Márcia Helena Barbian

Mapeamento de doenças utilizando modelos de mistura com correlação espacial

Belo Horizonte, fevereiro de 2010

Márcia Helena Barbian

Mapeamento de doenças utilizando modelos de mistura com correlação espacial

Dissertação apresentada como requisito parcial
para obtenção de grau de Mestre em Estatística
pela Universidade Federal de Minas Gerais.

Orientador: Prof. Dr. Renato Martins Assunção
Co-Orientador: Marcelo Azevedo Costa

Programa de Pós-Graduação em Estatística
Departamento de Estatística
Instituto de Ciências Exatas
Universidade Federal de Minas Gerais

Belo Horizonte, fevereiro de 2010

Agradecimentos

Agradeço a Deus por tudo que tem me proporcionado e por mais essa conquista.

Aos meus pais Deonisio e Alice, pelo amor, carinho, compreensão e principalmente pelo apoio e incentivo nos momentos mais difíceis que enfrentei, se fazendo sempre presentes mesmo eu estando longe de casa. Aos meus irmãos, Eduardo e Jackson, pelo apoio e motivação, além de toda compreensão e paciência ao ouvir as minhas lamentações.

Ao meu orientador, Professor Renato Assunção, pelo apoio, pelas explicações sempre claras e repletas de exemplos esclarecedores, além de toda paciência e compreensão. Ao meu co-orientador Professor Marcelo Azevedo Costa, pela ajuda na realização desse trabalho.

Aos professores do curso de mestrado em estatística, pelo conhecimento transmitido. Ao meu orientador na graduação, Professor Flávio, por me incentivar a fazer uma pós-graduação.

Aos membros da banca examinadora, Prof.ª Rosangela Loschi(UFMG) e Prof. Ronaldo Dias (UNICAMP), pela leitura, correções e sugestões da dissertação.

À CAPES pela bolsa de mestrado, à FAPEMIG por diversos apoios financeiros prestados para participação em eventos.

Aos amigos Markus e Rodrigo, que me incentivaram a fazer o mestrado na UFMG, pela convivência e por tudo que aprendi com eles, foram a minha família em Minas Gerais. Ao Fábio, Max, Carlito, Michele, Maristela e Ronaldo pela ótima recepção que tive ao chegar em Belo Horizonte. Aos colegas de mestrado e amigos que fiz aqui em Belo Horizonte, em especial a Letícia, Jacque, Thaís e Aline, que me ajudaram MUITO quando eu estava com o pé quebrado, bah gurias brigadão. A Érica pela paciência em ouvir as minhas reclamações. À Isabel e Grazi pelas risadas. Agradecer a todos os meus amigos do LESTE. Aos meus amigos de Porto Alegre que mesmo de longe me apoiaram muito.E a todos que de uma forma direta ou indireta contribuíram para a realização deste trabalho. Muito Obrigada!

Resumo

Uma área de estudo em bioestatística e de interesse epidemiológico é o mapeamento de doenças. O objetivo de mapear dados de determinada patologia é detectar áreas de risco relativo elevado ou reduzido. Uma maneira muito simples de estimar o risco relativo de uma região geográfica, dada a suposição de independência entre as contagens de eventos das áreas envolvidas, é a Taxa de Mortalidade Padronizada. Todavia, esse estimador possui grande variabilidade, principalmente no caso de doenças raras e em locais com pequenas populações. Uma solução para este problema é o uso de modelos de suavização do risco relativo estimado. Nesse trabalho, será abordado um método semiparamétrico que utiliza campos aleatórios markovianos ocultos. A função *a priori* assume um modelo de mistura correlacionado espacialmente. Utiliza-se o algoritmo de Monte Carlo via Cadeias de Markov com saltos reversíveis para obter-se aproximações para as distribuições *a posteriori* dos parâmetros. Como ilustração da metodologia estudada, foi analisada a taxa de mortalidade por câncer de traquéia, brônquios e pulmões nos estados de São Paulo, Paraná, Santa Catarina e Rio Grande do Sul no ano de 2007. Além disso, procedeu-se a simulação de dados, para avaliar o desempenho do modelo e para compará-lo com a metodologia paramétrica comumente aplicada.

Palavras-chaves: *Campos Aleatórios de Markov, Mapeamento de doença, MCMC de saltos reversíveis, Modelo de Potts, Modelos de Mistura, Semiparamétrico.*

Abstract

One interesting area of study in biostatistics and epidemiological is the mapping of diseases. The objective of mapping diseases is to detect areas of high or low relative risk. A very simple way to estimate the relative risk of a geographic region, given the assumption of independence between the event counts of the involved areas is the Standardized Mortality Rate. However, this estimator has high variability, especially for rare diseases and in places with small populations. A solution to this problem is the use of models smooth of the relative risk estimate. In this work, we will describe a method that uses semiparametric hidden Markov random fields. The prior function assumes a spatially correlated mixture model. We use the algorithm of reversible jump Markov Chain Monte Carlo in order to obtain approximations for posterior distributions of parameters. As an illustration of the methodology study, we analyzed the mortality rate for cancer of the trachea, bronchi and lungs in the states of Sao Paulo, Parana, Santa Catarina and Rio Grande do Sul in 2007. Moreover, we analyze the performance of the model and compare it with parametric methodology commonly applied through a collection of synthetic dataset.

Keywords: Disease mapping, Markov Random Fields, Mixture Models, Potts Model, Reversible Jump Markov Chain Monte Carlo, Semiparametric.

Sumário

Lista de Abreviaturas

RR	Risco Relativo.
TMP	Taxa de Mortalidade Padronizada (*Standardized Mortality Rate*).
SMR	*Standardized Mortality Rate*.
CAR	*Conditional AutoRegressive*.
UMVU	Não-Viesados e de Variância Uniformemente Mínima.
CAM	Campo Aleatório de Markov.
CAMG	Campo Aleatório de Markov Gaussiano (*Gaussian Markov Random Fields*).
GMRF	*Gaussian Markov Random Fields*.
ICAR	CAR impróprio.
MIX	Modelo de Mistura com Correlação Espacial.
DIC	*Deviance Information Criterion*.
CPO	*Conditional Predictive Ordinate*.
RAMSE	*Root Average of Mean Squared Error*.
RAMSEL	*Root Average of Mean Squared Error Logarithm*.

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Um interessante problema epidemiológico é a análise da variação geográfica em taxas de incidência ou de mortalidade de determinada moléstia. Essa análise é comumente realizada através do mapeamento de doenças. Esses mapas são instrumentos valiosos para apontar associações entre fontes potenciais de contaminação e áreas de risco elevado, sugerir determinantes locais de doenças e visualizar a distribuição geográfica da mesma.

Uma maneira simples de mapeamento de determinada patologia consiste em mapear o estimador de máxima verossimilhança do risco relativo (RR), sobre as diferentes regiões geográficas. Porém, quando a doença é rara ou a populacão é muito pequena, essas estimativas acabam causando distorções na visualização do mapa ao apresentar estimativas com valores extremamente altos ou baixos. Além disso, esse tipo de estimativa não leva em consideração a possível dependência espacial entre as áreas, que pode estar presente em algumas situações.

Para superar tais dificuldades, métodos hierárquicos Bayesianos têm sido propostos na literatura (ver Banerjee, Carlin e Gelfand (2004)). Estes métodos, além de suavizar as estimativas do risco relativo, fornecem medidas sobre a incerteza das mesmas. A estimativa é suavizada porque, para estimar o risco de uma área, utiliza-se informações das outras áreas que compõem a região de estudo.

Admitindo que há interesse em mapear o RR de uma determinada doença, a análise deve levar em conta todas as covariáveis relevantes. Entretanto, é pouco provável que todos os fatores que influenciam a patologia possam ser identificados ou mensurados por unidade de área. Nesse contexto, frequentemente, ainda permanece uma heterogeneidade espacial não explicada totalmente pelas covariáveis disponíveis. Essa associação espacial é introduzida no modelo através da distribuição *a priori* do risco relativo λ.

O método paramétrico mais utilizado para modelar a superfície do RR, introduzido por Besag (1974), é o modelo CAR (*Conditional AutoRegressive*). Esse modelo é especificado por um conjunto de distribuições condicionais, e propõe que a distribuição do efeito aleatório de uma área tem distribuição gaussiana, com o parâmetro de locação dado pela média ponderada dos efeitos das áreas vizinhas e variância inversamente proporcional ao número de áreas vizinhas. Embora imensamente popular, o modelo CAR sofre algumas críticas. Por exemplo, segundo Best, Richardson e Thomson (2005) o modelo CAR tende a super-suavizar a superfície do risco, pois é estimado globalmente.

Como alternativa surgiram os métodos semiparamétricos, que modelam o risco relativo conforme partições. Isto é, o mapa é dividido em grupos conforme o valor do RR. O número, o risco relativo e a classificação de cada área nos grupos são parâmetros desconhecidos. A vantagem em relação ao modelo paramétrico é essa adaptabilidade, pois áreas que possuem baixo risco não irão influenciar as áreas de alto risco, já que estarão em grupos diferentes que são independentes entre si. Algumas metodologias semiparamétricas foram propostas por: Green e Richardson (2002), Denison e Holmes (2001) e Knorr-Held e Raβer (2000).

Dentre as diferentes maneiras de estimar o risco semiparametricamente, o modelo proposto por Green e Richardson (2002) representa a dependência espacial sob a forma de um campo aleatório de Markov. Ou seja, a estimativa do risco de uma área vai depender somente das áreas vizinhas. Essa propriedade é muito conhecida na análise espacial e será descrita no capítulo 4.

1.1 Objetivos

O objetivo desta dissertação será a implementação do método proposto por Green e Richardson (2002) e através de simulações e análise de dados reais, comparar seu desempenho com o modelo paramétrico CAR comumente utilizado.

1.2 Organização do Trabalho

O restante deste texto está organizado da seguinte forma. O próximo capítulo fará uma breve descrição sobre mapeamento de doenças. O capítulo 3 abordará o modelo de mistura com correlação espacial. No capítulo 4 uma breve introdução sobre o modelo CAR será apresentada. O capítulo 5 mostrará estudos de simulação. O capítulo 6 ilustrará a metodologia através de dados da taxa de mortalidade por câncer de traquéia, brônquios e pulmões nos estados de São Paulo, Paraná, Santa Catarina e Rio Grande do Sul. No capítulo 7 serão mostradas as conclusões e algumas considerações

finais.

Capítulo 2

Mapeamento de Doenças

O estudo da variação espacial de determinada doença tem atraído grande interesse nos últimos anos por várias razões, como a crescente disponibilidade de dados e a constatação de que fatores espaciais podem ser importantes na saúde de uma população. Atualmente, por mais que a análise estatística seja sofisticada, o mapa continua a ser um importante instrumento descritivo. O mapeamento do risco se tornou uma ferramenta muito utilizada na compreensão de fatores que influenciam na incidência de determinada moléstia.

Considere a seguinte notação:

y_i: n° observado de casos da doença na área i, $i = 1, 2, \ldots, N$.

E_i: n° de casos esperados da doença na área i.

n_i: n° de pessoas em risco na área i.

Os casos Y_i são considerados variáveis aleatórias, enquanto E_i são fixos e calculados da seguinte forma:

$$E_i = n_i \left(\frac{\sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} n_i} \right) = n_i \overline{r}. \tag{2.1}$$

Dessa forma, $\overline{r}$ é a taxa de incidência global da doença em toda a região em estudo. O valor E_i é o número esperado de casos na área i caso a taxa de incidência da doença nessa área fosse igual à taxa de incidência global $\overline{r}$.

Embora a equação 2.1 considere a taxa global dependente somente do número total de habitantes e casos observados, a maioria das doenças afeta pessoas de diferentes idades desproporcionalmente. Como ilustração, consideremos o câncer de pulmão: quanto maior a idade, mais alta será a probabilidade de uma pessoa possuir a doença. Consequentemente, populações contendo um maior

número de pessoas com idade avançada terão maior incidência de doentes do que em populações mais jovens.

Quando realizamos o mapeamento de determinada doença estamos interessados no padrão espacial do risco relativo nas diferentes áreas, fazendo-se necessário remover os efeitos de fatores de risco conhecidos (como idade e sexo), que podem influenciar a incidência de casos em determinada área. Por exemplo, se uma área i possui uma proporção de doentes superior à área i', essa diferença pode ser devida às suas diferentes distribuições de idade e não devido à heterogeneidade espacial do risco. Uma maneira de remover os efeitos de fatores de risco conhecidos é utilizar a taxa padronizada, na qual o risco é estratificado por sexo, idade ou demais fatores, o que torna possível a comparação de taxas de diferentes populações. Como exemplo de aplicação, suponha que estamos estratificando a população por sexo e j grupos de idade. O valor de E_i é definido como:

$$E_i = \sum_j n_{ij} r_j,$$

onde n_{ij} é o número de indivíduos em risco da área i e grupo de idade j do sexo feminino (ou masculino) e r_j é a taxa da doença no grupo de idade j e sexo feminino (ou masculino). Para maiores detalhes sobre taxas padronizadas ver Waller e Gotway (2004). Estes valores E_i correspondem à hipótese nula de que a taxa r_j específica por idade e sexo em cada estrato é constante em todas as áreas.

No mapeamento de doenças, o objetivo é estudar a variação espacial do risco, assumindo que ele não é constante no mapa. Se a doença é rara, o modelo usual para modelar a variável aleatória de incidência da doença Y_i, na área i, é o de Poisson:

$$Y_i|\lambda_i \sim \text{Poisson}(E_i\lambda_i) \tag{2.2}$$

em que λ_i é o risco relativo da doença na região i. Condicionado nos valores $\lambda_1, \ldots, \lambda_N$, assume-se que as contagens Y_i são independentes. Esse modelo pode ser facilmente modificado, introduzindo dependência do risco em covariáveis x_{il}, com coeficientes γ_l, medidas em cada área i:

$$Y_i|\lambda_i \sim \text{Poisson}(\lambda_i e^{\sum_l x_{il}\gamma_l} E_i). \tag{2.3}$$

Uma forma simples de mapeamento no caso do modelo 2.2 é a taxa de mortalidade padronizada (TMP), obtida através do estimador de máxima verossimilhança do risco relativo, assumindo independência entre as variáveis aleatórias Y_i. A TMP é a razão dos casos observados pelos esperados ($\text{TMP}_i = y_i/E_i$). Esta estimativa é não-viciada e uniformemente de mínima variância (UMVU) e, portanto, ela é ótima nesse sentido.

Entretanto, essa estimativa possui grande variância se o número esperado E_i de eventos ou a população for pequena. De fato, temos:

$$Var(\text{TMP}_i|\lambda_i) = \frac{Var(Y_i|\lambda_i)}{E_i^2} = \frac{\lambda_i}{E_i}.$$

Na maioria das áreas o valor do risco relativo não se afasta muito de um. Entretanto, se E_i for próximo de zero, a estimativa TMP_i terá variância muito grande. Portanto, os maiores valores de TMP_i tendem a ser observados nas áreas com populações pequenas, e não nas áreas com riscos reais observados. As maiores oscilações das estimativas TMP_i não estarão associadas com a variação real do risco relativo λ_i, mas sim com flutuações aleatórias. Além disso, a TMP_i considera que os Y_i's são independentes, mas em alguns casos, uma parte desconhecida da variação do RR pode ser causada por fatores não observados, dependentes geograficamente. Diante disso, uma das alternativas comumente empregadas é a utilização de modelos hierárquicos para suavizar essa estimativa ao longo do mapa.

