, 31, 38 e 41.

CHAMBERS, J. *Software for Data Analysis: Programming with R*. Springer New York, 2010. (Statistics and Computing). ISBN 9781441926128. Disponível em: <http://books.google.com.br/books?id=5v-ncQAACAAJ>. Citado na página 36.

CUADROS, D. F.; AWAD, S. F.; ABU-RADDAD, L. J. Mapping hiv clustering: a strategy for identifying populations at high risk of hiv infection in sub-saharan africa. 2013. Citado na página 12.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, Wiley for the Royal Statistical Society, v. 39, n. 1, p. pp. 1–38, 1977. ISSN 00359246. Disponível em: <http://www.jstor.org/stable/2984875>. Citado na página 24.

DIAS, L. B.; ALMEIDA, S. C.; HAES, T. M.; MOTA, L. M.; RORIZ-FILHO, J. S. Dengue: transmissão, aspectos clínicos, diagnóstico e tratamento. In: FMRP-USP. *Medicina (Ribeirão Preto)*. [S.l.], 2010. (2, v. 43), p. 143–52. Citado na página 45.

DRUCK, S.; CARVALHO, M.; CÂMARA, G.; MONTEIRO, A. *Análise Espacial de Dados Geográficos*. [S.l.]: EMBRAPA, 2004. Citado na página 11.

EHLERS, R. S. Inferência bayesiana. *Departamento de Matemática Aplicada e Estatıstica, ICMC-USP*, 2011. Citado na página 27.

ELIAS, J.; HARMSEN, D.; CLAUS, H.; HELLENBRAND, W.; FROSCH, M.; VOGEL, U. Spatiotemporal analysis of invasive meningococcal disease, germany. Robert Koch-Institut, Infektionsepidemiologie, 2006. Citado na página 12.

GEMAN, S.; GEMAN, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6, n. 6, p. 721–741, Nov 1984. ISSN 0162-8828. Citado na página 32.

GÓMEZ-RUBIO, V.; LÓPEZ-QUÍLEZ, A. Statistical methods for the geographical analysis of rare diseases. In: PAZ, M. Posada de la; GROFT, S. C. (Ed.). *Rare Diseases Epidemiology*. Springer Netherlands, 2010, (Advances in Experimental Medicine and Biology, v. 686). p. 151–171. ISBN 978-90-481-9484-1. Disponível em: <http://dx.doi.org/10.1007/978-90-481-9485-8_10>. Citado na página 22.

GREEN, C.; HOPPA, R. D.; YOUNG, T. K.; BLANCHARD, J. Geographic analysis of diabetes prevalence in an urban area. *Social science & medicine*, Elsevier, v. 57, n. 3, p. 551–560, 2003. Citado na página 12.

JEFFREYS, H. *Theory of Probability*. third. [S.l.]: Clarendon Press, 1967. Citado na página 43.

JEMAL, A.; KULLDORFF, M.; DEVESA, S. S.; HAYES, R. B.; FRAUMENI, J. F. A geographic analysis of prostate cancer mortality in the united states, 1970–89. *International Journal of Cancer*, Wiley Online Library, v. 101, n. 2, p. 168–174, 2002. Citado na página 26.

JENNINGS, J. M.; CURRIERO, F. C.; CELENTANO, D.; ELLEN, J. M. Geographic identification of high gonorrhea transmission areas in baltimore, maryland. *American Journal of Epidemiology*, Oxford Univ Press, v. 161, n. 1, p. 73–80, 2005. Citado na página 12.

JUNG, I.; KULLDORFF, M.; KLASSEN, A. C. A spatial scan statistic for ordinal data. *Statistics in Medicine*, Wiley Online Library, v. 26, n. 7, p. 1594–1607, 2007. Citado na página 12.

JUNG, I.; KULLDORFF, M.; RICHARD, O. J. A spatial scan statistic for multinomial data. *Statistics in medicine*, Wiley Online Library, v. 29, n. 18, p. 1910–1918, 2010. Citado na página 12.

KASS, R. E.; RAFTERY, A. E. Bayes factors. *Journal of the American Statistical Association*, v. 90, n. 430, p. 773–795, 1995. Disponível em: <http://www.tandfonline-.com/doi/abs/10.1080/01621459.1995.10476572>. Citado na página 43.

KLEINMAN, K.; ABRAMS, A.; KULLDORFF, M.; PLATT, R. A model-adjusted space–time scan statistic with an application to syndromic surveillance. *Epidemiology and Infection*, Cambridge Univ Press, v. 133, n. 03, p. 409–419, 2005. Citado na página 12.

KULLDORFF, M. A spatial scan statistic. *Communications in Statistics - Theory and Methods*, v. 26, n. 6, p. 1481–1496, 1997. Disponível em: <http://dx.doi.org/10.1080-/03610929708831995>. Citado 5 vezes nas páginas 5, 6, 12, 15 e 26.

KULLDORFF, M.; HUANG, L.; KONTY, K. A scan statistic for continuous data based on the normal probability model. *International journal of health geographics*, BioMed Central Ltd, v. 8, n. 1, p. 58, 2009. Citado na página 42.

KULLDORFF, M.; MOSTASHARI, F.; DUCZMAL, L.; YIH, W. K.; KLEINMAN, K.; PLATT, R. Multivariate scan statistics for disease surveillance. *Statistics in Medicine*, Wiley Online Library, v. 26, n. 8, p. 1824–1833, 2007. Citado na página 12.

KULLDORFF, M.; TANGO, T.; PARK, P. J. Power comparisons for disease clustering tests. *Computational Statistics and Data Analysis*, n. 42, p. 665–684, 2003. Citado na página 39.

KUMARASWAMY, P. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, Elsevier, v. 46, n. 1, p. 79–88, 1980. Citado na página 50.

LAWSON, A. B.; KLEINMAN, K. Spatial and syndromic surveillance for public health. Wiley Online Library, 2005. Citado na página 12.

MINAMISAVA, R.; NOUER, S. S.; NETO, O. L. de M.; MELO, L. K.; ANDRADE, A. L. S. Spatial clusters of violent deaths in a newly urbanized region of brazil: Highlighting the social disparities. *International journal of health geographics*, BioMed Central, v. 8, n. 1, p. 66–66, 2009. Citado na página 12.

NEILL, D. B.; MOORE, A. W.; COOPER, G. F. A bayesian spatial scan statistic. In: WEISS, Y. et al. (Ed.). *Advances in Neural Information Processing Systems 18*. MIT Press, 2006. p. 1003–1010. Disponível em: <http://papers.nips.cc/paper/2819-a-bayesian-spatial-scan-statistic.pdf>. Citado 8 vezes nas páginas 5, 6, 12, 15, 26, 27, 29 e 31.

OPENSHAW, S.; CHARLTON, M.; WYMER, C.; CRAFT, A. A mark 1 geographical analysis machine for the automated analysis of point data sets. *International Journal of Geographical Information Systems*, v. 1, n. 4, p. 335–358, 1987. Disponível em: <http://dx.doi.org/10.1080/02693798708927821>. Citado na página 12.

SHEN, Y.; COOPER, G. A new prior for bayesian anomaly detection. *Methods Inf Med*, v. 49, n. 1, p. 44–53, 2010. Citado na página 27.

TANNER, M. A.; WONG, W. H. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, Taylor & Francis, v. 82, n. 398, p. 528–540, 1987. Citado na página 32.

TUYL, F.; GERLACH, R.; MENGERSEN, K. et al. Posterior predictive arguments in favor of the bayes-laplace prior as the consensus prior for binomial and multinomial parameters. *Bayesian analysis*, International Society for Bayesian Analysis, v. 4, n. 1, p. 151–158, 2009. Citado na página 30.