Formalmente, tais modelos hierárquicos podem ser descritos da seguinte forma:

$$f(\boldsymbol{\lambda}|y_1, ..., y_N) \propto L(y_1, ..., y_N|\boldsymbol{\lambda})p(\boldsymbol{\lambda})$$

em que $L(y_1, ..., y_n|\boldsymbol{\lambda})$ é a função de verossimilhança e $p(\boldsymbol{\lambda})$ é a distribuição *a priori* do vetor de parâmetros $(\lambda_1, ..., \lambda_N)$. Condicionalmente em $\lambda_1, ..., \lambda_N$, os valores $Y_1, ..., Y_N$ são supostos independentes com distribuição de Poisson com média $\lambda_i E_i$, (equação 2.2). A modelagem da distribuição *a priori* $p(\boldsymbol{\lambda})$ permite introduzir dependência espacial entre os riscos, de modo que regiões próximas tendem a ter riscos semelhantes. O que se faz, então, é modelar o risco relativo a partir de um modelo de efeitos aleatórios. Sob hipótese de independência espacial, poderíamos dizer que os efeitos aleatórios $\lambda_i's$ são i.i.d.. Porém, quando existe uma associação espacial entre as áreas,

a distribuição dos $\lambda_i's$ deve refletir isso. A grande questão aqui é como definir uma distribuição *a priori* que capte essa estrutura espacial de maneira adequada.

Diversos autores têm realizado estudos sobre distribuições *a priori* que agreguem a estrutura espacial ao RR. Por exemplo, Besag (1974) apresenta o modelo paramétrico CAR e anos depois Besag, Mellié e York. (1991) apresentam o ICAR; entre os modelos semiparamétricos podemos citar Green e Richardson (2002) com modelos de mistura, Denison e Holmes (2001) e Knorr-Held e Raβer (2000) que abordam a utilização de modelos de partição espacial.

Antes de prosseguirmos, a próxima seção apresentará a definição da matriz de vizinhança **W**. A matriz **W** fornece o mecanismo para introduzir a estrutura espacial necessária para a análise dos dados, e deve ser especificada antes de realizar a análise espacial, independentemente da metodologia.

2.1 Matriz de Vizinhança W

Dadas as unidades de área $1, 2, \ldots, n$, o elemento w_{ij} da matriz **W**, de dimensão $n \times n$, representa o peso, ou a intensidade da proximidade espacial entre as áreas i e j. É importante salientar que a diagonal da matriz de vizinhança é nula ($w_{ii} = 0$), independentemente da definição de vizinhança, pois uma área não pode ser vizinha de si mesma. Há várias formas de definir os elementos w_{ij}. Alguns exemplos para definição de w_{ij}são os seguintes:

- Suponha que $w_{ij} = 1$, para todo $i \neq j$, se o centróide da área i está a menos de 300 quilômetros da área j. Caso contrário, $w_{ij} = 0$.

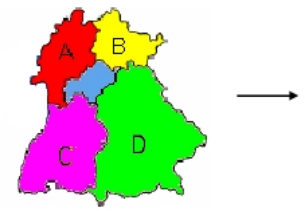
- Outra opção seria considerar graus intermediários de vizinhança. Seja d_{ij} a distância em quilômetros entre os centróides da área i e j. O elemento $w_{ij} = 1/(1 + d_{ij})$, quanto menor a distância entre os centróides mais próximo de um o elemento w_{ij} estaria. Ao contrário, quanto maior for a distância, mais w_{ij} tenderia à zero.

- Outra definição é que $w_{ij} = 1$ se as áreas i e j compartilham fronteiras e $w_{ij} = 0$, caso contrário.

Este último exemplo é uma das formas mais utilizadas para definição da matriz **W**. A escolha da estrutura da matriz **W** não possui regras estabelecidas, sendo uma decisão do pesquisador. No restante dessa dissertação utilizaremos uma das opções mais simples para definição de vizinhança,

em que os elementos w_{ij} da matriz $\mathbf{W}$ possuem valor um nas áreas que dividem fronteira e zero caso contrário. Para maiores detalhes sobre estruturas para matrizes de vizinhança consultar Assunção (2009).

O exemplo exposto a seguir mostra a definição da matriz $\mathbf{W}$ utilizada no restante da dissertação. O mapa selecionado para esse exemplo possui uma estrutura bem simples, com somente quatro áreas (A,B,C e D) e um l



$$\mathbf{W} = \begin{array}{c} \begin{matrix} A & B & C & D \end{matrix} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$$

Figura 2.1: Exemplo de um mapa fictício e a sua matriz $\mathbf{W}$.

Outra maneira de representar a matriz de vizinhança e a disposição das áreas no mapa é através de um grafo, em que os vértices são as áreas e a estrutura de vizinhança são as arestas. As áreas com $w_{ij} > 0$ são conectadas por uma aresta. Na figura 2.2 temos a representação do mapa e da matriz de vizinhança da figura 2.1

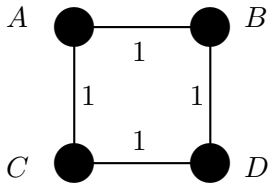


Figura 2.2: Figura 2.1 representada como um grafo.

Capítulo 3

Modelos de Mistura Correlacionados Espacialmente

Best, Richardson e Thomson (2005) afirmam que os modelos paramétricos tendem a super-suavizar a superfície do risco. Esse comportamento acontece porque a suavização é afetada globalmente por todas as áreas. Portanto, regiões de alto risco são influenciadas por áreas de menor risco de todo o mapa. Esse tipo de problema levou muitos autores a desenvolver modelos espaciais semiparamétricos, que distribuem a variação espacial através de modelos de partição. Cada componente da partição tem um risco relativo desconhecido constante, o que permite descontinuidades na superfície do risco. Exemplos de tais modelos semiparamétricos foram propostos por Knorr-Held e Raβer (2000), Denison e Holmes (2001) e Green e Richardson (2002). É este último trabalho que vamos abordar nesta dissertação.

A idéia básica do modelo de Green e Richardson (2002) é considerar que existe uma partição das áreas do mapa em um pequeno número k de subgrupos, também chamados de componentes. Cada componente pode ser formado por áreas que não são adjacentes, e o que os distingue é o valor do seu risco relativo. Subgrupos diferentes possuem riscos diferentes. A figura (3.1), apresenta uma possível partição do mapa em seis componentes, cada um deles com uma cor. Note que um componente pode ser composto por regiões disjuntas.

O método utiliza modelos de mistura, pois considera o risco de cada componente, partilhado pelas áreas de um mesmo subgrupo, como sendo um valor aleatório selecionado a partir de uma distribuição de probabilidade. Se o RR entre todas as áreas i for homôgeneo espacialmente, isto é, se há diferença entre os riscos das diferentes áreas, então a distribuição é a mesma para todas as áreas e o número de componentes é igual a um. Mas, se há alguma heterogeneidade espacial no RR, em que as áreas i proveêm de diferentes funções, haverá uma mistura de distribuições e o

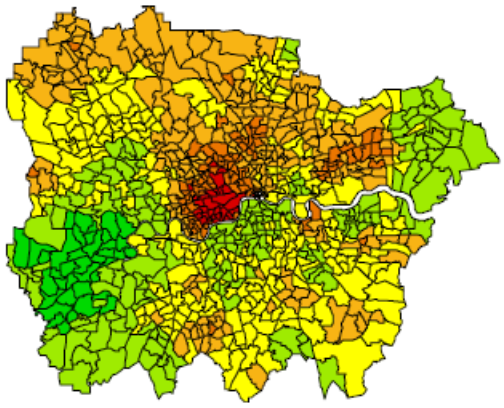


Figura 3.1: Exemplo de estimação da superfície do risco usando modelo de mistura.

número de componentes será maior que um.

Para deixar mais clara a idéia do método, observe a figura 3.2. Imagine que estamos fazendo o mapeamento do risco de determinada doença no litoral do Rio Grande do Sul, por isso as áreas são colocadas de modo sequencial no mapa. A área 1 é vizinha da área 2, a área 2 é vizinha das áreas 1 e 3, e assim sucessivamente. Existem três componentes nesta figura. O primeiro é formado pelas áreas 1 à 4. O segundo componente é composto pelas áreas de 5 à 8 e também pelas áreas 15 à 22. Portanto, esse componente é formado por grupos de áreas não adjacentes. O terceiro componente é formado pelas áreas de 9 à 14. Existem três riscos relativos, um para cada componente, e eles estão representados pelos seguimentos de reta horizontais. A altura vertical do segmento indica o valor do RR da área. Observe que as áreas de um mesmo componente possuem o mesmo valor para o seu risco relativo.

A escolha do número k de componentes e a alocação de cada área a um componente é feita através de um modelo de mistura. Esse modelo de mistura considera que áreas vizinhas tendem a pertencer a um mesmo componente. Esta indução de similaridade espacial é feita através do modelo de Potts, descrito na seção 3.2. Para maiores informações teóricas sobre modelo de Potts pode-se citar Beaudin (2009).

3.1 Modelo de Mistura com Alocações Dependendo Espacialmente

Visto que o interesse está em modelar a superfície do risco de doenças raras, a maneira mais comum de fazer tal análise é através de métodos hierárquicos utilizando a distribuição de Poisson no primeiro nível da hierarquia (ver equação 2.2). A estrutura da distribuição conjunta do risco

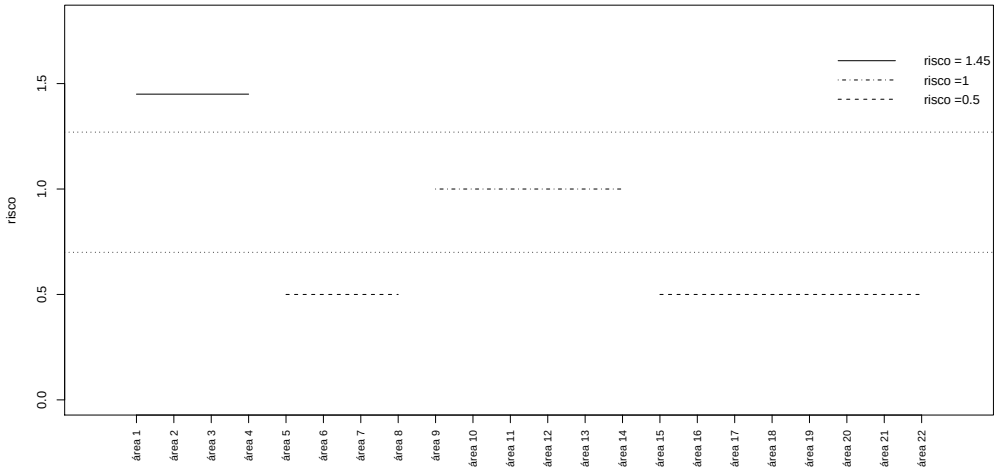


Figura 3.2: Exemplo da divisão de áreas em componentes, segundo o risco estimado.

$\{\lambda_i, i = 1, 2, \ldots, N\}$ é particionada de acordo com os componentes, de forma que todas as áreas de um mesmo componente têm um único risco relativo. A classificação das áreas nos componentes é feita através do modelo de Potts, muito usado em processamento de imagens.

Suponha que existam k componentes. Existem então k valores distintos para os λ_i's. Seja z_i a variável de alocação da área i ao seu componente. Isto é, $z_i = j$ indica que a área i tem RR λ_j, onde j é um dos possíveis valores $1, 2, \ldots, k$. Essa alocação é feita pelo modelo de Potts e, dessa forma, $\lambda_i = \lambda_{z_i}$. O modelo hierárquico procede assumindo uma distribuição *a priori* para os modelos, isto é, para o número de componentes. É comum usar uma distribuição Uniforme$\{1, 2, \ldots, k_{\max}\}$, onde $k_{\max}$ é um valor especificado pelo usuário.

É importante salientar que o custo computacional da metodologia aumenta consideravelmente quanto maior for $k_{\max}$. Em vista disso, não é aconselhável utilizar um valor de $k_{\max}$ muito elevado, tal como o número de áreas do mapa. Além disso, quando fazemos o mapeamento de determinada doença, espera-se que a distribuição do risco subjacente não seja muito inconstante. Logo, se supõem que toda a variabilidade do RR possa ser explicada por um pequeno número de componentes.

Quantos grupos são necessários para especificar toda a variabilidade do RR? Não há uma resposta padrão para essa pergunta, pois depende da variabilidade dos dados a serem analisados. A maneira encontrada para a determinação de $k_{\max}$ é através de simulações, em que analisa-se os dados com um valor de $k_{\max}$ supostamente suficiente para explicar a variabilidade do risco. Se o

resultado da simulação indicar que $k_{\max}$ não é um valor provável para o número de componentes do mapa, então possivelmente a variabilidade do RR pode ser explicada por um número de componentes inferior a $k_{\max}$. Por outro lado, se a proporção de $k_{\max}$ é alta, há indícios de que seja necessário especificar um valor maior para o limite de k, já que o número máximo de componentes não parece representar toda a heterogeneidade do risco. Nesse caso, uma nova simulação deve ser realizada com um valor de $k_{\max}$ superior ao anterior.

No restante da dissertação utilizaremos $k = 10$ como valor inicial para $k_{\max}$, visto que este é o valor utilizado nos estudos de Green e Richardson (2002) e Best, Richardson e Thomson (2005).

Os distintos riscos relativos $\lambda_1, \lambda_2, \ldots, \lambda_k$ são escolhidos independentemente a partir de uma distribuição Gama:

$$\lambda_j \sim \Gamma(\alpha, \beta) \tag{3.1}$$

onde $\alpha = 1$ e $\beta = \sum_i E_i / \sum_i y_i$. Pode-se utilizar *hiperprioris* para os hiperparâmetros α e β. Nessa dissertação eles serão considerados fixos. Para assegurar identificabilidade, os componentes da mistura serão classificados em ordem crescente, para maiores detalhes sobre identificabilidade ver Richardson e Green (1997). Logo, a distribuição *a priori* conjunta para $\boldsymbol{\lambda}$ é

$$p(\boldsymbol{\lambda}|k, \alpha, \beta) = k! I[\lambda_1 < \lambda_2 < \ldots < \lambda_k] \prod_{j=1}^{k} \frac{\beta^\alpha \lambda_j^{\alpha-1} e^{-\beta\lambda_j}}{\Gamma(\alpha)} \tag{3.2}$$

em que I é indicadora do conjunto de riscos ordenados crescentemente e $k!$ são todas as possíveis maneiras de dispor os λ's.

3.2 Modelo de Potts

Cientistas e matemáticos usam o modelo de Potts para estudar e predizer resultados estocásticos de sistemas complexos. Por essa razão, esse modelo é muito conhecido na área de mecânica estatística. A distribuição de probabilidade do modelo de Potts possui a seguinte especificação:

$$p(\boldsymbol{z}|\psi) = e^{\psi U(\boldsymbol{z}) - \theta_k(\psi)}, \tag{3.3}$$

$$U(\boldsymbol{z}) = \sum_{i \sim i'} I[z_i = z_{i'}], \tag{3.4}$$

e

$$\theta_k(\psi) = log\left(\sum_{\boldsymbol{z}\in\{1,2,\ldots,k\}^n} e^{\psi U(\boldsymbol{z})}\right), \tag{3.5}$$

em que z_i é a variável que indica a qual componente pertence a área i, $\boldsymbol{z}$ é o vetor composto pelos z_i's que representa a partição do mapa, $i \sim i'$ indica que as áreas i e i' são vizinhas e ψ é o parâmetro que mede o grau de dependência espacial entre as áreas. Se ψ=0 as áreas serão independentes e quanto maior o valor de ψ, mais alta a probabilidade de que áreas vizinhas sejam classificadas no mesmo componente.

A equação (3.4) calcula a quantidade de vizinhos da área i que estão no mesmo componente. Isto é, se duas áreas vizinhas estão alocadas no mesmo componente, a indicadora $I[z_i = z_{i'}]$ é igual a um. Dado que o mapa está particionado nos diferentes componentes, calcula-se o valor dessa indicadora para todos os pares de áreas vizinhas.

A constante $\theta_k(\psi)$, equação 3.5, é a constante de normalização da densidade 3.3, também chamada de função de partição. Antes de proceder à análise, utilizando o modelo de Potts, é necessário calcular essa função de partição. Ela assume diferentes valores, dependentes dos parâmetros k e ψ escolhidos. Se $k \in \{1, 2, 3, 4\}$ e $\psi \in \{0, 0.5, 1\}$, deve-se calcular a função de partição doze vezes, uma para cada combinação de k e ψ (ver mais detalhes na seção 3.2.1. O problema é que o custo computacional desse cálculo é alto. Então, para otimizar o processamento do método e, além disso, por acreditar que a discretização não tenha muito impacto sobre as estimativas, a distribuição *a priori* de ψ terá uma distribuição uniforme discreta no intervalo $\{0; 0,1; \ldots; \psi_{\max}\}$.

O valor limite de ψ depende do mapa e do problema em questão. Como exemplo, Green e Richardson (2002) citam um estudo de simulação para escolher $\psi_{\max}$. Para $\psi = 1,0$ e $k = 2$ a probabilidade de que duas áreas adjacentes sejam classificadas no mesmo componente é de 0,96, declinando para 0,7 quando $k = 8$. Isso indica que $\psi = 1$ implica em grande correlação espacial, forçando áreas vizinhas a estarem no mesmo componente. Logo, nesse caso, não é necessário $\psi_{\max} > 1$.

A distribuição *a priori* correspondente à formulação do modelo de Potts é composta pelas seguintes variáveis:

- $\{\lambda_1, \ldots, \lambda_k\}$ representam os riscos relativos dos k componentes do modelo. O risco relativo

λ_j é independente dos demais e possui distribuição Gama. A distribuição *priori* conjunta é dada pela equação 3.2.

- k é a variável que representa o número de componentes do modelo. Sua distribuição *a priori* é $p(M_k) \sim \text{Uniforme}\{1, 2, \ldots, k_{\max}\}$, M_k é o modelo com k componentes.
- ψ é a variável que controla o grau de dependência espacial. Sua distribuição *a priori* é $p(\psi) \sim \text{Uniforme}\{0; 0.1; \ldots; \psi_{\max}\}$.
- $\boldsymbol{z}$ representa a particular partição do mapa. É o vetor $\{z_1, z_2, \ldots, z_N\}$ que indica como as áreas do mapa estão divididas entre os componentes. A distribuição *a priori* é a equação 3.3.

Logo, a distribuição *a posteriori* conjunta é:

$$p(\boldsymbol{\lambda}, k, \boldsymbol{z}, \psi | y, \alpha, \beta) \propto p(M_k)p(\psi)p(\boldsymbol{\lambda}|k, \alpha, \beta)p(\boldsymbol{z}|k, \psi)p(\boldsymbol{y}|\boldsymbol{\lambda}, \boldsymbol{z}).$$

3.2.1 Aproximação da função de partição para o modelo de Potts

No modelo de Potts é necessário calcular uma constante de normalização, dada pela equação (3.5). Esta constante leva em conta todas as partições possíveis do mapa. Se o mapa possui k componentes e n áreas, existem k^n diferentes partições.

Como ilustração, iremos calcular a constante de normalização do mapa representado na figura 2.1. O número escolhido de componentes é igual a dois, em que a cor branca representa o componente um e a cor preta o componente dois. Se o vértice do grafo for da cor branca, indica que a área, representada pelo vértice, faz parte do componente um, e se for da cor preta, a área está no componente dois. Como o mapa possui quatro áreas e dois componentes há 2^4 possíveis partições. A figura 3.3 reproduz todas as 16 configurações possíveis. Note que a aresta possui valor um quando as áreas conectadas fazem parte do mesmo componente e valor zero, se os vértices possuem cores diferentes. A soma dos valores associados às arestas é a $U(\boldsymbol{z})$.

Para calcular a constante de normalização deve-se calcular a equação 3.4 para todas as configurações ilustradas na figura 3.3. Como exemplo, iremos demonstrar como a função $U(\boldsymbol{z})$ é calculada para a partição representada pela figura 3.4. Nota-se que no grafo abaixo, há dois vértices da cor preta e dois da cor branca, indicando que duas áreas fazem parte do componente um e duas do componente dois.

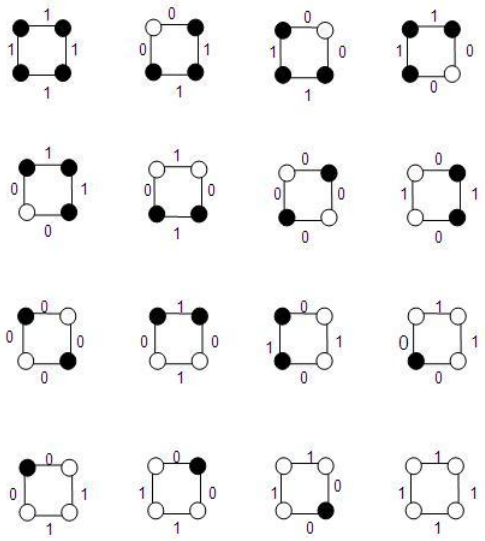


Figura 3.3: Possíveis configurações do mapa da figura 2.1.

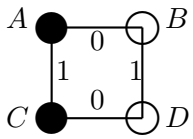


Figura 3.4: Exemplo de configuração.

$$\begin{aligned} U(\boldsymbol{z}) &= I_{A\sim B}[z_A = z_B] + I_{A\sim C}[z_A = z_C] + I_{B\sim D}[z_B = z_D] + I_{C\sim D}[z_C = z_D] \\ &= I_{A\sim B}[2 = 1] + I_{A\sim C}[2 = 2] + I_{B\sim D}[1 = 1] + I_{C\sim D}[2 = 1] \\ &= 0 + 1 + 1 + 0 = 2 \end{aligned}$$

Após calcular a função $U(\boldsymbol{z})$ para as demais partições, temos que o valor da função de partição para o mapa é:

$$\begin{aligned} \theta_2(\psi) &= ln\left(e^{4\psi} + e^{2\psi} + e^{2\psi} + \ldots + e^{2\psi} + e^{4\psi}\right) \\ &= ln\left(12e^{2\psi} + 2e^{4\psi} + 2e^{0\psi}\right). \end{aligned}$$

Quando o mapa possui poucas áreas e grupos é possível calcular a constante analiticamente, como no exemplo acima. Mas, em geral, estaremos tratando de reticulados mais complexos, em que k^n é um valor extremamente grande, tornando inviável listar todas as possibilidades.

Dado esse número enorme de configurações, a probabilidade de uma dessas partições acontecer

é próxima de zero. Contudo, o interesse está na característica média que um sistema exibe ao longo do tempo. Uma maneira obter $\theta_k(\psi)$ é através de MCMC.

Esse método, segundo Gelman e Meng (1998), tem sido utilizado há muito tempo em estatística-física. Dado o modelo de Potts com k componentes e n áreas, derivando $\theta_k(\psi)$ com relação à ψ nós obtemos:

$$\begin{aligned}\frac{\partial}{\partial\psi}\theta_k(\psi) &= \frac{\partial}{\partial\psi}\log\sum_{\boldsymbol{z}\in Z} e^{\psi U(\boldsymbol{z})} \\ &= \sum_{\boldsymbol{z}\in Z} U(\boldsymbol{z})p(\boldsymbol{z}|\psi) \\ &= E(U(\boldsymbol{z})|\psi,k)\end{aligned}$$

a esperança de $U(z)$ quando z é distribuído de acordo com o modelo de Potts. Além disso, $Z = \{1,2,\ldots,k\}^n$ é o conjunto de todas as possíveis classificações, mas $\theta_k(0) = \log\sum_{\boldsymbol{z}\in Z} 1 = n\log k$, e então

$$\theta_k(\psi) = n\log k + \int_0^{\psi} E(U|\psi',k)\mathrm{d}\psi'.$$

Nesse método de estimação de $\theta_k(\psi)$, a esperança é substituída pela média amostral através de uma simulação MCMC. A integral é calculada numericamente.

3.3 MCMC

Como nenhum dos parâmetros possui uma distribuição *a posteriori* analiticamente fechada é necessário obter aproximações para as distribuições *a posteriori* dos parâmetros através de métodos MCMC, sendo que três deles são cadeias com movimentos de dimensão fixa e um com dimensão variável. Movimentos de dimensão fixa são utilizados quando a dimensão do vetor de parâmetros é conhecida. Isto é, sabemos *a priori* o número de parâmetros a serem estimados. Nesse caso, métodos usuais de simulação como Gibbs e Metropolis, podem ser utilizados. Entretanto, quando não conhecemos o tamanho do vetor de parâmetros, não conhemos a dimensão da cadeia de Markov e o algoritmo de MCMC usual não pode ser empregado. Nesses casos, devemos usar uma técnica que salta entre espaços de diferentes dimensões, o MCMC de saltos reversíveis, proposto por Green (1995).

3.3.1 Dimensão Fixa

As três variáveis de dimensão fixa são: o parâmetro de interação espacial ψ, a variável de alocação z_i e o risco dos diferentes componentes λ_j. A variável ψ é estimada através do Metropolis *walk* (Gilks, Richardson e Spiegelhalter (1996)), propondo perturbações de $\pm 0,1$ com igual probabilidade. A sua distribuição condicional completa é:

$$p(\psi|\ldots) \propto p(\psi)e^{\psi U(\boldsymbol{z})-\theta_k(\psi)}.$$

A variável de alocação é atualizada através do amostrador de Gibbs (Robert e Casella (1999)). A condicional completa tem distribuição multinomial, sendo que a probabilidade de cada componente é:

$$p(z_i = j|\ldots) \propto e^{\lambda_j E_i}\lambda_j^{y_i}e^{\psi n_{ij}}$$

O risco é estimado através de Metropolis-Hastings (Robert e Casella (1999)), uma aproximação para atualizar simultaneamente as distribuições do $\boldsymbol{\lambda}$ é somar valores de uma distribuição de média zero para cada log de λ_j, os valores são então substituídos em ordem crescente. A probabilidade de aceitação formada pela verossimilhança, a distribuição *a priori* e o jacobiano da transformação log se reduz a

$$\min\left\{1, \prod_{j=1}^{k}\left[\left(\frac{\lambda_j'}{\lambda_j}\right)^{\alpha+\sum_{i:z_i=j}y_i}\exp\{-(\lambda_j'-\lambda_j)(\beta+\sum_{i:z_i=j}E_i)\}\right]\right\}.$$

3.3.2 Dimensão Variável

Quando particionamos o mapa, o número de componentes é desconhecido, fazendo com que seja necessário a utilização do algoritmo MCMC de saltos reversíveis. Segundo Brooks, Giudici e Roberts (2003), o algoritmo MCMC com saltos reversíveis é uma extensão do popular algoritmo de Metropolis-Hastings, com o objetivo de permitir movimentos entre diferentes dimensões. Estes algoritmos são muito aplicados em seleção de modelos bayesianos. A seguir, uma breve descrição da metodologia do algoritmo MCMC com saltos reversíveis.

MCMC com Saltos Reversíveis

Considere os seguintes modelos $M_1, M_2, \ldots, M_k, \ldots, M_{k_{max}}$. Denota-se o espaço paramétrico de M_k como Ξ_k. Além disso, χ_k (vetor de dimensão m_k) é um elemento de Ξ_k. No restante do texto descreveremos movimentos entre M_k e M_{k+1} com $m_k < m_{k+1}$. Segundo Green (1995), dado que a cadeia está atualmente no estado (M_k, χ_k), gera-se um novo valor para a cadeia (M_{k+1}, χ_{k+1}) de alguma distribuição proposta $U(\chi_k, \mathrm{d}\chi_{k+1})$, que é então posteriormente aceito ou rejeitado.

Para mover do modelo M_k para M_{k+1}, é gerado um vetor aleatório $\mathbf{U}$ de tamanho $m_{k+1} - m_k$ consistindo de variáveis amostradas de alguma densidade proposta $\varphi(\cdot)$. A densidade conjunta de $\mathbf{U}$ é denotada por:

$$\varphi_{m_{k+1}-m_k}(\boldsymbol{u}) = \prod_{i=k}^{m_{k+1}-m_k} \varphi(u_k)$$

Tendo gerado $\mathbf{U}$, propõem-se a mudança de modelo de χ_k para χ_{k+1}, em que $\chi_{k+1} = f_{k,(k+1)}(\chi_k, \boldsymbol{u})$. Esse movimento é aceito com probabilidade

$$\alpha\{(M_k, \chi_k), (M_{k+1}, \chi_{k+1})\} = \min\{1, A_{k,k+1}(\chi_k, \chi_{k+1})\},$$

em que

$$A_{k,k+1}(\chi_k, \chi_{k+1}) = \frac{\pi(M_{k+1}, \chi_{k+1}) r_{k,k+1}(\chi_{k+1})}{\pi(M_k, \chi_k) r_{k+1,k}(\chi_k) \varphi_{m_{k+1}-m_k}(\boldsymbol{u})} \left| \frac{\partial f_{k,k+1}(\chi_k, \boldsymbol{u})}{\partial(\chi_k, \boldsymbol{u})} \right|, \tag{3.6}$$

em que $r_{k,k+1}$ é a probabilidade de propor uma mudança do modelo M_k para o modelo M_{k+1}, π é a distribuição *a posteriori* dos diferentes modelos $M_1, \ldots, M_k, \ldots, M_{k_{max}}$, com probabilidades *a priori* $p(M_1), \ldots, p(M_k), \ldots, p(M_{k_{max}})$ e o termo final da equação 3.6 é o jacobiano. Além disso, $L_k(y|\chi_k)$ é a verossimilhança do modelo M_k e $p_k(\chi_k)$ é a densidade *a priori* do vetor de parâmetros χ_k. A distribuição *a posteriori* do modelo M_k é definida da seguinte forma:

$$\pi(M_k, \chi_k) \propto L_i(y|\chi_k) p_k(\chi_k) p(M_k).$$

Para fazer a mudança oposta, isto é, saltar do modelo M_{k+1} para o modelo M_k, a probabilidade de aceitação é o $\min\{1, 1/A\}$.