WAKEFIELD, J.; KIM, A. A bayesian model for cluster detection. *Biostatistics*, Biometrika Trust, v. 14, n. 4, p. 752–765, 2013. Citado na página 26.

ZHANG, Z.; ASSUNÇÃO, R.; KULLDORFF, M. Spatial scan statistics adjusted for multiple clusters. *Journal of Probability and Statistics*, Hindawi Publishing Corporation, v. 2010, 2010. Citado na página 27.

Anexos

ANEXO A – Programações

A.1 Scan Binomial

```
#Scan - Binomial
#Input data:
#pop=population
#cases=number of cases
#x=x coordinate
#y=y coordinate
#nsim = number of simulations

scanbin<-function(pop,cases,x,y,nsim){

  a<-cbind(pop,cases,x,y)
  coord<-cbind(x,y)
  n<-nrow(a)

  #Distance Matrix
  dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
  ss<-seq(1,n,1)
  d<-cbind(dist,ss)

  #Auxiliary variables
  N=sum(a[,1]) #Total population
  C=sum(a[,2]) #Total number of cases
  cmax=n
  nmax=0.25*N  #Maximum population allowed inside a cluster


    R<-matrix(NA,ncol=n,nrow=n)
    P<-matrix(NA,ncol=n,nrow=n)
    ex_matrix<-matrix(NA,ncol=n,nrow=n)
    matrix_a<-matrix(NA,ncol=n,nrow=n)
    for(j in 1:n){
      d<-d[order(d[,j]),]
      R[,j]<-d[,n+1]
      for (i in 1:n) {
        P[i,j]<-sum(a[R[1:i,j],1])
        ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
        if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
          matrix_a[i,j]<-1}
          else {  matrix_a[i,j]<-0}
          }
      }
regions<-which(matrix_a==1,arr.ind=TRUE)

    nz=0
    xz=0
```

```
  llr_matrix<-matrix(0,ncol=n,nrow=n)
  p_matrix<-matrix(0,ncol=n,nrow=n)
  theta<-C/N

for(ii in 1:nrow(regions)){
    nz=0
    xz=0
  i<-regions[ii,1]
  j<-regions[ii,2]

            xz<-sum(a[R[i:1,j],2])
          nz<-sum(a[R[i:1,j],1])
          xz_bar<-C-xz
          nz_bar<-N-nz
          theta_0<-xz_bar/nz_bar
          theta_1<-xz/nz
          llr_matrix[i,j] <- xz*log(theta_1)+(nz-xz)*log(1-theta_1)+
          xz_bar*log(theta_0)+(nz_bar-xz_bar)*log(1-theta_0)-
          C*log(theta)-(N-C)*log(1-theta)
                       }

  t<-max(na.omit(llr_matrix)) #Maximum LLR Value
  max<-which(llr_matrix==t, arr.ind=TRUE)
  clustera<-sort(R[1:max[1,1],max[1,2]]) #Detected cluster
  ncluster<-sum(a[clustera,1]) #Population inside detected cluster
  xcluster<-sum(a[clustera,2]) #Observed Cases inside detected cluster
  expected_cases<-ncluster*C/N #Expected Cases inside detected cluster

    #Simulation
    data_sim<-cbind(a,rep(0,n))
    data_sim[,2]<-0
      for (i in 1:n){
        data_sim[i,5]=data_sim[i,1]/N
                    }
      for (i in 2:n){
       data_sim[i,5]=data_sim[i,5]+data_sim[i-1,5]
                    }

      llremp<-rep(NA,nsim) #Vector of empiricals LLR
      for (yy in 1:nsim){ #Simulation loop
          data_sim[,2]<-0
          aux<-runif(C)
          for (i in 1:length(aux)){
            j<-1
            while (data_sim[j,5]<aux[i]){j<-j+1}
            data_sim[j,2]<-data_sim[j,2]+1
                                    }
          llr_matrixs<-matrix(0,ncol=n,nrow=n)

          for (j in 1:n){
            nz<-0
            xz<-0
            for (i in 1:n){
                nz<-P[i,j]
                if (nz <= nmax) {
                  xz<-xz+data_sim[R[i,j],2]
```

```
                    xz_bar<-C-xz
                    nz_bar<-N-nz
                    theta_0<-xz_bar/nz_bar
                    theta_1<-xz/nz
                    llr_matrixs[i,j] <- xz*log(theta_1)+
                    (nz-xz)*log(1-theta_1)+
                    xz_bar*log(theta_0)+(nz_bar-xz_bar)*log(1-theta_0)-
                    C*log(theta)-(N-C)*log(1-theta)
                  } else {j+1}

                          }
                          }
                llremp[yy]<-max((llr_matrixs),na.rm=TRUE)
                yy<-yy+1
                          }
#End of simulations

  z_95<-quantile(llremp,0.95)
  if(t>z_95){signif<-"TRUE"} else {signif<-"FALSE"}

  output<-list(t=t[1],cluster_population=ncluster[1],
  cluster_size=length(clustera),
  expected_cases=expected_cases[1],observed_cases=xcluster[1],
  Significance=signif,cluster=clustera)

plot1<-plot(x,y,main="Scan Binomial")
pt<-points(x=x[clustera],y=y[clustera],col = "red", pch=21, cex = 2)

return(list(output,plot1, pt))


}#end
```

A.2 Scan ZIB

```
#Scan -  Zero Inflated Binomial
#Input data:
#pop=population
#cases=number of cases
#x=x coordinate
#y=y coordinate
#delta=1 for Structural Zeros, 0 for sampling Zeros
#nsim = number of simulations

scanzib<-function(pop,cases,x,y,delta,nsim){
  a<-cbind(pop,cases,x,y)
  coord<-cbind(x,y)
  n<-nrow(a)

  #Distance Matrix
  dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
  ss<-seq(1,n,1)
  d<-cbind(dist,ss)

  #Auxiliary variables
  N=sum(a[,1]) #Total population
  C=sum(a[,2]) #Total number of cases
  cmax=n
  nmax=0.25*N  #Maximum population allowed inside a cluster

ad<-matrix(NA,ncol=2,nrow=n)
    ad[,1]<-a[,1]*(1-delta)
    ad[,2]<-a[,2]*(1-delta)
    R<-matrix(NA,ncol=n,nrow=n)
    P<-matrix(NA,ncol=n,nrow=n)
    ex_matrix<-matrix(NA,ncol=n,nrow=n)
    matrix_a<-matrix(NA,ncol=n,nrow=n)
    for(j in 1:n){
      d<-d[order(d[,j]),]
      R[,j]<-d[,n+1]
      for (i in 1:n) {
        P[i,j]<-sum(a[R[1:i,j],1])
        ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
        if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
          matrix_a[i,j]<-1}
          else {  matrix_a[i,j]<-0}
          }
      }
regions<-which(matrix_a==1,arr.ind=TRUE)

    nz=0
    xz=0

  llr_matrix<-matrix(0,ncol=n,nrow=n)
  p_matrix<-matrix(0,ncol=n,nrow=n)
  theta<-C/N

for(ii in 1:nrow(regions)){
    nz=0
    xz=0
```

```
i<-regions[ii,1]
j<-regions[ii,2]

          xz<-sum(a[R[i:1,j],2])
        nz<-sum(a[R[i:1,j],1])
        xz_bar<-C-xz
        nz_bar<-N-nz
        theta_0<-xz_bar/nz_bar
        theta_1<-xz/nz
        llr_matrix[i,j] <- xz*log(theta_1)+(nz-xz)*log(1-theta_1)+
        xz_bar*log(theta_0)+(nz_bar-xz_bar)*log(1-theta_0)-
        C*log(theta)-(N-C)*log(1-theta)
                    }
a<-cbind(pop,cases,x,y)
coord<-cbind(x,y)
n<-nrow(a)

#Distance Matrix
dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
ss<-seq(1,n,1)
d<-cbind(dist,ss)



R<-matrix(NA,ncol=n,nrow=n)
P<-matrix(NA,ncol=n,nrow=n)
for(j in 1:n){
  d<-d[order(d[,j]),]
  R[,j]<-d[,n+1]
  for (i in 1:n) {P[i,j]<-sum(a[R[1:i,j],1])}
}


#Auxiliary variables
N=sum(a[,1]) #Total population
C=sum(a[,2]) #Total number of cases
cmax=n
nmax=0.25*N  #Maximum population allowed inside a cluster

nz=0
xz=0
llr_matrix<-matrix(0,ncol=n,nrow=n)
theta<-C/N

for (j in 1:n){
  nz<-0
  xz<-0
  for (i in 1:n){

    if (nz <= nmax) {
      xz<-xz+ad[R[i,j],2]
      nz<-nz+ad[R[i,j],1]
      ad1<-ad[-R[1:i,j],]
      if (length(ad1)>2){
      xz_bar<-sum(ad1[,2])
      nz_bar<-sum(ad1[,1])} else {
```

```
          xz_bar<-ad1[2]
          nz_bar<-ad1[1]
        }
        theta_0<-xz_bar/nz_bar
        theta_1<-xz/nz
        llr_matrix[i,j] <- xz*log(theta_1)+
        (nz-xz)*log(1-theta_1)+xz_bar*log(theta_0)+
        (nz_bar-xz_bar)*log(1-theta_0)-sum(ad[,2])*log(theta)-
        (sum(ad[,1])-sum(ad[,2]))*log(1-theta)
      } else {j+1}

    }
  }

  t<-max(na.omit(llr_matrix)) #Maximum LLR Value
  max<-which(llr_matrix==t, arr.ind=TRUE)
  clustera<-sort(R[1:max[1,1],max[1,2]]) #Detected cluster
  ncluster<-sum(a[clustera,1]) #Population inside detected cluster
  xcluster<-sum(a[clustera,2]) #Observed Cases inside detected cluster
  expected_cases<-ncluster*C/N #Expected Cases inside detected cluster

    #Simulation

d1<-a[-which(sz==1),]
d2<-a[which(sz==1),]
d2<-cbind(d2,rep(0,nrow(d2)))

delta1<-which(sz==1)
delta2<-which(sz==0)
delta<-c(rep(1,length(delta1)),rep(0,length(delta2)))

data_sim<-cbind(d1,rep(0,nrow(d1)))
data_sim[,2]<-0
for (i in 1:nrow(d1)){
  data_sim[i,5]=data_sim[i,1]/sum(data_sim[,1])
}
for (i in 2:nrow(d1)){
  data_sim[i,5]=data_sim[i,5]+data_sim[i-1,5]
}

        data_sim2<-rbind(d2,data_sim)

coor1<-coord[-which(sz==1),]
coor2<-coord[which(sz==1),]
coord<-rbind(coor2,coor1)

dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
ss<-seq(1,n,1)
d<-cbind(dist,ss)

R<-matrix(NA,ncol=n,nrow=n)
P<-matrix(NA,ncol=n,nrow=n)
for(j in 1:n){
  d<-d[order(d[,j]),]
  R[,j]<-d[,n+1]
  for (i in 1:n) {P[i,j]<-sum(data_sim2[R[1:i,j],1])}
}
```

```
ad<-matrix(NA,ncol=2,nrow=n)
ad[,1]<-a[,1]*(1-delta)
ad[,2]<-a[,2]*(1-delta)

llremp<-rep(NA,nsim) #Vector of empiricals LLR
for (yy in 1:nsim){ #Simulation loop
  data_sim<-data_sim2
  data_sim[,2]<-0
  aux<-runif(C)
  for (i in 1:length(aux)){
    j<-1
    while (data_sim[j,5]<aux[i]){j<-j+1}
    data_sim[j,2]<-data_sim[j,2]+1
  }

        data_sim<-rbind(d2,data_sim)

  adsim<-matrix(NA,ncol=2,nrow=n)
  adsim[,1]<-ad[,1]
  for (i in 1:n){
    adsim[i,2]<-data_sim[i,2]*(1-delta[i])

  }

  llr_matrixs<-matrix(0,ncol=n,nrow=n)

  for (j in 1:n){

    for (i in 1:n){
                iz<-R[1:i,j]  #index inside
        oz<-R[-(1:i),j] #index outside
            c_o<-sum(data_sim[iz,2])
                ex<-sum(data_sim[iz,1])*C/N
            nzz<-sum(data_sim[R[1:i,j],1])
      if (j>R[i,j]){llr_matrixs[i,j]<-llr_matrixs[j,i]} else {
        if (nzz <= nmax) {
      if (c_o>ex){
             nz<-0
          xz<-0
          xz<-xz+sum(adsim[R[1:i,j],2])
          nz<-nz+sum(adsim[R[1:i,j],1])
          ad1<-adsim[-R[1:i,j],]
          if (length(ad1)>2){
            xz_bar<-sum(ad1[,2])
            nz_bar<-sum(ad1[,1])} else {
              xz_bar<-ad1[2]
              nz_bar<-ad1[1]
            }
          theta_0<-xz_bar/nz_bar
          theta_1<-xz/nz
          llr_matrixs[i,j] <- xz*log(theta_1)+
          (nz-xz)*log(1-theta_1)+
          xz_bar*log(theta_0)+
          (nz_bar-xz_bar)*log(1-theta_0)-
          sum(adsim[,2])*log(theta)-(sum(adsim[,1])-
          sum(adsim[,2]))*log(1-theta)
```

```
        } else {i<-i+1}} else {j<-j+1}
      }
    }
  }
  llremp[yy]<-max((llr_matrixs),na.rm=TRUE)
  yy<-yy+1
}
#End of simulations

z_95_zib<-quantile(llremp,0.95)
  if(t>z_95_zib){signif<-"TRUE"} else {signif<-"FALSE"}

  output<-list(t=t[1],cluster_population=ncluster[1],
  cluster_size=length(clustera),expected_cases=expected_cases[1],
  observed_cases=xcluster[1],Significance=signif,cluster=clustera)

plot1<-plot(x,y,main="Scan ZIB")
pt<-points(x=x[clustera],y=y[clustera],col = "red", pch=21, cex = 2)

return(list(output,plot1, pt))
}#end
```