Para o modelo de mistura consideraremos que a probabilidade de transição entre os modelos, $r_{k,k+1}$, seja a mesma, exceto para o caso de $k = 1$ e $k = k_{\max}$. Além disso, a distribuição *a priori*

para os modelos será $p(M_k) \sim \text{Uniforme}\{1, 2, \ldots, k_{max}\}$, isto é, $p(M_k) = 1/k_{max}$.

Dado que estamos no estado k, primeiro escolhe-se aleatoriamente se o número de parâmetros (componentes) irá aumentar ou diminuir e depois sorteia-se qual componente vai ser dividido ou incorporado aos demais. Após isso, faz-se o *Metropolis-Hastings* para ver se a mudança de modelo é aceita.

Movimento de Divisão

Dado que estamos em um modelo com k componentes, e $r_{k,k+1}$ foi amostrado, então procede-se ao movimento de divisão, no qual um componente é escolhido aleatoriamente entre $\{1, 2, \ldots, k\}$ e é dividido em dois. Suponha que o subgrupo j foi sorteado, ele é substituído pelo componente λ_- e λ_+, com os valores gerados de:

$$\lambda_- = \lambda_j u^c \qquad\qquad \lambda_+ = \lambda_j u^{-c}$$

em que $u \sim \text{Uniforme}(0, 1)$ é a densidade proposta $\mathbf{U}$ e c é o parâmetro que controla a variabilidade do λ_- e λ_+. Observe que a distribuição proposta é univariada, isso ocorre pois, quando aumentamos um componente, teremos o acréscimo de somente um parâmetro ao modelo.

Depois que λ_- e λ_+ são criados, é necessário verificar se eles serão incorporados ao modelo. Se $\lambda_- > \lambda_{j-1}$ e $\lambda_+ < \lambda_{j+1}$ os novos valores são aceitos, senão os candidatos são rejeitados e novos λ_- e λ_+ são gerados até que as condições acima sejam satisfeitas.

Depois que os novos λ's estão definidos é necessário alocar as áreas que faziam parte do componente antigo λ_j, para os novos componentes λ_- e λ_+. Primeiramente, ordenamos as áreas pela TMP, a com menor TMP fará parte de λ_- e a com maior valor de λ_+. Os demais sítios são alocados de forma aleatória, com probabilidade dada por:

$$P(\lambda_{z_i} = \lambda_-) = \frac{e^{\psi n_- - \lambda_- E_i} \lambda_-^{y_i}}{e^{\psi n_- - \lambda_- E_i} \lambda_+^{y_i} + e^{\psi n_+ - \lambda_+ E_i} \lambda_+^{y_i}}, \tag{3.7}$$

em que n_- e n_+ são o número de sítios adjacentes à i que já fazem parte do componente λ_- e λ_+, respectivamente. Se há n_j áreas no componente j, há $2^{n_j - 2}$ configurações possíveis de alocação das áreas, e há uma probabilidade associada a cada uma dessas configurações. Para calcular a equação 3.6 é necessário especificar a probabilidade da disposição escolhida, ela será denotada por P_{alloc} e

será a acumulada da equação 3.7.

O jacobiano da função $\lambda_{k+1} = f_{k,(k+1)}(\lambda_j, \mathbf{u})$ é definido da seguinte forma:

$$\left|\frac{\partial f_{k,k+1}(\lambda_j, \mathbf{u})}{\partial(\lambda_j, \mathbf{u})}\right| = \begin{vmatrix} \frac{\partial \lambda_j u^c}{\partial \lambda_j} & \frac{\partial \lambda_j u^{-c}}{\partial \lambda_j} \\ \frac{\partial \lambda_j u^c}{\partial u} & \frac{\partial \lambda_j u^{-c}}{\partial u} \end{vmatrix} = \begin{vmatrix} u^c & u^{-c} \\ c\lambda_j u^{c-1} & -c\lambda_j u^{-(c+1)} \end{vmatrix} = \left|\frac{-2c\lambda_j}{u}\right|.$$

Portanto, a probabilidade de mudança do modelo é:

$$\begin{aligned} A &= \prod_{i:z'=-} e^{-(\lambda_- - \lambda_j)E_i} \left(\frac{\lambda_-}{\lambda_j}\right)^{y_i} \prod_{i:z'=+} e^{-(\lambda_- + \lambda_j)E_i} \left(\frac{\lambda_+}{\lambda_j}\right)^{y_i} \times \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{\lambda_- \lambda_+}{\lambda_j}\right)^{\alpha-1} \\ &= \times e^{-\beta(\lambda_- + \lambda_+ - \lambda_j)}(k+1)\frac{p(M_{k+1})}{p(M_k)} \times \exp\{\psi(U(\boldsymbol{z}') - U(\boldsymbol{z})) + \theta_k(\psi) - \theta_{k+1}(\psi)\} \\ &= \times \frac{r_{k,k+1}}{r_{k+1,k} P_{\text{alloc}}} \times \frac{2c\lambda_j}{u}. \end{aligned}$$

em que $U(\boldsymbol{z}')$ é o valor da função 3.4 no mapa candidato, e $U(\boldsymbol{z})$ é o valor da mesma função na configuração do mapa no modelo com k componentes.

Movimento de União

O movimento de união é muito similar ao movimento de divisão, em que um componente j é escolhido aleatoriamente entre $\{1, 2, \ldots, k\}$. Se o componente j foi sorteado, λ_j é incorporado à λ_{j-1} *ou* λ_{j+1}, isto é, todas as áreas que compõem o componente j são alocados em somente um dos componentes. Essa probabilidade é dada novamente pela equação 3.7, mas ao contrário do movimento de divisão, ela não é calculada separadamente para cada área i integrante do componente j. A probabilidade é única, para todas as áreas do componente.

O valor de u, ao contrário do movimento de divisão, não é gerado de uma distribuição uniforme. Ele depende do componente ao qual as áreas de j foram alocadas. Suponha que λ_{j-1} foi o componente sorteado, o valor de u será:

$$u = \sqrt[c]{\frac{\lambda_{j-1}}{\lambda_j}},$$

e probabilidade de mudança do modelo de k para $k-1$ componentes é definida como:

$$\begin{aligned}
\frac{1}{A} &= \prod_{i:z'=j-1} e^{-(\lambda_{j-1}\lambda_j)E_i} \left(\frac{\lambda_{j-1}}{\lambda_j}\right)^{y_i} \times \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{\lambda_{j-1}}{\lambda_j}\right)^{\alpha-1} e^{-\beta(\lambda_{j-1}-\lambda_j)} \\
&= \times(k-1)\frac{p(M_{k-1})}{p(M_k)} \times \exp\{\psi(U(\boldsymbol{z}') - U(\boldsymbol{z})) + \theta_k(\psi) - \theta_{k-1}(\psi)\} \\
&= \times\frac{r_{k-1,k}P_{\text{alloc}}}{r_{k,k-1}} \times \frac{u}{2c\lambda_j}.
\end{aligned}$$

Capítulo 4

Campo Aleatório de Markov Gaussiano

O método paramétrico mais utilizado para mapeamento de doenças é baseado em modelos hierárquicos, em que a especificação da distribuição *a priori* de $\boldsymbol{\lambda}$ permite introduzir dependência espacial entre os riscos, de modo que regiões próximas tendem a ter riscos semelhantes. A maneira mais comum de acrescentar essa característica é modelar o risco relativo a partir de um modelo de efeitos aleatórios, da seguinte maneira:

$$\log(\lambda_i e^{\sum_l x_{il}\gamma_l}) = \mu + \sum_l x_{il}\gamma_l + \varepsilon_i \tag{4.1}$$

em que μ é a média global do risco relativo, x_{il} é a l-ésima covariável da i-ésima área, com coeficiente γ_l e ε_i é o risco específico da i-ésima região. Sob hipótese de independência espacial, poderíamos dizer que os efeitos aleatórios ε_i são i.i.d. com uma distribuição $N(0, \sigma^2)$. Porém, quando existe uma associação espacial entre as áreas, a distribuição dos ε_i deve introduzir essa característica. Essa dependência pode ser representada sob a forma de um Campo Aleatório de Markov (CAM), para maiores detalhes consultar Rue e Held (2005).

Um Campo Aleatório Markoviano é o equivalente espacial de uma cadeia de Markov no tempo. Suponha que $\boldsymbol{\varpi}$ são parâmetros associados às áreas $i \in \{1, 2, \ldots, N\}$ de um determinado mapa. A distribuição de $\boldsymbol{\varpi}$ é chamada de CAM se, para todo i, sua distribuição conjunta satisfaz a seguinte propriedade:

$$f(\varpi_i|\boldsymbol{\varpi}_{-i}) = f(\varpi|\{\varpi_l \text{ tal que } w_{ij} > 0\}), \tag{4.2}$$

em que w_{ij} é obtido da matriz de vizinhança e $\boldsymbol{\varpi}_{-i} = (\varpi_1, \ldots, \varpi_{i-1}, \varpi_{i+1}, \ldots, \varpi_N)$. Logo, da equação 4.2, tem-se que $\boldsymbol{\varpi}$ é um CAM se a distribuição de ϖ_i condicionada à todas as áreas do

mapa, for a mesma distribuição de ϖ_i condicionada somente às áreas vizinhas.

Uma forma de definir a distribuição *a priori* de $\boldsymbol{\varepsilon}$ é através de suas densidades condicionais. Dessa maneira, podemos fazer com que áreas próximas tenham riscos relativos semelhantes. Mas a questão é: como inserir a dependência espacial através das distribuições condicionais? A solução encontrada foi modelar esses dados através de Campos Aleatórios de Markov Gaussianos (CAMG), conhecidos também como GMRF (*Gaussian Markov Random Fields*), em que as variáveis aleatórias apresentam distribuição normal.

Nas próximas seções iremos apresentar alguns exemplos de Campos Gaussianos que são muito utilizados como distribuições *a priori* em modelos hierárquicos para mapeamento de doenças.

4.1 Modelo CAR

O modelo CAR é um exemplo de um CAMG que é definido a partir de distribuições condicionais, em que ρ é o parâmetro que mede a correlação espacial. As distribuições condicionais são definidas da seguinte forma:

$$\varepsilon_i|\boldsymbol{\varepsilon}_{-i} \sim N\left(\rho\bar{\varepsilon}_{-i}, \frac{\sigma^2}{v_i}\right), \tag{4.3}$$

em que $\bar{\varepsilon}_{-i}$ denota a média dos vizinhos do sítio i, v_i é a quantidade de vizinhos da área i e σ^2 controla a variabilidade dos ε_i's.

A média de uma área, dado o resto do mapa, é definida como uma proporção da média dos sítios adjacentes. Além disso, a precisão dessa distribuição condicional é diretamente proporcional ao número de vizinhos da i-ésima área, o que faz sentido visto que, quanto mais vizinhos temos para uma determinada área, mais informação temos a respeito da mesma.

A distribuição conjunta do vetor aleatório $\boldsymbol{\varepsilon}$ será uma normal multivariada com vetor de médias zero, sendo $\sum$ a matriz de covariância e $\mathbf{Q} = \sum^{-1}$ a matriz de precisão, que é dada por:

$$\mathbf{Q} = \text{diag}(\mathbf{v}) - \rho\mathbf{W} \quad \text{ou} \quad Q_{ij} = \begin{cases} v_i & \text{se } i = j \\ -\rho & \text{se } i \sim j \\ 0 & \text{caso contrário} \end{cases}$$

onde $\mathbf{v}$ é o vetor cuja a i-ésima entrada é o número de vizinhos do sítio i e $\mathbf{W}$ é a matriz de adjacência

(ver seção 2.1). Como ilustração, para a figura 2.1 a matriz $\mathbf{Q}$ possui a seguinte estrutura:

$$\mathbf{Q} = \begin{bmatrix} 2 & -\rho & -\rho & 0 \\ -\rho & 2 & 0 & -\rho \\ -\rho & 0 & 2 & -\rho \\ 0 & -\rho & -\rho & 2 \end{bmatrix}. \tag{4.4}$$

É importante salientar que uma condição para que essa função de densidade represente, de fato, uma distribuição de probabilidade é necessário que a matriz de precisão seja simétrica e definida positiva. Essa condição só é satisfeita se o parâmetro ρ da matriz $\mathbf{Q}$ estiver entre $1/\eta_{(1)}$ e 1, onde $1/\eta_{(1)}$ é o menor autovalor da matriz $(diag(\mathbf{v}))^{-1}\mathbf{W}$. O menor autovalor será sempre negativo, de modo que ρ não será maior que 1.

Apesar de amplamente utilizado, esse modelo apresenta alguns problemas, como por exemplo: o parâmetro ρ não possui um comportamento linear, isto é, para se obter uma correlação marginal apenas moderada entre as áreas, é necessário que o valor de ρ fique muito próximo de seu limite superior, que é igual a 1. Por outro lado, quando $\rho = 0$, o que indica independência entre as áreas, a variância da distribuição condicional (equação 4.3) depende do número de vizinhos de cada sítio v_i, que não é constante através das áreas. Quando $\rho = 1$, a distribuição torna-se imprópria e caímos em um caso específico desse modelo que é denominado CAR intrínseco ou CAR impróprio, denotado por ICAR. Assunção e Krainski (2009) abordaram outros inconvenientes não intuitivos do modelo CAR, e analisaram a estrutura de $\sum$, a partir da expansão assintótica dessa matriz.

4.2 Modelo ICAR

Quando fazemos alguma análise estatística, temos como objetivo explicar o máximo da variabilidade de determinada variável. Em mapeamento de doenças, isso é o mesmo que conhecer todos os fatores de risco responsáveis por determinada patologia. Baseado nesse pensamento, Besag, Mellié e York. (1991) propuseram dividir o componente aleatório. Cada área possui efeitos aleatórios intrínsecos, e um erro espacialmente estruturado, compartilhado entre vizinhos.

Nessa abordagem, o ε_i é dividido em duas partes: uma não estruturada espacialmente, à qual se atribui distribuições normais independentes, e uma parte espacial, à qual se atribui uma distribuição

ICAR. De forma que a equação 4.1 assume o seguinte formato

$$\log(\lambda_i e^{\sum_l x_{il}\gamma_l}) = \mu + \sum_l x_{il}\gamma_l + \phi_i + \delta_i$$

onde δ_i é o erro não estruturado característico de cada área individualmente e ϕ_i é o componente do erro com correlação espacial entre as áreas.