A.3 Scan ZIB-EM

```
#Scan -  Zero Inflated Binomial - Incomplete Data
#Input data:
#pop=population
#cases=number of cases
#x=x coordinate
#y=y coordinate
#nsim = number of simulations

scanzib_EM<-function(pop,cases,x,y,nsim){

  a<-cbind(pop,cases,x,y)
  coord<-cbind(x,y)
  n<-nrow(a)

  #Distance Matrix
  dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
  ss<-seq(1,n,1)
  d<-cbind(dist,ss)


  #Auxiliary variables
  N=sum(a[,1]) #Total population
  C=sum(a[,2]) #Total number of cases
  cmax=n
  nmax=0.25*N  #Maximum population allowed inside a cluster

    R<-matrix(NA,ncol=n,nrow=n)
    P<-matrix(NA,ncol=n,nrow=n)
    ex_matrix<-matrix(NA,ncol=n,nrow=n)
    matrix_a<-matrix(NA,ncol=n,nrow=n)
    for(j in 1:n){
      d<-d[order(d[,j]),]
      R[,j]<-d[,n+1]
      for (i in 1:n) {
        P[i,j]<-sum(a[R[1:i,j],1])
        ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
        if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
          matrix_a[i,j]<-1}
          else {  matrix_a[i,j]<-0}
          }
      }
regions<-which(matrix_a==1,arr.ind=TRUE)

    nz=0
    xz=0

  llr_matrix<-matrix(0,ncol=n,nrow=n)
  p_matrix<-matrix(0,ncol=n,nrow=n)
  theta<-C/N

for(ii in 1:nrow(regions)){
  nz=0
  xz=0
  i<-regions[ii,1]
  j<-regions[ii,2]
```

```
iz<-R[1:i,j]  #index inside
oz<-R[-(1:i),j] #index outside

      #EM Algorithm
      maxit<-10

  d=rep(0,n)
  I=which(a[,2]==0, arr.ind=TRUE)

  izz=intersect(iz,I) #regions with zero cases inside the cluster
  ozz=intersect(oz,I) #regions with zero cases outside the cluster

  xx=a[I,2]
  nn=length(xx)
  d[I]=.5

  M=matrix(0,nrow=maxit+1,ncol=n)
  M[1,]=d

  for (ii in 1:maxit){
      theta0=sum(a[oz,2]*(1-d[oz]))/sum(a[oz,1]*(1-d[oz]))
      thetaz=sum(a[iz,2]*(1-d[iz]))/sum(a[iz,1]*(1-d[iz]))
      p=sum(d)/n
     d[ozz]=p*((p+(1-p)*(1-theta0)^(a[ozz,1])))^(-1)
              d[izz]=p*((p+(1-p)*(1-thetaz)^(a[izz,1])))^(-1)
      M[ii+1,]=d
                  }

  delta<-M[maxit+1,]
  p<-sum(delta)/n
  p_matrix[i,j]<-p

  ad<-matrix(NA,ncol=2,nrow=n)
  ad[,1]<-a[,1]*(1-delta)
  ad[,2]<-a[,2]*(1-delta)


      xz<-sum(ad[R[1:i,j],2])
      nz<-sum(ad[R[1:i,j],1])
      ad1<-ad[-R[1:i,j],]
      if (length(ad1)>2){
       xz_bar<-sum(ad1[,2])
       nz_bar<-sum(ad1[,1])} else {
                       xz_bar<-ad1[2]
                       nz_bar<-ad1[1]
                                       }
      theta_0<-xz_bar/nz_bar
      theta_1<-xz/nz
      llr_matrix[i,j] <- xz*log(theta_1)+
      (nz-xz)*log(1-theta_1)+xz_bar*log(theta_0)+
      (nz_bar-xz_bar)*log(1-theta_0)-
      sum(ad[,2])*log(theta)-(sum(ad[,1])-sum(ad[,2]))*log(1-theta)
    }


t<-max(na.omit(llr_matrix)) #Maximum LLR Value
max<-which(llr_matrix==t, arr.ind=TRUE)
```

```
  clustera<-sort(R[1:max[1,1],max[1,2]]) #Detected cluster
  ncluster<-sum(a[clustera,1]) #Population inside detected cluster
  xcluster<-sum(a[clustera,2]) #Observed Cases inside detected cluster
  expected_cases<-ncluster*C/N #Expected Cases inside detected cluster

#Simulation

source("ZIB.R") #Require Scan ZIB

llremp<-rep(NA,nsim) #Vector of empiricals LLR
for (yy in 1:nsim){ #Simulation loop
  data_sim<-cbind(a,rep(0,n))
  data_sim[,2]<-0
  delta<-rep(0,nrow(R))
  #probs=a[,1]/N
  #S<-data_sim

  for (i in 1:n){
    u<-runif(1)
    if (u<p_cluster[1]) {
      delta[i]<-1
    } }
  data_sim[,5]<-delta
  I0<-which(delta==1, arr.ind=TRUE) #simulated structural 0 regions

  if (length(I0) > 0){

    S<-data_sim[-I0,]
    rr<-data_sim[I0,]
    }
  probs=S[,1]/sum(S[,1])
  II0<-which(probs==0, arr.ind=TRUE)

  SS1<-S[-II0,]
  SS2<-S[II0,]
  probs<-probs[-II0]
  for (ii in 2:length(probs)) {
    probs[ii]<-probs[ii]+probs[ii-1]
  }

  aux<-runif(C)
  for (i in 1:length(aux)){
    if (aux[i]>probs[length(probs)])
    {SS1[nrow(SS1),2]<-SS1[nrow(SS1),2]+1}  else {
      j<-1
      while (probs[j]<aux[i]){
        j<-j+1
      }
      SS1[j,2]<-SS1[j,2]+1
    } }

  if (length(I0) > 0){
    data_sim<-rbind(SS1,SS2,rr)}

  aaa<-scanzib(data_sim[,1],data_sim[,2],data_sim[,3],
  data_sim[,4],data_sim[,5],0)
  llremp[yy]<-aaa[[1]]$t
```

```
  yy<-yy+1

}


z_95<-quantile(llremp,0.95,na.rm=TRUE)
if(t>z_95){signif<-"TRUE"} else {signif<-"FALSE"}

output<-list(t=t[1],cluster_population=ncluster[1],
cluster_size=length(clustera),expected_cases=expected_cases[1],
observed_cases=xcluster[1],
Significance=signif,cluster=clustera)


plot1<-plot(x,y,main="Scan␣ZIB-EM")
pt<-points(x=x[clustera],y=y[clustera],col = "red", pch=21, cex = 2)

return(list(output,plot1, pt))


}#end
```