O componente $\boldsymbol{\delta}$ representa as características de determinada área não compartilhada pelas demais, é o efeito aleatório puro. Como essa variabilidade é exclusiva de determinado sítio, não ultrapassando sua fronteira, pode ser considerada como independente do restante do mapa. Como ilustração, pode-se citar políticas públicas de determinada cidade, características econômicas de uma área, e demais fatores que podem influenciar na estimativa do RR naquele sítio.

O segundo componente $\boldsymbol{\phi}$ possui uma estrutura espacial. Ele é definido de forma a incentivar que áreas próximas sejam mais semelhantes com relação à determinada característica, do que áreas escolhidas ao acaso. Esse componente representa efeitos aleatórios compartilhados dentro de determinada região, como fatores ambientais e genéticos, que são semelhantes entre um conjunto de áreas adjacentes.

Supomos que a parte não estruturada espacialmente $\delta_1, \delta_2, \ldots, \delta_N$ são variáveis aleatórias i.i.d. com distribuição

$$\delta_i \sim N\left(0, \sigma_\delta^2\right),$$

em que σ_δ^2 controla a variabilidade dos efeitos aleatórios não estruturados espacialmente.

O segundo termo ϕ_i, que leva em conta a correlação espacial, tem distribuição ICAR. A sua função de densidade condicional assume o seguinte formato:

$$\phi_i|\boldsymbol{\phi}_{-i} \sim N\left(\bar{\phi}_{-i}, \frac{\sigma_\phi^2}{v_i}\right), \tag{4.5}$$

$\bar{\phi}_{-i}$ denota a média dos vizinhos do sítio i, σ_ϕ^2 é o parâmetro que representa a variância dos efeitos

aletórios estruturados espacialmente. A matriz de precisão do modelo ICAR é dada por

$$Q_{ij} = \begin{cases} v_i & \text{se } i = j \\ -1 & \text{se } i \sim j \\ 0 & \text{caso contrário.} \end{cases}$$

Entretanto, essa especificação não conduz à uma matriz $\mathbf{Q}$ invertível, porque as somas das linhas de $\mathbf{Q}$ são todas iguais a zero. Dado que a matriz de precisão não possui inversa, a distribuição conjunta do vetor aleatório $\boldsymbol{\phi}$ não será uma distribuição de probabilidade, já que não possui $\sum$. Consequentemente, a distribuição *a priori* de $\boldsymbol{\phi}$ é imprópria.

Ghosh et al. (1998) provaram que a distribuição *a posteriori* obtida empregando distribuições ICAR como distribuição *a priori* é própria. Isto torna possível a utilização dessa distribuição *a priori* em modelos hierárquicos bayesianos.

Capítulo 5

Simulações

Neste capítulo o desempenho do modelo semiparamétrico abordado é comparado com o modelo paramétrico comumente utilizado. No restante dessa dissertação, o modelo de mistura com correlação espacial será denotado por MIX.

Para avaliar a eficiência das diferentes metodologias, as simulações foram divididas em dois tipos: um com o risco variando suavemente entre os diferentes grupos, outro em que a diferença entre o RR dos grupos é mais abrupta. Essas duas configurações são abordadas com o intuito de verificar como diferentes situações interferem na qualidade do estimador de $\boldsymbol{\lambda}$, no modelo MIX e no modelo ICAR. Com este desafio, foram simulados vários cenários:

- Simulação em que o risco é homogêneo entre todas as áreas, isto é, $\lambda_i = 1$, $i = 1, \ldots, N$.
- Cenário gradiente, corresponde ao risco decrescendo suavemente em direção ao norte. Para a simulação em que a diferença entre os riscos é suave, o $\boldsymbol{\lambda}$ varia de 0.6 à 1.3. Já, quando a diferença é abrupta, $0.1 > \boldsymbol{\lambda} > 2.3$.
- Cenário norte-sul, onde o mapa é dividido ao meio. O risco é igual a 0.9 no norte e 1.1 no sul, para a configuração suave. E $\lambda = 0.7$ no norte e $\lambda = 1.4$ no sul, para o caso de mudança súbita na superfície do RR.
- Cenário 2 clusters, situação na qual o RR é dividido em 2 componentes, um componente possui 4 clusters, que se comportam como *outliers*. As áreas restantes fazem parte do componente com risco inferior. O valor de $\lambda = 0.85$ e $\lambda = 1.2$, para a variação suave. E $\lambda = 0.7$ e $\lambda = 1.4$ para a mudança abrupta.

- Cenário 4 clusters, em que o risco relativo assume 4 componentes, esses subgrupos são distribuídos aleatoriamente no mapa. No caso em que a mudança entre os riscos é suave, $0.7 < \boldsymbol{\lambda} < 2$. Para a simulação de mudança rápida entre os riscos, $0.1 < \boldsymbol{\lambda} < 2.3$.

Para cada conjunto de dados o número observado de eventos foi simulado de:

$$y_i \sim \text{Poisson}(\lambda_i E_i) \quad \text{independentemente para } i = 1, 2, \ldots, n,$$

em que foi utilizado dados reais para definir o valor esperado. E_i é a taxa padronizada de mortalidade por câncer de traquéia, brônquios e pulmões, estratificada segundo o sexo masculino e faixa etária no ano 2.000. O mapa utilizado para as análises é composto pelos estados de São Paulo, Paraná, Santa Catarina e Rio Grande do Sul dividido por 157 microregiões.

Para o algoritmo utilizado para estimar a constante de normalização (3.5) foram feitas 60.000 replicações para cada combinação de (k, ψ). O algoritmo está implementado nas linguagens R e C.

Para cada cenário simulado o algoritmo foi simulado uma vez, a cadeia do MCMC tinha tamanho 500.000 com *bur in* de 20.000 e *lag*=100, para o algoritmo de estimação do RR através do modelo MIX. A linguagem de programação utilizada nas simulações foi a R e a confecção dos gráficos foi realizada no *software* R 2.8.1 disponível em http://cran.r-project.org/.

Para a simulação do modelo ICAR, utilizou-se o *software* WinBUGS versão 1.4, disponível em http://www.mrcbsu.cam.ac.uk/bugs/winbugs/contents.shtml. Os gráficos dos resultados foram produzidos pelo *software* R 2.8.1.

A complexidade e ajuste dos modelos foram comparados usando o DIC (Spiegelhalter, Best, Carlin e Linde (2003)), CPO (Gilks, Richardson e Spiegelhalter (1996)), RAMSE e RAMSEL. Em dados simulados conhecemos o verdadeiro risco relativo, o que torna possível calcular o RAMSE = $(\sum_i E((\lambda_i^t - \lambda_i)^2|y)/n)^{1/2}$ e o RAMSEL = $(\sum_i E((log(\lambda_i^t) - log(\lambda_i))^2|y)/n)^{1/2}$, em que λ_i^t é o verdadeiro risco da área i. O DIC] $= E(D|y) + p_D$ é dividido em duas partes: $E(D|y)$ é a soma da deviance média *a posteriori* e p_D é o termo que penaliza a complexidade do modelo. Quando maior o valor do DIC, p_D e $E(D|y)$, pior é o desempenho desse modelo, segundo esse critério. O CPO_i pode ser interpretado como a contribuição da i-ésima observação para a adequabilidade do modelo, ele será representado pela estatística B, em que $B = \sum_i \log(\text{CPO}_i)/n$. Baseado nesse raciocínio, quanto maior o valor de CPO_i, e consequentemente da estatística B, maior será a qualidade do

ajuste.

5.1 Resultados

Risco homogêneo

A Figura 5.1 apresenta o mapa com a verdadeira superfície dos riscos (a). Nesse cenário o risco não apresenta nenhuma tendência ou correlação espacial. As estimativas para os RR são mostrados nos mapas (b-c-d). Diferentes cores indicam distintos valores associados ao RR.

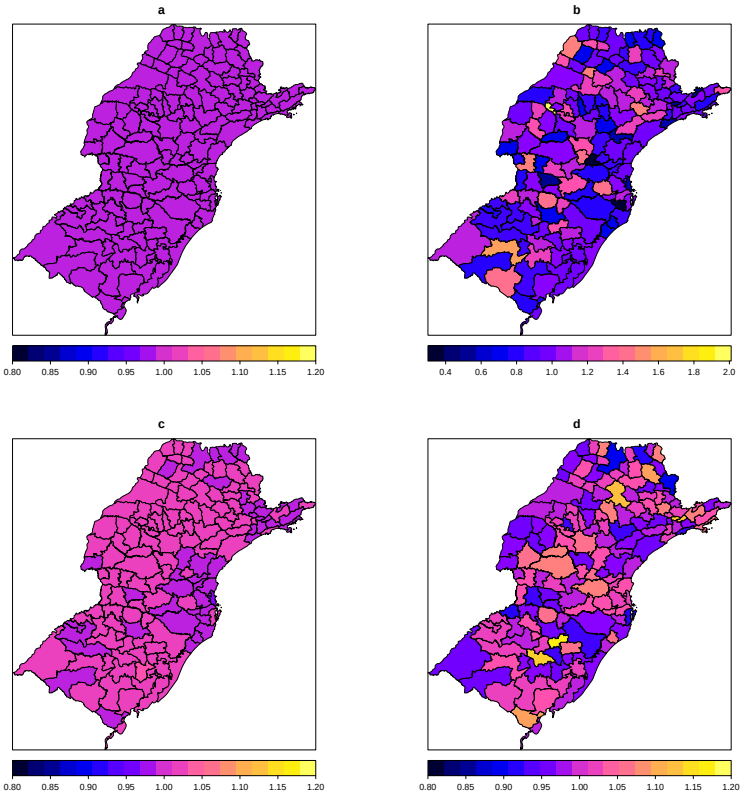


Figura 5.1: Simulação com risco homogêneo: verdadeiro risco (a) TMP observada (b) estimativa do modelo ICAR (c) estimativa do modelo MIX (d).

Pela figura 5.1 é notável a diferença entre construir uma análise utilizando somente a TMP e outra que leva em consideração a dependência espacial. A TMP (b) apresenta um comportamento heterogêneo, onde os riscos estão espalhados de forma aleatória pelo mapa. A estimativa não é suavizada e não estima de forma adequada a verdadeira superfície do risco. É importante salientar que o risco estimado através da TMP, possui um mapa com a escala diferente, em que a estimava dos λ's tem como intervalo 0,3 à 2. Nos demais mapas da figura 5.1, a escala é de 0,8 à 1,2.

Pelo mapa (c), é fácil notar que o risco estimado através do modelo ICAR é o que apresenta a

melhor estimativa, em que a superfície estimada é quase idêntica à superfície subjacente.

As médias *a posteriori* do modelo MIX são apresentadas no mapa (d). O modelo parece tentar encontrar alguma heterogeneidade espacial, dividindo o risco em vários componentes, mostrando a tendência do modelo de mistura em super-ajustar os dados nesse caso. Essa suspeita é confirmada pela valor superior de p_D do modelo MIX (tabela 5.1), em comparação com o modelo ICAR.

Para avaliar a qualidade dos modelos, os valores para RAMSE, RAMSEL, DIC e estatística B são apresentados na tabela 5.1. Como era esperado, o modelo ICAR apresentou melhores resultados em todos critérios. O que confirma sua adequabilidade na estimação do risco desse cenário.

Tabela 5.1: Resultados da simulação comparando o modelo MIX e ICAR para o cenário 1

Modelos	RAMSE	RAMSEL	B	DIC	$E(D\|y)$	p_D
MIX	0,00120	0,00115	-2,8839	175,665	137,046	38,618
ICAR	0,00012	0.00012	-2,8831	172,724	159,675	13,048

Cenário gradiente

Nesta simulação, a superfície subjacente apresenta um comportamento gradiente, onde áreas ao norte apresentam um menor risco, que aumenta na direção sul.

A figura 5.2 ilustra o verdadeiro risco e a estimativa da TMP para a simulação em que a diferença entre os riscos é suave.

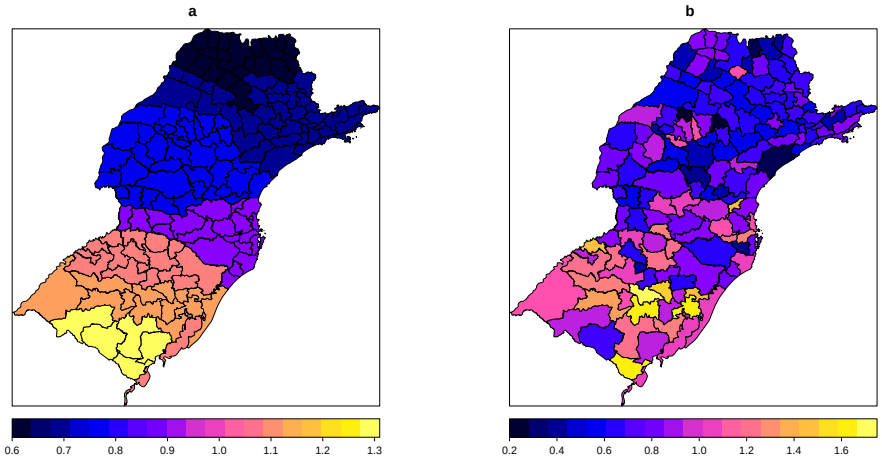


Figura 5.2: Simulação gradiente com risco variando suavemente: verdadeiro risco (a) TMP observada (b).

Na figura 5.2(b) vê-se que a TMP não capta a heterogeneidade espacial presente no RR, os valores parecem estar levemente divididos em alto risco na região sul e baixo risco na região norte.

Novamente, a TMP possui um mapa com escala diferente, com um intervalo superior aos mapas das figuras 5.2(a), 5.3 e 5.4.

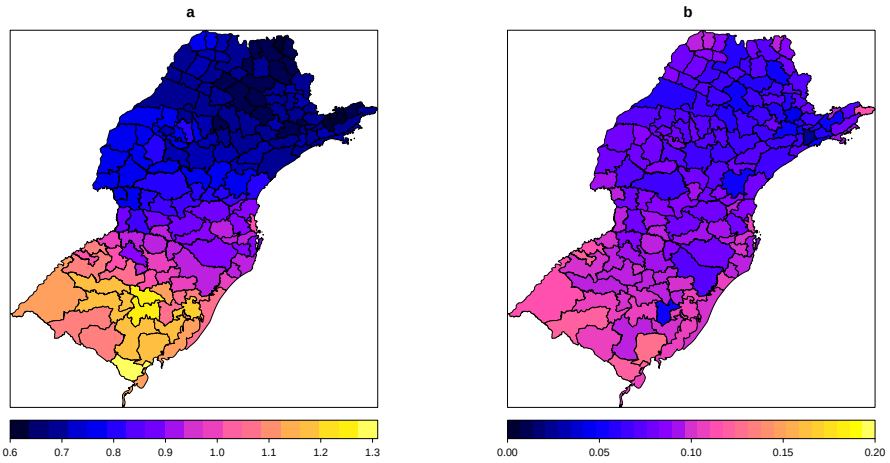


Figura 5.3: Simulação gradiente com risco variando suavemente: médias *a posteriori* (a) desvios *a posteriori* (b) do risco estimado pelo modelo ICAR.