A.4 Scan Beta-Binomial

```
#Scan - Beta-Binomial
#Input data:
#pop=population
#cases=number of cases
#alpha,beta=priors parameters
#x=x coordinate
#y=y coordinate


scan_betabin<-function(pop,cases,x,y,alpha=NULL,beta=NULL){
  if(is.null(alpha) & is.null(beta)) {  ##Alpha and Beta Unknown
    a<-cbind(pop,cases,x,y)
    coord<-cbind(x,y)
    n<-nrow(a)
    alpha<-1
    beta<-1

    N=sum(a[,1]) #Total population
    C=sum(a[,2]) #Total number of cases
    cmax=n
    nmax=0.25*N  #Maximum population allowed inside a cluster

    #Distance Matrix
    dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
    ss<-seq(1,n,1)
    d<-cbind(dist,ss)

    R<-matrix(NA,ncol=n,nrow=n)
    P<-matrix(NA,ncol=n,nrow=n)
    ex_matrix<-matrix(NA,ncol=n,nrow=n)
    matrix_a<-matrix(NA,ncol=n,nrow=n)
    for(j in 1:n){
      d<-d[order(d[,j]),]
      R[,j]<-d[,n+1]
      for (i in 1:n) {
        P[i,j]<-sum(a[R[1:i,j],1])
        ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
        if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
          matrix_a[i,j]<-1}
          else {  matrix_a[i,j]<-0}
          }
      }

    nreg<-sum(P<n/2)
regions<-which(matrix_a==1,arr.ind=TRUE)

    nz=0
    xz=0
    llr_matrix<-matrix(0,ncol=n,nrow=n)

    xhz_matrix<-llr_matrix
        bfac<-matrix(NA,ncol=n,nrow=n)
    theta<-C/N
    p_1<-0.5
    ph0<-1-p_1
```

```
    phz<-p_1/nreg
  pxh0<-beta(sum(a[,2])+alpha,sum(a[,1])-sum(a[,2])+beta)/beta(alpha,beta)


for(ii in 1:nrow(regions)){
  i<-regions[ii,1]
  j<-regions[ii,2]
  iz<-R[1:i,j]  #index inside
  oz<-R[-(1:i),j] #index outside

  az<-alpha*sum(a[iz,2])/C
  bz<-beta*(sum(a[iz,1])-sum(a[iz,2]))/(N-C)
  ao<-alpha*sum(a[oz,2])/C
  bo<-beta*(sum(a[oz,1])-sum(a[oz,2]))/(N-C)

  pxhz<-(beta(sum(a[iz,2])+az, sum(a[iz,1])-sum(a[iz,2])+bz)/beta(az,bz))*
  (beta(sum(a[oz,2])+ao, sum(a[oz,1])-sum(a[oz,2])+bo)/beta(ao,bo))
  xhz_matrix[i,j]<-pxhz*phz
  bfac[i,j]<-pxhz/pxh0


}

        p_d<-sum(xhz_matrix,na.rm=TRUE)+pxh0*ph0
        llr_matrix<-xhz_matrix/p_d

    t<-max(na.omit(llr_matrix)) #Maximum Posterior Probability Value
    max<-which(llr_matrix==t, arr.ind=TRUE)
        bf<-bfac[max[1,1],max[1,2]] #Bayes Factor
    clustera<-sort(R[1:max[1,1],max[1,2]]) #Detected cluster
    ncluster<-sum(a[clustera,1]) #Population inside detected cluster
    xcluster<-sum(a[clustera,2]) #Observed Cases inside detected cluster
    expected_cases<-ncluster*C/N #Expected Cases inside detected cluster

    output<-list(t=t[1],bf=bf[1],cluster_population=ncluster[1],
    cluster_size=length(clustera),expected_cases=expected_cases[1],
    observed_cases=xcluster[1],cluster=clustera)


plot1<-plot(x,y,main="Scan␣Beta-Binomial␣-␣Non-Informative␣Prior")
pt<-points(x=x[clustera],y=y[clustera],col = "red", pch=21, cex = 2)

return(list(output,plot1, pt))

  }

  else {   ##Alpha and Beta Known
   a<-cbind(pop,cases,x,y)
    coord<-cbind(x,y)
    n<-nrow(a)

    N=sum(a[,1]) #Total population
    C=sum(a[,2]) #Total number of cases
    cmax=n
    nmax=0.10*N  #Maximum population allowed inside a cluster
```

```
    #Distance Matrix
    dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
    ss<-seq(1,n,1)
    d<-cbind(dist,ss)

    R<-matrix(NA,ncol=n,nrow=n)
    P<-matrix(NA,ncol=n,nrow=n)
    ex_matrix<-matrix(NA,ncol=n,nrow=n)
    matrix_a<-matrix(NA,ncol=n,nrow=n)
    for(j in 1:n){
      d<-d[order(d[,j]),]
      R[,j]<-d[,n+1]
      for (i in 1:n) {
        P[i,j]<-sum(a[R[1:i,j],1])
        ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
        if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
          matrix_a[i,j]<-1}
          else {  matrix_a[i,j]<-0}
          }
      }

    nreg<-sum(P<n/2)
regions<-which(matrix_a==1,arr.ind=TRUE)

    nz=0
    xz=0
    llr_matrix<-matrix(0,ncol=n,nrow=n)

    xhz_matrix<-llr_matrix
        bfac<-matrix(NA,ncol=n,nrow=n)
    theta<-C/N
    p_1<-0.5
    ph0<-1-p_1
    phz<-p_1/nreg
  pxh0<-beta(sum(a[,2])+alpha, sum(a[,1])-sum(a[,2])+beta)/beta(alpha,beta)


for(ii in 1:nrow(regions)){
  i<-regions[ii,1]
  j<-regions[ii,2]
  iz<-R[1:i,j]  #index inside
  oz<-R[-(1:i),j] #index outside

  az<-alpha*sum(a[iz,2])/C
  bz<-beta*(sum(a[iz,1])-sum(a[iz,2]))/(N-C)
  ao<-alpha*sum(a[oz,2])/C
  bo<-beta*(sum(a[oz,1])-sum(a[oz,2]))/(N-C)

pxhz<-(beta(sum(a[iz,2])+az, sum(a[iz,1])-sum(a[iz,2])+bz)/beta(az,bz))*
  (beta(sum(a[oz,2])+ao, sum(a[oz,1])-sum(a[oz,2])+bo)/beta(ao,bo))
  xhz_matrix[i,j]<-pxhz*phz
  bfac[i,j]<-pxhz/pxh0

}

                p_d<-sum(xhz_matrix,na.rm=TRUE)+pxh0*ph0
                llr_matrix<-xhz_matrix/p_d
```

```
    t<-max(na.omit(llr_matrix)) #Maximum Posterior Probability Value
    max<-which(llr_matrix==t, arr.ind=TRUE)
        bf<-bfac[max[1,1],max[1,2]] #Bayes Factor
    clustera<-sort(R[1:max[1,1],max[1,2]]) #Detected cluster
    ncluster<-sum(a[clustera,1]) #Population inside detected cluster
    xcluster<-sum(a[clustera,2]) #Observed Cases inside detected cluster
    expected_cases<-ncluster*C/N #Expected Cases inside detected cluster


    output<-list(t=t[1],bf=bf[1],cluster_population=ncluster[1],
    cluster_size=length(clustera),expected_cases=expected_cases[1],
    observed_cases=xcluster[1],cluster=clustera)


plot1<-plot(x,y,main="Scan␣Beta-Binomial")
pt<-points(x=x[clustera],y=y[clustera],col = "red", pch=21, cex = 2)

return(list(output,plot1, pt))
  }



}#end
```