Pela figura 5.3 nota-se que as estimativas do RR do método paramétrico captaram de forma eficaz a tendência espacial do risco, com os valores estimados aumentando à medida que se aproximam da região sul. Além disso, os desvios *a posteriori* (b) são muito baixos em comparação com a figura 5.4(b), o que indica uma menor variabilidade da estimativa em comparação com o modelo semiparámetrico. Outra característica do modelo que o mapa (b) parece sugerir, é que quanto maior for o RR, maior o valor dos desvios.

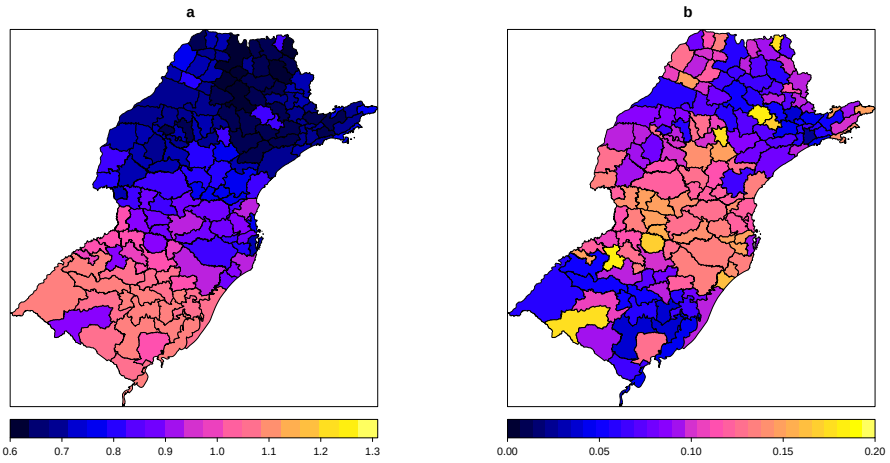


Figura 5.4: Simulação gradiente com risco variando suavemente: médias *a posteriori* (a) desvios *a posteriori* (b) do risco estimado pelo modelo MIX.

Através da figura 5.4 (a), percebe-se que o modelo MIX captou a tendência espacial, mas supersuavizou o risco relativo para esse cenário. A variabilidade do risco estimado pelo modelo parece seguir um comportamento aleatório, assumindo valores baixos tanto para áreas onde o risco é alto, quanto para áreas onde o risco é baixo.

A tabela 5.2 apresenta os resultados para comparação entre os modelos. O modelo ICAR apresentou melhores resultados em todos critérios.

Tabela 5.2: Resultados da simulação comparando o MIX e o ICAR no cenário gradiente, com RR variando suavemente.

Modelos	RAMSE	RAMSEL	B	DIC	$E(D\|y)$	p_D
MIX	0,0035	0,0046	-2,8008	182,581	133,126	49,455
ICAR	0.0016	0,0021	-2,7167	153,040	126,4815	26,558

Na figura 5.5 temos o verdadeiro risco e a estimativa da TMP para a simulação em que a diferença entre os riscos é abrupta.

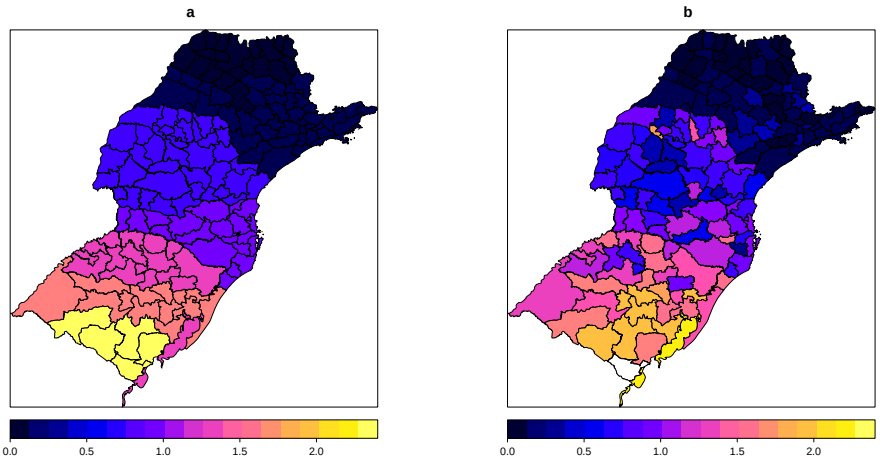


Figura 5.5: Simulação gradiente com risco variando abruptamente: verdadeiro risco (a) TMP observada (b)

A figura 5.2(b) mostra a TMP, a estimativa exibe um comportamento gradiente, de acordo com a superfície subjacente do RR.

A média *a posteriori* estimada para o modelo ICAR está ilustrada na figura 5.6 (a), novamente as estimativas captam a tendência gradiente do verdadeiro risco. Sugerindo que o modelo paramétrico foi eficiente. O comportamento dos desvios-padrão *a posteriori* parecem seguir novamente, e de forma mais evidente, a mesma variação gradiente da média.

As médias *a posteriori* do modelo MIX são apresentadas na figura 5.7 no mapa (a), a estimativa

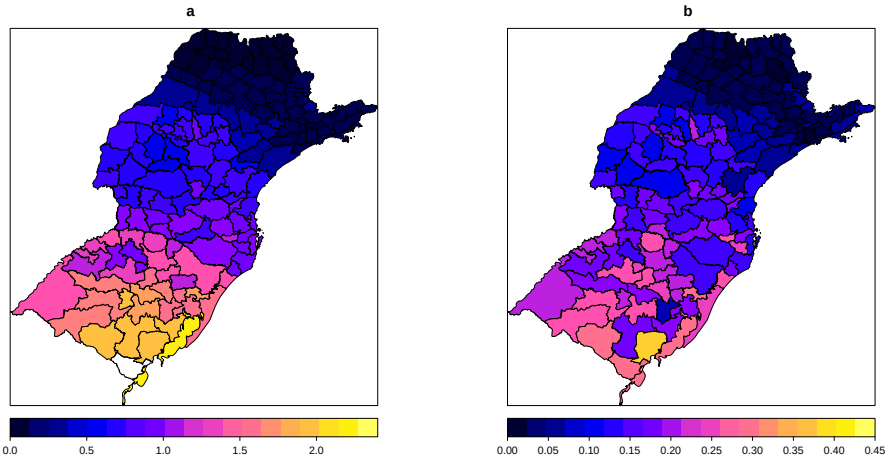


Figura 5.6: Simulação gradiente com risco variando abruptamente: médias *a posteriori* (a) desvios *a posteriori* (b) do risco estimado pelo modelo ICAR.

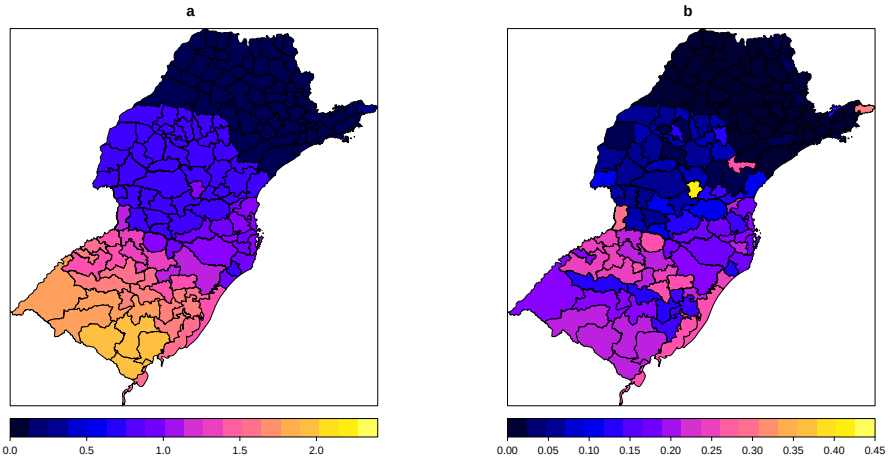


Figura 5.7: Simulação gradiente com risco variando abruptamente: médias *a posteriori* (a) desvios *a posteriori* (b) do risco estimado pelo modelo MIX.

para o RR possui um comportamento gradiente muito parecido com o do risco subjacente, como se quebrasse a variação do RR. O mapa (b) da figura mostra que os desvios possuem um comportamento diferente do cenário anterior, em que os riscos variavam de forma suave. Nessa simulação, a variabilidade da estimativa possui uma tendência gradiente, como a da média e como os desvios do modelo ICAR, em que quanto maior o valor do RR maior a variabilidade da estimativa.

Visualmente não é possível distinguir se o modelo semiparamétrico possui um desempenho superior ao método paramétrico, os mapas parecem muito simulares. Ou seja, é necessário verificar e comparar a adequabilidade dos modelos através de outros critérios. Pela tabela 5.3, temos que o modelo ICAR apresentou melhores resultados para RAMSEL, DIC, $E(D|y)$ e estatística B,

indicando ser o melhor modelo para esse cenário. O modelo MIX possui o menor valor para p_D, refletindo que o risco estimado é mais suavizado.

Tabela 5.3: Resultados da simulação comparando o MIX e o ICAR no cenário gradiente, com RR variando abruptamente.

Modelos	RAMSE	RAMSEL	B	DIC	$E(D\|y)$	p_D
MIX	0,0071	0,0307	-2,7388	216,208	169,339	46,868
ICAR	0,0086	0,0205	-2,6813	198,474	130,276	68,197

Cenário norte-sul

Nesta simulação, a superfície do RR é dividida em duas partes, as áreas ao sul apresentam risco superior que as áreas ao norte.

A figura 5.8 ilustra o verdadeiro risco (a), a média estimada *a posteriori* do ICAR (b) e do MIX (c), na simulação em que a diferença entre os riscos é suave.

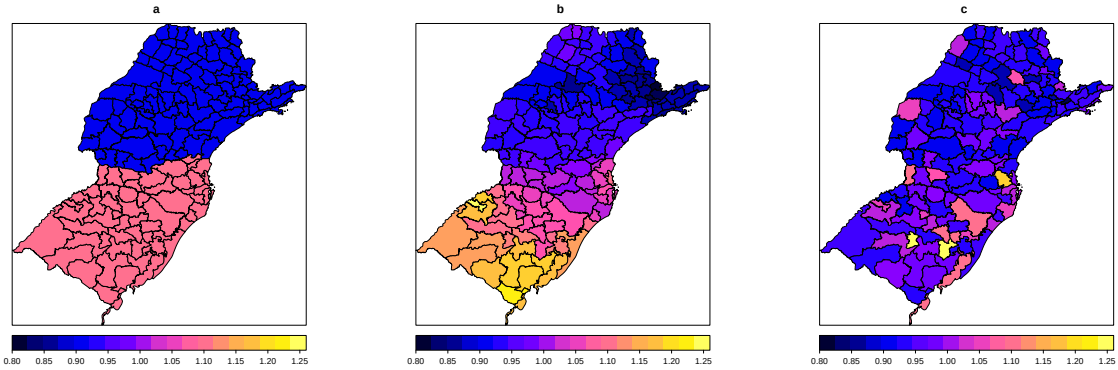


Figura 5.8: Simulação norte-sul com risco variando suavemente: verdadeiro risco (a) médias *a posteriori* do modelo ICAR (b) e do modelo MIX (c).

Pela figura 5.8 é visível a diferença das análises utilizando os modelos ICAR e MIX. O modelo paramétrico (b) apresenta um desempenho superior, em que os riscos ao norte são inferiores comparados aos do sul. Mas, é importante destacar que essa mudança é suave, ao contrário do risco subjacente, em que a troca do RR dos dois grupos do mapa é descontínua. A estimativa da média *a posteriori* do método semiparamétrico (c) apresenta um comportamento homogêneo, não captando a dependência espacial do risco. Os dados da tabela 5.4 reforçam a superioridade do modelo ICAR em relação ao MIX, pois o ICAR teve um melhor desempenho em todos os critérios avaliados.

A figura 5.9 mostra o verdadeiro risco (a), a média estimada *a posteriori* do ICAR (b) e do MIX (c), na simulação norte-sul em que a diferença entre os riscos é abrupta. Percebe-se que os dois

Tabela 5.4: Resultados da simulação comparando o MIX e o ICAR no cenário norte sul, com RR variando suavemente.

Modelos	RAMSE	RAMSEL	B	DIC	$E(D\|y)$	p_D
MIX	0,0036	0,0035	-2,8684	180,073	137,771	42,302
ICAR	0,0009	0,0009	-2,8120	164,469	144,3908	20,078

métodos estimaram de forma satisfatória o RR, destacando a diferença entre os dois grupos do mapa. Entretanto, novamente o modelo ICAR não distingue a descontinuidade do risco, a estimativa do RR parece ter um comportamento gradiente, que aumenta na direção sul. Essa característica eleva a complexidade do modelo, pois ele precisa de mais parâmetros do que necessário, a tabela 5.5 confirma essa suspeita com o valor alto de p_D.

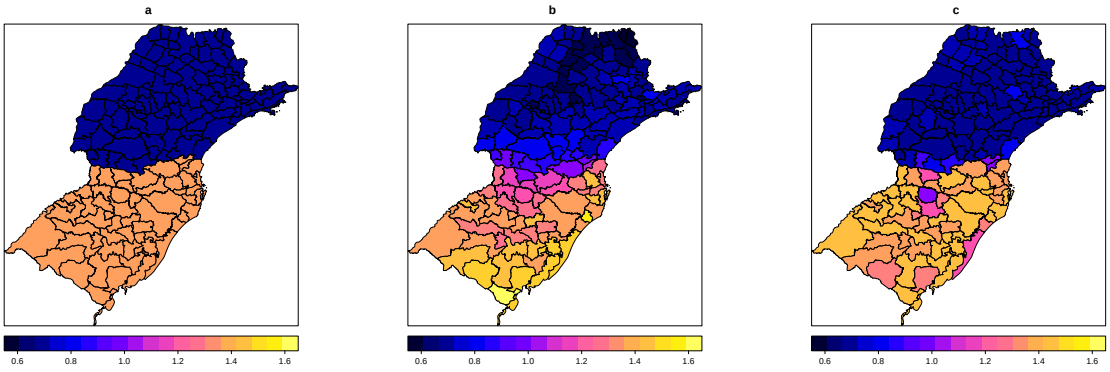


Figura 5.9: Simulação norte-sul com risco variando abruptamente: verdadeiro risco (a) médias *a posteriori* do modelo ICAR (b) e do modelo MIX (c).

Visualmente, o método semiparamétrico é mais eficiente ao estimar a superfície subjacente do RR, pois percebe a descontinuidade presente no risco. Os dados da tabela 5.5 sugerem que o melhor modelo para esse cenário é o MIX, que possui melhor desempenho na maioria dos critérios avaliados.

Tabela 5.5: Resultados da simulação comparando o MIX e o ICAR no cenário norte sul, com RR variando abruptamente.

Modelos	RAMSE	RAMSEL	B	DIC	$E(D\|y)$	p_D
MIX	0,0021	0,0019	-2,7940	160,387	129,563	30,824
ICAR	0,0049	0,0050	-2,8176	170,328	127,614	42,713

Cenário 2 clusters

Simulação na qual o RR é dividido em 2 componentes. A figura 5.10 ilustra o verdadeiro risco (a) a estimativa do modelo ICAR (b) e a estimativa do modelo MIX (c), na simulação em que a diferença entre os riscos é suave.