A.5 Scan ZIBB

```
#Scan - Bayesian ZIB
#Input data:
#pop=population
#cases=number of cases
#alpha,beta=priors parameters
#x=x coordinate
#y=y coordinate


scan_zibb<-function(pop,cases,x,y,delta,alpha=NULL,beta=NULL){
  if(is.null(alpha) & is.null(beta)) {  ##Alpha and Beta Unknown
    a<-cbind(pop,cases,x,y)
    coord<-cbind(x,y)
    n<-nrow(a)
    alpha<-1
    beta<-1

    N=sum(a[,1]) #Total population
    C=sum(a[,2]) #Total number of cases
    cmax=n
    nmax=0.25*N  #Maximum population allowed inside a cluster

    #Distance Matrix
    dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
    ss<-seq(1,n,1)
    d<-cbind(dist,ss)

    R<-matrix(NA,ncol=n,nrow=n)
    P<-matrix(NA,ncol=n,nrow=n)
    ex_matrix<-matrix(NA,ncol=n,nrow=n)
    matrix_a<-matrix(NA,ncol=n,nrow=n)
    for(j in 1:n){
      d<-d[order(d[,j]),]
      R[,j]<-d[,n+1]
      for (i in 1:n) {
        P[i,j]<-sum(a[R[1:i,j],1])
        ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
        if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
          matrix_a[i,j]<-1}
          else {  matrix_a[i,j]<-0}
          }
      }

regions<-which(matrix_a==1,arr.ind=TRUE)
    nreg<-sum(P<n/2)

ad<-matrix(NA,ncol=2,nrow=n)
ad[,1]<-a[,1]*(1-delta)
ad[,2]<-a[,2]*(1-delta)

    nz=0
    xz=0
    llr_matrix<-matrix(0,ncol=n,nrow=n)

    xhz_matrix<-llr_matrix
```

```
        bfac<-matrix(NA,ncol=n,nrow=n)
    theta<-C/N
    p_1<-0.5
    ph0<-1-p_1
    phz<-p_1/nreg

    pxh0<-beta(sum(ad[,2])+alpha, sum(ad[,1])-sum(ad[,2])+beta)/
    beta(alpha,beta)

for(ii in 1:nrow(regions)){
  i<-regions[ii,1]
  j<-regions[ii,2]
  iz<-R[1:i,j]  #index inside
  oz<-R[-(1:i),j] #index outside

  az<-alpha*sum(a[iz,2])/C
  bz<-beta*(sum(a[iz,1])-sum(a[iz,2]))/(N-C)
  ao<-alpha*sum(a[oz,2])/C
  bo<-beta*(sum(a[oz,1])-sum(a[oz,2]))/(N-C)


  pxhz<-(beta(sum(ad[iz,2])+az, sum(ad[iz,1])-
  sum(ad[iz,2])+bz)/beta(az,bz))*
  (beta(sum(ad[oz,2])+ao, sum(ad[oz,1])-sum(ad[oz,2])+bo)/beta(ao,bo))
  bfac[i,j]<-pxhz/pxh0
xhz_matrix[i,j]<-pxhz*phz


}

                p_d<-sum(xhz_matrix,na.rm=TRUE)+pxh0*ph0
                llr_matrix<-xhz_matrix/p_d

        t<-max(na.omit(llr_matrix)) #Maximum Posterior Probability Value
        max<-which(llr_matrix==t, arr.ind=TRUE)
        bf<-bfac[max[1,1],max[1,2]] #Bayes Factor
        clustera<-sort(R[1:max[1,1],max[1,2]]) #Detected cluster
        ncluster<-sum(a[clustera,1]) #Population inside detected cluster
        xcluster<-sum(a[clustera,2]) #Observed Cases inside detected cluster
        expected_cases<-ncluster*C/N #Expected Cases inside detected cluster


output<-list(t=t[1],bf=bf[1],cluster_population=ncluster[1],
cluster_size=length(clustera),expected_cases=expected_cases[1],
observed_cases=xcluster[1],cluster=clustera)

plot1<-plot(x,y,main="Bayesian␣ZIB␣Scan␣-␣Non-Informative␣Prior")
pt<-points(x=x[clustera],y=y[clustera],col = "red", pch=21, cex = 2)

return(list(output,plot1, pt))


  }

  else { ##Alpha and Beta Known
      #Distance Matrix
    dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
```

```
    ss<-seq(1,n,1)
    d<-cbind(dist,ss)

    R<-matrix(NA,ncol=n,nrow=n)
    P<-matrix(NA,ncol=n,nrow=n)
    ex_matrix<-matrix(NA,ncol=n,nrow=n)
    matrix_a<-matrix(NA,ncol=n,nrow=n)
    for(j in 1:n){
      d<-d[order(d[,j]),]
      R[,j]<-d[,n+1]
      for (i in 1:n) {
        P[i,j]<-sum(a[R[1:i,j],1])
        ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
        if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
          matrix_a[i,j]<-1}
          else {  matrix_a[i,j]<-0}
          }
      }

regions<-which(matrix_a==1,arr.ind=TRUE)
    nreg<-sum(P<n/2)

ad<-matrix(NA,ncol=2,nrow=n)
ad[,1]<-a[,1]*(1-delta)
ad[,2]<-a[,2]*(1-delta)

    nz=0
    xz=0
    llr_matrix<-matrix(0,ncol=n,nrow=n)

    xhz_matrix<-llr_matrix
        bfac<-matrix(NA,ncol=n,nrow=n)
    theta<-C/N
    p_1<-0.5
    ph0<-1-p_1
    phz<-p_1/nreg

    pxh0<-beta(sum(ad[,2])+alpha,
    sum(ad[,1])-sum(ad[,2])+beta)/beta(alpha,beta)

for(ii in 1:nrow(regions)){
  i<-regions[ii,1]
  j<-regions[ii,2]
  iz<-R[1:i,j]  #index inside
  oz<-R[-(1:i),j] #index outside

  az<-alpha*sum(a[iz,2])/C
  bz<-beta*(sum(a[iz,1])-sum(a[iz,2]))/(N-C)
  ao<-alpha*sum(a[oz,2])/C
  bo<-beta*(sum(a[oz,1])-sum(a[oz,2]))/(N-C)


  pxhz<-(beta(sum(ad[iz,2])+az, sum(ad[iz,1])-
  sum(ad[iz,2])+bz)/beta(az,bz))*
  (beta(sum(ad[oz,2])+ao, sum(ad[oz,1])-sum(ad[oz,2])+bo)/beta(ao,bo))
  bfac[i,j]<-pxhz/pxh0
xhz_matrix[i,j]<-pxhz*phz
```

```
}

                p_d<-sum(xhz_matrix,na.rm=TRUE)+pxh0*ph0
                llr_matrix<-xhz_matrix/p_d

        t<-max(na.omit(llr_matrix)) #Maximum Posterior Probability Value
        max<-which(llr_matrix==t, arr.ind=TRUE)
        bf<-bfac[max[1,1],max[1,2]] #Bayes Factor
        clustera<-sort(R[1:max[1,1],max[1,2]]) #Detected cluster
        ncluster<-sum(a[clustera,1]) #Population inside detected cluster
        xcluster<-sum(a[clustera,2]) #Observed Cases inside detected cluster
        expected_cases<-ncluster*C/N #Expected Cases inside detected cluster


output<-list(t=t[1],bf=bf[1],cluster_population=ncluster[1],
cluster_size=length(clustera),expected_cases=expected_cases[1],
observed_cases=xcluster[1],cluster=clustera)

plot1<-plot(x,y,main="Bayesian␣ZIB␣Scan")
pt<-points(x=x[clustera],y=y[clustera],col = "red", pch=21, cex = 2)

return(list(output,plot1, pt))


  }

}#end
```