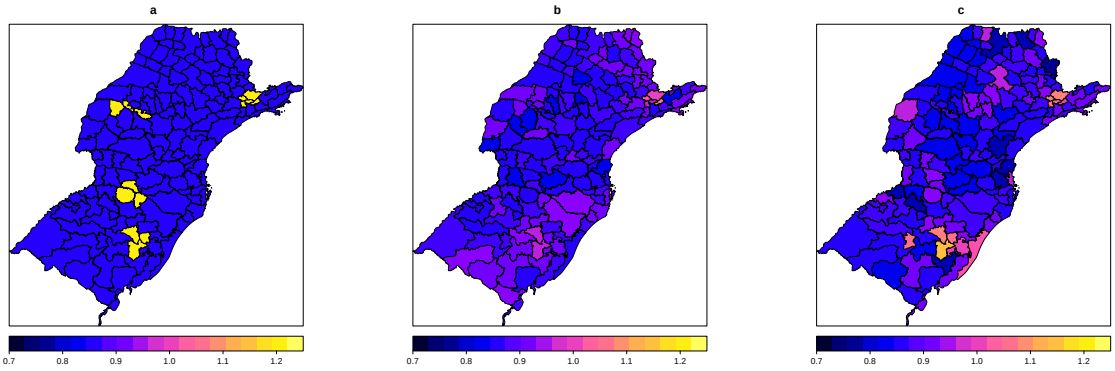


Figura 5.10: Simulação 2 clusters com risco variando suavemente: verdadeiro risco (a) médias *a posteriori* do modelo ICAR (b) e do modelo MIX (c).

Através da figura 5.10, nota-se que os dois métodos não foram muito eficazes em detectar a presença dos clusters, o risco estimado do modelo ICAR possui um comportamento homogêneo. O modelo MIX também apresenta uma superfície estimada homogênea, mas nas áreas em que o risco é superior, a estimativa é levemente maior que o restante do mapa. A tabela 5.6 sugere que os dois modelos possuem um desempenho muito parecido para esse cenário, pois a diferença entre os valores é muito pequena.

Tabela 5.6: Resultados da simulação comparando o MIX e o ICAR no cenário 2 clusters, com RR variando suavemente.

Modelos	RAMSE	RAMSEL	B	DIC	$E(D\|y)$	p_D
MIX	0,0035	0,0036	-2,8398	181,368	138,868	42,499
ICAR	0,0037	0.0037	-2,8336	177,414	144,2387	33,1762

A figura 5.11 ilustra o verdadeiro risco (a) a estimativa do modelo ICAR (b) e a estimativa do modelo MIX (c), no cenário 2 clusters em que a diferença entre os riscos é abrupta.

Na figura 5.11, percebe-se visualmente que os dois modelos foram mais eficientes em detectar a presença dos clusters, comparando com o cenário anterior. A diferença entre os métodos não é muito clara, mas os dados da tabela 5.7 indicam que o modelo MIX é mais eficiente que o ICAR.

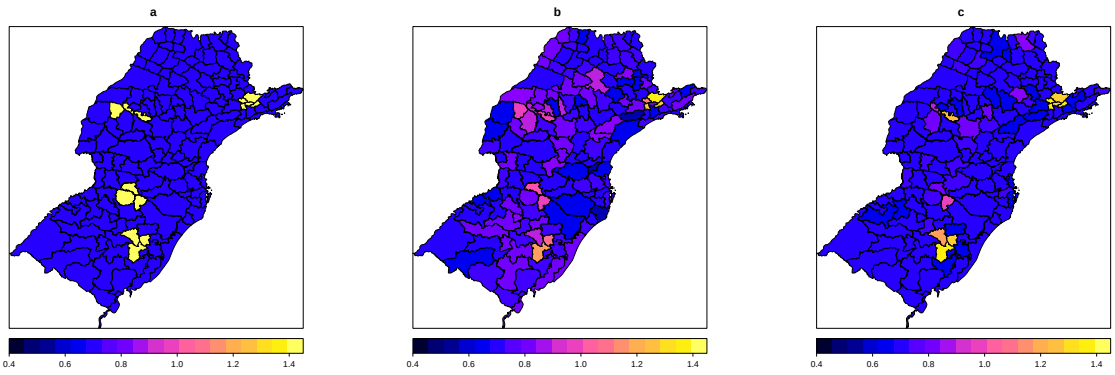


Figura 5.11: Simulação 2 clusters com risco variando abruptamente: verdadeiro risco (a) médias *a posteriori* do modelo ICAR (b) e do modelo MIX (c).

Tabela 5.7: Resultados da simulação comparando o MIX e o ICAR no cenário 2 clusters, com RR variando abruptamente.

Modelos	RAMSE	RAMSEL	B	DIC	$E(D\|y)$	p_D
MIX	0,0058	0,0054	-2,7389	165,794	127,607	38,187
ICAR	0,0093	0,0095	-2,882	203,435	136,975	66,459

Cenário 4 clusters

Cenário em que RR é dividido em 4 valores. Pensando na metodologia do modelo de mistura, seria o mesmo que 4 componentes.

A figura 5.12 (a) ilustra o verdadeiro risco e as estimativas *a posteriori* (b-c) para a simulação em que a diferença entre os riscos é suave.

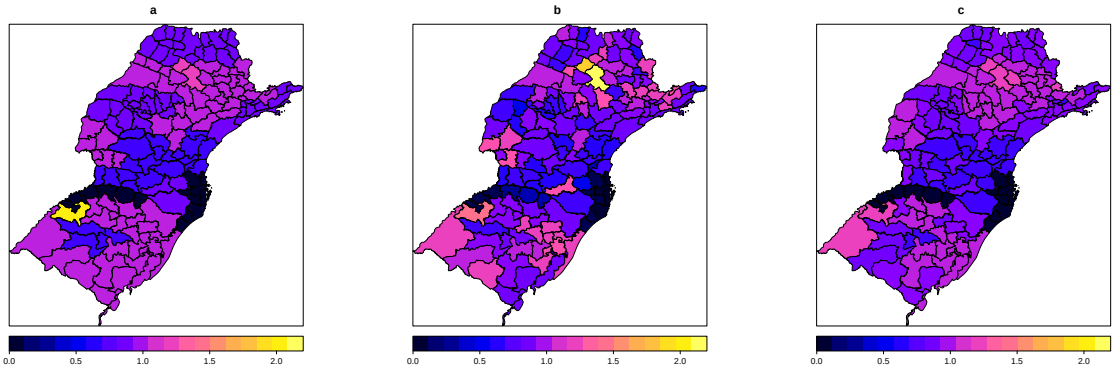


Figura 5.12: Simulação 4 clusters com risco variando suavemente: verdadeiro risco (a) médias *a posteriori* do modelo ICAR (b) e do modelo MIX (c).

Através do mapa (b-c) da figura 5.12, é fácil notar que o risco estimado através do modelo MIX é o que apresenta a melhor estimativa, em que a média *a posteriori* é muito parecida com a

superfície subjacente do RR. O modelo paramétrico parece não detectar a descontinuidade do risco entre as áreas.

Para avaliar a qualidade dos modelos, os valores para RAMSE, RAMSEL, DIC e estatística B são apresentados na tabela 5.8. Como era esperado, o modelo MIX tem um desempenho superior, com os melhores resultados em todos critérios. O que confirma sua adequabilidade no mapeamento do RR num cenário que apresenta descontinuidades.

Tabela 5.8: Resultados da simulação comparando o MIX e o ICAR no cenário 4 clusters, com RR variando suavemente.

Modelos	RAMSE	RAMSEL	B	DIC	$E(D\|y)$	p_D
MIX	0,0060	-	-2,7706	193,282	143,412	49,869
ICAR	0,0166	-	-3,3368	236,457	134,0622	102,3950

A figura 5.13 ilustra o verdadeiro risco (a), a média estimada *a posteriori* do ICAR (b) e do MIX (c), na simulação em que a diferença entre os riscos é abrupta.

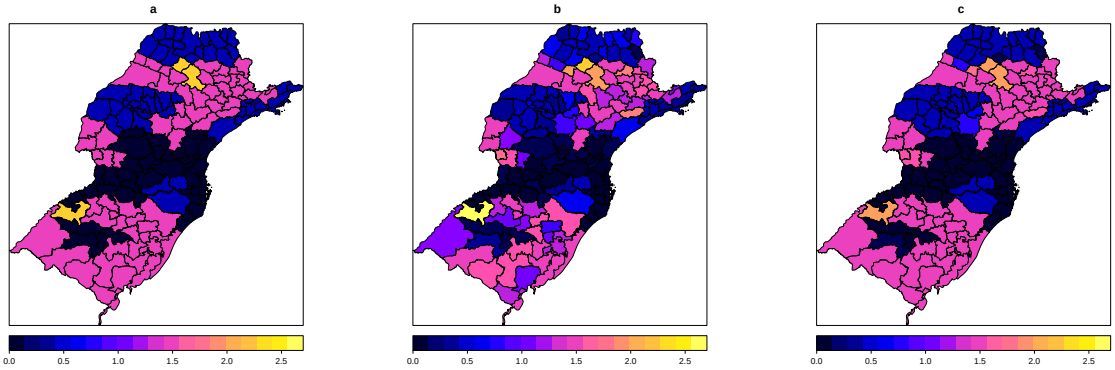


Figura 5.13: Simulação 4 clusters com risco variando abruptamente: verdadeiro risco (a) médias *a posteriori* do modelo ICAR (b) e do modelo MIX (c).

Nota-se pela figura 5.13 que o modelo MIX apresentou um desempenho superior ao modelo paramétrico (ICAR). O mapa (c) com a superfície estimada do risco é quase idêntico ao mapa (a) da superfície subjacente.

O modelo paramétrico conseguiu captar a tendência do RR, mas continua a suavizar o risco nas áreas em que ocorre descontinuidades. Ou seja, se uma área apresenta um risco baixo e uma área vizinha possui um RR alto, o modelo tende a não captar essa diferença e a estimativa acaba sendo o intermediário do risco entre as duas áreas.

Os valores apresentados na tabela 5.9 sugerem, como era esperado, que o modelo MIX tem um

desempenho superior, com os melhores resultados em todos critérios.

Tabela 5.9: Resultados da simulação comparando o MIX e o ICAR no cenário 4 clusters, com RR variando abruptamente.

Modelos	RAMSE	RAMSEL	B	DIC	$E(D\|y)$	p_D
MIX	0,0060	-	-2,6883	160,578	119,696	40,882
ICAR	0,0157	-	-3,1305	251,602	133,958	117,644

Capítulo 6

Aplicação

Neste capítulo, para ilustração da metodologia abordada, são analisados os dados da taxa de mortalidade em homens por câncer de traquéia, brônquios e pulmões nos estados de São Paulo, Paraná, Santa Catarina e Rio Grande do Sul, no ano de 2007. Os valores observados foram obtidos do DataSus, disponível no endereço eletrônico http://tabnet.datasus.gov.br/cgi/deftohtm.exe?sim/-cnv/obtbr.def. Os valores esperados foram calculados utilizando-se a técnica de padronização indireta segundo sexo(masculino) e faixa etária. As unidades geográficas utilizadas foram microrregiões definidas pelo IBGE.

A TMP de cada micro-região pode ser vista na figura 6.1. As maiores estimativas foram encontradas no Rio Grande do Sul e leste e oeste de Santa Catarina.

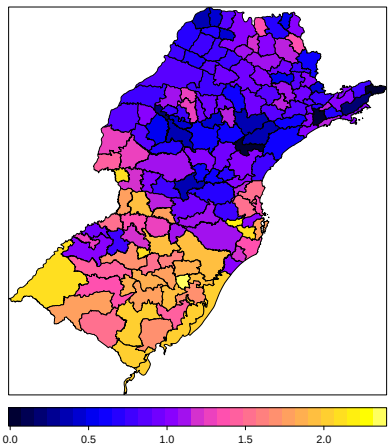


Figura 6.1: Mapa com a TMP de câncer de pulmão em homens no ano de 2007.

A figura 6.1 apresenta os valores da média *a posteriori* e do desvio-padrão *a posteriori* para o risco relativo, estimado pelo modelo CAR. O RR estimado possui um comportamento muito

parecido com a TMP, sendo levemente suavizado em comparação com a figura 6.1.

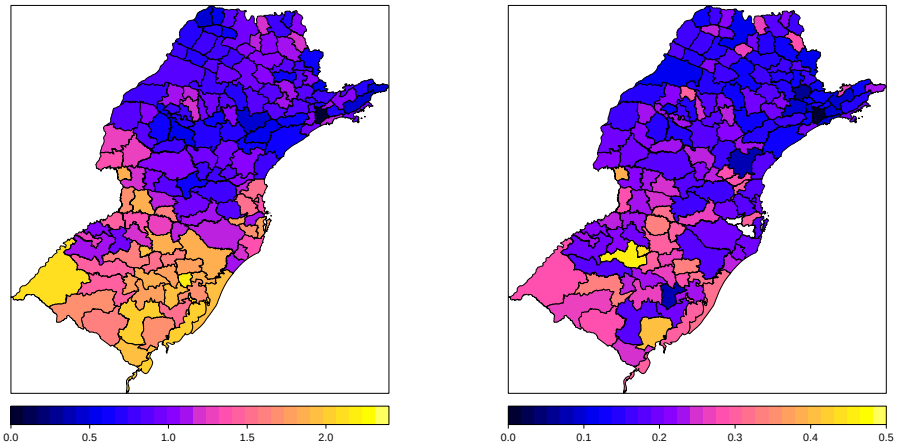


Figura 6.2: Risco estimado pelo modelo ICAR: médias *a posteriori* (esquerda) e desvios *a posteriori* (direita).

A figura 6.3 mostra os valores da média e desvios *a posteriori* do RR, estimados pelo modelo de mistura. Pode-se notar que a estimativa é suavizada, comparada com a figura 6.1 e 6.2. O RR estimado continua sendo superior no Rio Grande do Sul e Santa Catarina, mas ao contrário das estimativas anteriores, é mais homogêneo nessas regiões. Além disso, o mapa que ilustra os desvios do modelo MIX, possuem valores inferiores ao do modelo ICAR, indicando que o a estimativa do modelo paramétrico é mais instável.