A.6 Scan ZIBB-Gibbs

```
#Scan - Bayesian ZIB - Incomplete Data
#Input data:
#pop=population
#cases=number of cases
#alpha,beta=priors parameters
#x=x coordinate
#y=y coordinate
#ngibss = number of Gibbs Sampling


scan_zibbg<-function(pop,cases,x,y,ngibbs,alpha=NULL,beta=NULL){
  if(is.null(alpha) & is.null(beta)) {  ##Alpha and Beta Unknown
    a<-cbind(pop,cases,x,y)
    coord<-cbind(x,y)
    n<-nrow(a)
    alpha<-1
    beta<-1

    N=sum(a[,1]) #Total population
    C=sum(a[,2]) #Total number of cases
    cmax=n
    nmax=0.25*N  #Maximum population allowed inside a cluster

    #Distance Matrix
    dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
    ss<-seq(1,n,1)
    d<-cbind(dist,ss)

    R<-matrix(NA,ncol=n,nrow=n)
    P<-matrix(NA,ncol=n,nrow=n)
    ex_matrix<-matrix(NA,ncol=n,nrow=n)
    matrix_a<-matrix(NA,ncol=n,nrow=n)
    for(j in 1:n){
      d<-d[order(d[,j]),]
      R[,j]<-d[,n+1]
      for (i in 1:n) {
        P[i,j]<-sum(a[R[1:i,j],1])
        ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
        if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
          matrix_a[i,j]<-1}
          else {  matrix_a[i,j]<-0}
          }
      }

    nreg<-sum(P<n/2)
regions<-which(matrix_a==1,arr.ind=TRUE)



    nz=0
    xz=0
    llr_matrix<-matrix(0,ncol=n,nrow=n)

    xhz_matrix<-llr_matrix
    bfac<-matrix(NA,ncol=n,nrow=n)
    theta<-C/N
```

```
    p_1<-0.5
    ph0<-1-p_1
    phz<-p_1/nreg


i0<-which(a[,2]==0)

#pxH0 calculation
iz<-NULL
oz<-R[,1]

izz<-intersect(iz,i0)
ozz<-intersect(oz,i0)

#Gibbs sampling
az<-alpha*sum(a[iz,2])/C
bz<-beta*(sum(a[iz,1])-sum(a[iz,2]))/(N-C)
ao<-alpha*sum(a[oz,2])/C
bo<-beta*(sum(a[oz,1])-sum(a[oz,2]))/(N-C)
delta<-rep(0,n)

delta[i0]<-0.5
nw<-ngibbs
ng<-ngibbs
dmatrix<-matrix(NA,ncol=n,nrow=nw+ng)
p<-rbeta(1,alpha+sum(delta),beta+sum(1-delta))
ad<-matrix(NA,ncol=2,nrow=n)
ad[,1]<-a[,1]*(1-delta)
ad[,2]<-a[,2]*(1-delta)
theta_o<- rbeta(1,sum(ad[oz,2])+ao,
 sum(ad[oz,1])-sum(ad[oz,2])+bo)   #outside region

delta[ozz]<-p/(p+((1-theta_o)^a[ozz,1])*(1-p))

dmatrix[1,]<-delta
for (zz in 2:nrow(dmatrix)){
  delta<-dmatrix[zz-1,]
  p<-rbeta(1,alpha+sum(delta),beta+sum(1-delta))
  ad<-matrix(NA,ncol=2,nrow=n)
  ad[,1]<-a[,1]*(1-delta)
  ad[,2]<-a[,2]*(1-delta)
  theta_o<- rbeta(1,sum(ad[oz,2])+ao,
  sum(ad[oz,1])-sum(ad[oz,2])+bo)   #outside region
  delta[ozz]<-p/(p+((1-theta_o)^a[ozz,1])*(1-p))
  dmatrix[zz,]<-delta

}

deltaz<-dmatrix[-(1:nw),]
for (z in 1:n){
  delta[z]<-mean(deltaz[,z])}

ad[,1]<-a[,1]*(1-delta)
ad[,2]<-a[,2]*(1-delta)

pxh0<-beta(sum(ad[,2])+alpha, sum(ad[,1])-
sum(ad[,2])+beta)/beta(alpha,beta)
```

```
#pxHZ
for(ii in 1:nrow(regions)){
  i<-regions[ii,1]
  j<-regions[ii,2]
  iz<-R[1:i,j]  #index inside
  oz<-R[-(1:i),j] #index outside

  izz<-intersect(iz,i0)
  ozz<-intersect(oz,i0)

  #Gibbs sampling
  az<-alpha*sum(a[iz,2])/C
  bz<-beta*(sum(a[iz,1])-sum(a[iz,2]))/(N-C)
  ao<-alpha*sum(a[oz,2])/C
  bo<-beta*(sum(a[oz,1])-sum(a[oz,2]))/(N-C)
  delta<-rep(0,n)

  delta[i0]<-0.5
  nw<-ngibbs
  ng<-ngibbs
  dmatrix<-matrix(NA,ncol=n,nrow=nw+ng)
  p<-rbeta(1,alpha+sum(delta),beta+sum(1-delta))
  ad<-matrix(NA,ncol=2,nrow=n)
  ad[,1]<-a[,1]*(1-delta)
  ad[,2]<-a[,2]*(1-delta)
  theta_z<-rbeta(1,sum(ad[iz,2])+az, sum(ad[iz,1])-
  sum(ad[iz,2])+bz) #inside region
  theta_o<- rbeta(1,sum(ad[oz,2])+ao, sum(ad[oz,1])-
  sum(ad[oz,2])+bo)   #outside region

  delta[izz]<-p/(p+((1-theta_z)^a[izz,1])*(1-p))
  delta[ozz]<-p/(p+((1-theta_o)^a[ozz,1])*(1-p))

  dmatrix[1,]<-delta
  for (zz in 2:nrow(dmatrix)){
    delta<-dmatrix[zz-1,]
    p<-rbeta(1,alpha+sum(delta),beta+sum(1-delta))
    ad<-matrix(NA,ncol=2,nrow=n)
    ad[,1]<-a[,1]*(1-delta)
    ad[,2]<-a[,2]*(1-delta)
    theta_z<-rbeta(1,sum(ad[iz,2])+az, sum(ad[iz,1])-
    sum(ad[iz,2])+bz) #inside region
    theta_o<- rbeta(1,sum(ad[oz,2])+ao, sum(ad[oz,1])-
    sum(ad[oz,2])+bo)   #outside region

    delta[izz]<-p/(p+((1-theta_z)^a[izz,1])*(1-p))
    delta[ozz]<-p/(p+((1-theta_o)^a[ozz,1])*(1-p))

    dmatrix[zz,]<-delta

  }

  deltaz<-dmatrix[-(1:nw),]
  for (z in 1:n){
    delta[z]<-mean(deltaz[,z])}
```

```
  ad[,1]<-a[,1]*(1-delta)
  ad[,2]<-a[,2]*(1-delta)

  pxhz<-(beta(sum(ad[iz,2])+az, sum(ad[iz,1])-
  sum(ad[iz,2])+bz)/beta(az,bz))*(beta(sum(ad[oz,2])+ao, sum(ad[oz,1])-
  sum(ad[oz,2])+bo)/beta(ao,bo))
  xhz_matrix[i,j]<-pxhz*phz
  bfac[i,j]<-pxhz/pxh0

}
                p_d<-sum(xhz_matrix,na.rm=TRUE)+pxh0*ph0
                llr_matrix<-xhz_matrix/p_d


    t<-max(na.omit(llr_matrix)) #Maximum Posterior Probability
    max<-which(llr_matrix==t, arr.ind=TRUE)
    bf<-bfac[max[1,1],max[1,2]] #Bayes Factor
    clustera<-sort(R[1:max[1,1],max[1,2]]) #cluster

    ncluster<-sum(a[clustera,1]) #population inside cluster
    xcluster<-sum(a[clustera,2]) #cases inside cluster
    expected_cases<-ncluster*C/N #expected cases inside cluster


output<-list(t=t[1],bf=bf[1],cluster_population=ncluster[1],
cluster_size=length(clustera),expected_cases=expected_cases[1],
observed_cases=xcluster[1],cluster=clustera)
plot1<-plot(x,y,main="Bayesian ZIB Scan - Non-Informative Prior - Incomplete")
pt<-points(x=x[clustera],y=y[clustera],col = "red", pch=21, cex = 2)

return(list(output,plot1, pt))
  }

  else {
    a<-cbind(pop,cases,x,y)
    coord<-cbind(x,y)
    n<-nrow(a)

    N=sum(a[,1]) #Total population
    C=sum(a[,2]) #Total number of cases
    cmax=n
    nmax=0.25*N  #Maximum population allowed inside a cluster

    #Distance Matrix
    dist<-as.matrix(dist(coord,diag=TRUE,upper=TRUE))
    ss<-seq(1,n,1)
    d<-cbind(dist,ss)

    R<-matrix(NA,ncol=n,nrow=n)
    P<-matrix(NA,ncol=n,nrow=n)
    ex_matrix<-matrix(NA,ncol=n,nrow=n)
    matrix_a<-matrix(NA,ncol=n,nrow=n)
    for(j in 1:n){
      d<-d[order(d[,j]),]
      R[,j]<-d[,n+1]
      for (i in 1:n) {
        P[i,j]<-sum(a[R[1:i,j],1])
```

```
        ex_matrix[i,j]<-sum(a[R[1:i,j],1])*C/N
        if(sum(a[R[1:i,j],2])>ex_matrix[i,j] & P[i,j]<nmax  ) {
          matrix_a[i,j]<-1}
          else {  matrix_a[i,j]<-0}
          }
      }

    nreg<-sum(P<n/2)
regions<-which(matrix_a==1,arr.ind=TRUE)


    nz=0
    xz=0
    llr_matrix<-matrix(0,ncol=n,nrow=n)

    xhz_matrix<-llr_matrix
    bfac<-matrix(NA,ncol=n,nrow=n)
    theta<-C/N
    p_1<-0.5
    ph0<-1-p_1
    phz<-p_1/nreg


i0<-which(a[,2]==0)

#pxH0 calculation
iz<-NULL
oz<-R[,1]

izz<-intersect(iz,i0)
ozz<-intersect(oz,i0)

#Gibbs sampling
az<-alpha*sum(a[iz,2])/C
bz<-beta*(sum(a[iz,1])-sum(a[iz,2]))/(N-C)
ao<-alpha*sum(a[oz,2])/C
bo<-beta*(sum(a[oz,1])-sum(a[oz,2]))/(N-C)
delta<-rep(0,n)

delta[i0]<-0.5
nw<-ngibbs
ng<-ngibbs
dmatrix<-matrix(NA,ncol=n,nrow=nw+ng)
p<-rbeta(1,alpha+sum(delta),beta+sum(1-delta))
ad<-matrix(NA,ncol=2,nrow=n)
ad[,1]<-a[,1]*(1-delta)
ad[,2]<-a[,2]*(1-delta)
theta_o<- rbeta(1,sum(ad[oz,2])+ao,
 sum(ad[oz,1])-sum(ad[oz,2])+bo)   #outside region

delta[ozz]<-p/(p+((1-theta_o)^a[ozz,1])*(1-p))

dmatrix[1,]<-delta
for (zz in 2:nrow(dmatrix)){
  delta<-dmatrix[zz-1,]
  p<-rbeta(1,alpha+sum(delta),beta+sum(1-delta))
  ad<-matrix(NA,ncol=2,nrow=n)
```

```
  ad[,1]<-a[,1]*(1-delta)
  ad[,2]<-a[,2]*(1-delta)
  theta_o<- rbeta(1,sum(ad[oz,2])+ao,
  sum(ad[oz,1])-sum(ad[oz,2])+bo)   #outside region
  delta[ozz]<-p/(p+((1-theta_o)^a[ozz,1])*(1-p))
  dmatrix[zz,]<-delta

}

deltaz<-dmatrix[-(1:nw),]
for (z in 1:n){
  delta[z]<-mean(deltaz[,z])}

ad[,1]<-a[,1]*(1-delta)
ad[,2]<-a[,2]*(1-delta)

pxh0<-beta(sum(ad[,2])+alpha, sum(ad[,1])-
sum(ad[,2])+beta)/beta(alpha,beta)

#pxHZ
for(ii in 1:nrow(regions)){
  i<-regions[ii,1]
  j<-regions[ii,2]
  iz<-R[1:i,j]  #index inside
  oz<-R[-(1:i),j] #index outside

  izz<-intersect(iz,i0)
  ozz<-intersect(oz,i0)

  #Gibbs sampling
  az<-alpha*sum(a[iz,2])/C
  bz<-beta*(sum(a[iz,1])-sum(a[iz,2]))/(N-C)
  ao<-alpha*sum(a[oz,2])/C
  bo<-beta*(sum(a[oz,1])-sum(a[oz,2]))/(N-C)
  delta<-rep(0,n)

  delta[i0]<-0.5
  nw<-ngibbs
  ng<-ngibbs
  dmatrix<-matrix(NA,ncol=n,nrow=nw+ng)
  p<-rbeta(1,alpha+sum(delta),beta+sum(1-delta))
  ad<-matrix(NA,ncol=2,nrow=n)
  ad[,1]<-a[,1]*(1-delta)
  ad[,2]<-a[,2]*(1-delta)
  theta_z<-rbeta(1,sum(ad[iz,2])+az, sum(ad[iz,1])-
  sum(ad[iz,2])+bz) #inside region
  theta_o<- rbeta(1,sum(ad[oz,2])+ao, sum(ad[oz,1])-
  sum(ad[oz,2])+bo)   #outside region

  delta[izz]<-p/(p+((1-theta_z)^a[izz,1])*(1-p))
  delta[ozz]<-p/(p+((1-theta_o)^a[ozz,1])*(1-p))

  dmatrix[1,]<-delta
  for (zz in 2:nrow(dmatrix)){
    delta<-dmatrix[zz-1,]
    p<-rbeta(1,alpha+sum(delta),beta+sum(1-delta))
    ad<-matrix(NA,ncol=2,nrow=n)
```

```
    ad[,1]<-a[,1]*(1-delta)
    ad[,2]<-a[,2]*(1-delta)
    theta_z<-rbeta(1,sum(ad[iz,2])+az, sum(ad[iz,1])-
    sum(ad[iz,2])+bz) #inside region
    theta_o<- rbeta(1,sum(ad[oz,2])+ao, sum(ad[oz,1])-
    sum(ad[oz,2])+bo)   #outside region

    delta[izz]<-p/(p+((1-theta_z)^a[izz,1])*(1-p))
    delta[ozz]<-p/(p+((1-theta_o)^a[ozz,1])*(1-p))

    dmatrix[zz,]<-delta

  }

  deltaz<-dmatrix[-(1:nw),]
  for (z in 1:n){
    delta[z]<-mean(deltaz[,z])}

  ad[,1]<-a[,1]*(1-delta)
  ad[,2]<-a[,2]*(1-delta)

  pxhz<-(beta(sum(ad[iz,2])+az, sum(ad[iz,1])-
  sum(ad[iz,2])+bz)/beta(az,bz))*(beta(sum(ad[oz,2])+ao, sum(ad[oz,1])-
  sum(ad[oz,2])+bo)/beta(ao,bo))
  xhz_matrix[i,j]<-pxhz*phz
  bfac[i,j]<-pxhz/pxh0

}
                p_d<-sum(xhz_matrix,na.rm=TRUE)+pxh0*ph0
                llr_matrix<-xhz_matrix/p_d


    t<-max(na.omit(llr_matrix)) #Maximum Posterior Probability
    max<-which(llr_matrix==t, arr.ind=TRUE)
    bf<-bfac[max[1,1],max[1,2]] #Bayes Factor
    clustera<-sort(R[1:max[1,1],max[1,2]]) #cluster

    ncluster<-sum(a[clustera,1]) #population inside cluster
    xcluster<-sum(a[clustera,2]) #cases inside cluster
    expected_cases<-ncluster*C/N #expected cases inside cluster


output<-list(t=t[1],bf=bf[1],cluster_population=ncluster[1],
cluster_size=length(clustera),expected_cases=expected_cases[1],
observed_cases=xcluster[1],cluster=clustera)
plot1<-plot(x,y,main="Bayesian ZIB Scan - Incomplete")
pt<-points(x=x[clustera],y=y[clustera],col = "red", pch=21, cex = 2)

return(list(output,plot1, pt))
 }

}#end
```