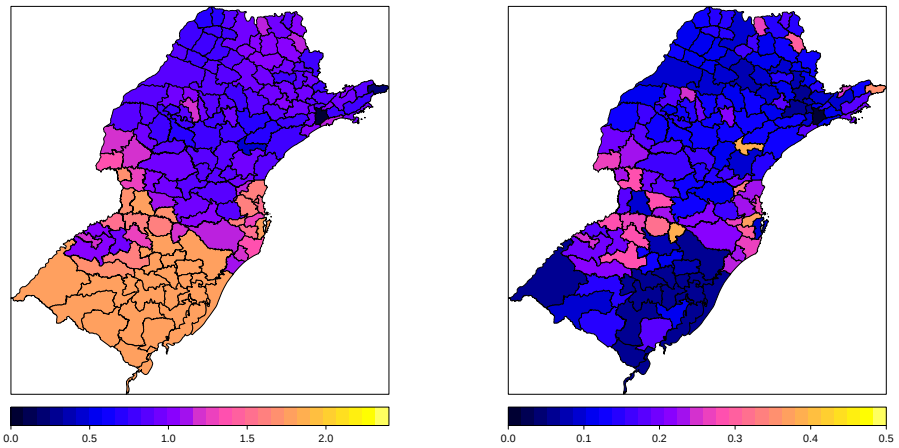


Figura 6.3: Risco estimado pelo modelo MIX: médias *a posteriori* (esquerda) e desvios *a posteriori* (direita).

Para efeito de comparação, os valores para o DIC e estatística B, para os dois modelos, são apresentados na tabela 6.1. Os resultados sugerem que o modelo MIX se ajusta melhor à esses

Tabela 6.1: Resultado da comparação do modelo MIX e ICAR para os dados de câncer de pulmão.

Modelos	B	DIC	$E(D\|y)$	p_D
Modelo de Mistura	-3,161	249,973	171,414	78,558
ICAR	-3,385	290.7870	169,292	121,495

dados.

Capítulo 7

Conclusão

7.1 Conclusões

Nos últimos anos, muitos departamentos de saúde nacionais, estaduais ou municiais coletam informações sobre doenças. Para esses departamentos de saúde é de grande interesse a utilização desses dados em uma análise mais acurada das características de determinada patologia em uma região. Com a demanda desse tipo de análise faz-se necessário a utilização de metodologias para o mapeamento do risco relativo de determinada doença.

O método mais utilizado para esse tipo de análise é o modelo CAR, que é implementado em vários *softwares* e possui baixo custo computacional. Entretanto, em alguns casos, essa metodologia não é adequada. Nessa dissertação, comparamos o CAR com uma metodologia semiparamétrica, o modelo MIX. A comparação tem como objetivo investigar a versatilidade dos dois modelos. Outro objetivo da dissertação foi a implementação do modelo semiparamétrico MIX, que não está disponível em nenhum *software*.

O modelo semiparámetrico e paramétricos foram avaliados com base nos seguintes critérios RAMSE, RAMSEL, DIC, p_D,$E(D|y)$ e CPO. Os resultados de simulação 5, indicam que, empiricamente, os modelos possuem diferentes características na estimação do RR. Para tornar mais clara essas diferenças, as simulações foram divididas em dois tipos: um em que a associação espacial é suave, e outro em que a diferença entre os riscos é acentuada.

Nos cenários em que a diferença da dependência espacial é suave, o modelo CAR demonstrou um melhor desempenho. Mas, quando o risco exibia descontinuidades, o modelo estimava de forma suave o RR, isso fez com que o modelo tivesse mais parâmetros do que o necessário, e

consequentemente valores superiores para p_D.

Ao contrário, quando o RR exibia quebras, ou mudanças abruptas, o modelo semiparamétrico captou de maneira mais eficiente o padrão espacial presente no verdadeiro risco. Mas, nos casos em o RR não exibia nenhum padrão espacial, ou esse padrão era muito suave, o modelo apresentava alguma heterogeneidade espacial, dividindo o risco em vários componentes, indicando a tendência do modelo de mistura em super-ajustar os dados nesses casos.

O tempo gasto com cada simulação do algoritmo do modelo MIX foi de 3 horas. Para a constante de normalização foram necessárias 5 horas para um mapa composto de 157 áreas. Ao contrário do artigo de Green e Richardson (2002), a cadeia MCMC do parâmetro k, não apresentou uma taxa de aceitação de 10%, mas um valor muito inferior. Baseado em estudos empíricos, suspeita-se que essa taxa de aceitação seja mais baixa em mapas que possuem uma maior quantidade de áreas, como é o nosso caso.

7.2 Perspectivas Futuras

Como sugestão de pesquisa futura, propõem-se estender a comparação do mapeamento de doenças, com outros modelos além do MIX e CAR. Como por exemplo, os modelos semiparamétricos proposto por Knorr-Held e Raβer (2000) e por Denison e Holmes (2001).

Outra sugestão seria estender o modelo de mistura espacial para o caso espaço-temporal. Uma provável metodologia é: em vez de considerar somente um mapa, utilizar vários mapas conjuntamente, sendo que cada um deles representaria um período de tempo.

Referências Bibliográficas

ASSUNÇÃO, R. M.; *Estatística Espacial com Aplicações em Epidemiologia, Economia e Sociologia*. 10 2009. Disponível em: <http://www.est.ufmg.br/~assuncao/teste.zip>.

ASSUNÇÃO, R. M.; KRAINSKI, E. Neighborhood dependence in bayesian spatial models. *Biometrical Journal*, v. 51, p. 851 – 869, 2009.

BANERJEE, S.; CARLIN, B. P.; GELFAND, A. E. *Hierarchical Modeling and Analysis for Spatial Data*. [S.l.]: Chapman & Hall, 2004.

BEAUDIN, L. *A Review of the Potts Model: Its Connection to the Tutte Polynomial and its Application to Complex Experiments*. 9 2009. Disponível em: <http://www.rose-hulman.edu-/mathjournal/archives/2007/vol8-n1/paper13/v8n1-13wo.doc>.

BESAG, J. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, v. 36, p. 135–178, 1974.

BESAG, J.; MELLIÉ, A.; YORK., J. Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, v. 43, p. 1–21, 1991.

BEST, N.; RICHARDSON, S.; THOMSON, A. A comparison of bayesian spatial models for disease mapping. *Statistical Methods in Medical Research*, v. 14, p. 35–59, 2005.

BROOKS, S. P.; GIUDICI, P.; ROBERTS, G. O. Efficiente construction of reversible jump chain monte carlo proposal distributions. *Journal of the Royal Statistical Society, Series B*, v. 65, p. 2–55, 2003.

DENISON, D. G.; HOLMES, C. C. Bayesian partitioning for estimating disease risk. *Biometrics*, v. 57, p. 143–190, 2001.

GELMAN, A.; MENG, X.-L. Simulating normalizing constants:from importance sampling to bridge sampling to path sampling. *Statistical Science*, v. 13, p. 163–185, 1998.

GHOSH, M.; NATARAJAN, K. T.; STROUD, W. F.; CARLIN, B. P. Generalized linear models for small-area estimation. *Journal of the American Statistical Association*, v. 93, p. 273–282, 1998.

GILKS, W. R.; RICHARDSON, S.; SPIEGELHALTER, D. J. *Markov Chain Monte Carlo in Practice*. [S.l.]: Chapman & Hall, 1996.

GREEN, P. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, v. 82, p. 711–732, 1995.

GREEN, P.; RICHARDSON, S. Hiden markov models and disease mapping. *Journal of the American Statistical Association*, v. 97, p. 1055–1070, 2002.

KNORR-HELD, L.; RAβER, G. Bayesian detection of clusters and discontinuities in disease maps. *Biometrics*, v. 56, p. 13–21, 2000.

RICHARDSON, S.; GREEN, P. On Bayesian Analysis of Mixtures with an Unknown Number of Components. *Journal of the Royal Statistical Society, Series B*, v. 59, p. 731–792, 1997.

ROBERT, P. C.; CASELLA, G. *Monte Carlo Statistical Methods*. [S.l.]: Springer, 1999.

RUE, H.; HELD, L. *Gaussian Markov random fields: theory and applications*. [S.l.]: Chapman & Hall, 2005.

SPIEGELHALTER D. J.; BEST N. G.; CARLIN B. P.; LINDE A. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Series B*, v. 64, p. 583–639, 2002.

WALLER, L. A.; GOTWAY, C. A. *Applied Spatial Statistics for Public Health Data*. [S.l.]: Wiley-Interscience, 2004.

Anexo: Algoritmo da constante de normalização

```
################### FUNÇÕES
coloração_do_mapa=function(vizinhos,n)
{
  n_v=rep(NA,n)
  for (i in 1:n) {n_v[i]=length(vizinhos[[i]])}
  m=matrix(nrow=2,ncol=n)
  m[2,]=n_v
  m[1,]=seq(1:n)
  o=order(m[2,],decreasing = T)
  m=m[,o]
  m[2,]=rep(-1,n) #cores
  m[2,1]=1 #coloca as áreas com o maior n° de vizinhos 1°

  for (i in 2:n) #controla as áreas
  {
    j=1
    while (j<n) #controla as cores
      {
        m[2,i]=j
        aux=vizinhos[[m[1,i]]]
        a=length(aux)
        aux2=rep(NA,a)
        for (l in 1:a){aux2[l]=which(m[1,]==aux[l])}
        cor_vizinhos=m[2,aux2]
        j=ifelse (all(cor_vizinhos != m[2,i]), n+1, j+1)
       }
  }
return(m)
}
##############################################################################
geração_mapa=function(psi,k,n,vizinhos,m,ordem,mapa)
{
  for (j in 1:n) #simulação do mapa
    {
      prob_z=rep(0,k)
      for(l in 1:k)
      {
        aux=vizinhos[[ordem[j]]]
        a=length(aux)
```

```
        aux2=rep(NA,a)
        bb=function (a,mapa,aux) {aux2[a]=which(mapa[1,]==aux[a])}
        aux2=sapply(1:a,bb,mapa=mapa, aux=aux)
        prob_z[l]= exp(psi*sum(mapa[2,aux2]==l))
       }
      probabilidade=prob_z/sum(prob_z)
      mapa[2,j]=which(rmultinom(1,n=1,prob=probabilidade) == 1)
    }
return(mapa)
}
################################################################################
função_U=function(p,vizinhos,mapa2)
{
   aux=vizinhos[[p]]
   u[p]=sum(mapa2[2,aux]==mapa2[2,p]) #vetor com o nº de vizinhos com a mesma cor
   para cada área
}
################################################################################
esperança = function(psi,k,n,simulação,vizinhos,m,mapa,ordem)
{
  U=matrix(ncol=n, nrow=simulação)
  for (i in 1:simulação) # número de simulações
  {
    mapa=geração_mapa(psi,k,n,vizinhos,m,ordem,mapa)
    o=order(mapa[1,])
    mapa2=mapa[,o]
    u=rep(NA,n)
    U[i,]=sapply(1:n,função_U,vizinhos,mapa2)
  }
  lista=list(sum(U),mapa)
return(lista)
}
################################################################################
###################DADOS
vizinhos <- list(v1 = c(2,3), v2 = c(1,4),v3=c(1,4),v4=c(2,3,5),v5=c(4))
n= 5 #tamanho de áreas do mapa
k=2 #número de grupos
simulação=60000
################################################################################
############# SIMULAÇÃO
m=coloração_do_mapa(vizinhos,n)
o=order(m[2,])
m=m[,o] #agora está ordenado por cores do grafo
ordem=rep( m[1,],simulação)

mapa=matrix(ncol=n,nrow=2) #pra 1º simulação gera o mapa
mapa[1,]=m[1,]
mapa[2,]=rep(-1,n) #em qual componente a área foi alocada
```

```
psi=seq(0,1,by=0.1)
monte_carlo=rep(NA,length(psi))
mapa_ultimo=mapa

for (j in 1:length(psi))
{
  mapa_e_U=esperança(psi[j],k,n,simulação,vizinhos,m,mapa_ultimo,ordem)
  mapa_ultimo=mapa_e_U[[2]]
  monte_carlo[j]=sum(mapa_e_U[[1]])/(2*simulação)
  write.table(monte_carlo[j],file="Monte Carlo_k=.txt",quote=F,row.names=F,
  col.names=F,append=TRUE)
  write.table(mapa_ultimo,file="mapas_k=.txt",quote=F,row.names=F,col.names=F,
  append=TRUE,sep=)
}
############  MATRIZ THETA
setwd("C:/Documents and Settings/barbian/Desktop/simulação_CONSTANTE N=5")
k_max=5
n=5
psi=seq(0,1,by=0.1)
vec_k=seq(1:k_max)
theta_k=matrix(NA,ncol=k_max,nrow=length(psi)) #colunas=grupos linhas=psi
for (i in 1:length(psi)) {theta_k[i,1]=psi[i]*n}  #n é o nº de bordas
aux_ler=rep(NA,k_max-1)
for (i in 2:k_max){aux_ler[i]=sprintf("Monte Carlo_k=%1.0f.txt",i) }

for (i in 2:k_max)
{
  MC=c(scan(aux_ler[i]))
  integral=splinefun(psi, MC,method = "fmm")
  theta_estimado=rep(0,11)
  theta_estimado[1]=n*log(vec_k[i])
  for (j in 2:length(psi))
  {
    aux=integrate(integral, 0, psi[j])
    theta_estimado[j]=aux[[1]]+n*log(vec_k[i])
  }
  theta_k[,i]=theta_estimado

}
```

Anexo: Algoritmo da estimação do risco relativo através do modelo de mistura

```
#############  METROPOLIS  lambda
risco=function(lambda_r,zi_r,k_r,E,y,beta1)
{
   incrementos=rnorm(k_r,0,0.01)
   lambda_novo_r=sort(exp(log(lambda_r)+incrementos))
   R_r=0
   for (i_r in 1:k_r)
   {
    R_r= R_r+ (1+sum(y[zi_r==i_r]))*log(lambda_novo_r[i_r]/lambda_r[i_r])-
    (lambda_novo_r[i_r]-lambda_r[i_r])*{beta1+sum(E[zi_r==i_r])}
   }
   R_r=exp(R_r)
   probabilidade_r=min(1,R_r)
   up_r=runif(1,0,1)
   if (up_r < probabilidade_r)
   {
     cont_lambda=1
     return(list(lambda_novo_r,cont_lambda))
   } else {
          cont_lambda=0
          return(list(lambda_r,cont_lambda))
          }
}
################### METROPOLIS   PSI
correlação=function(psi_c,lambda_c,n,zi_c,vizinhos,k_c,theta_k)
{
  mapa_c=matrix(ncol=n,nrow=2)
  mapa_c[1,]=seq(1:n)
  mapa_c[2,]=zi_c
  U_c=sapply(1:n,função_U,vizinhos,mapa2=mapa_c)
  U_c=sum(U_c)/2
  aux_psi=seq(0,1,by=0.1)
  if (psi_c>0 && psi_c<1)
  {
    aux_c=rbinom(1,1,0.5)
    psi_candidato_c=round((aux_c*0.1)+((1-aux_c)*(-0.1))+psi_c,dig=1)
   } else {
```

```
            if (psi_c==0)
            {
              psi_candidato_c=0.1
            }
            if (psi==1)
            {
              psi_candidato_c=0.9
            }
           }
  numerador_c=exp(psi_candidato_c*U_c-theta_k[which(aux_psi== psi_candidato_c),k_c])
  denominador_c=exp(psi_c*U_c-theta_k[which(aux_psi== psi_c),k_c])
  R_c=numerador_c/denominador_c
  probabilidade_c=min(1,R_c)
  up_c=runif(1,0,1)
  if (up_c < probabilidade_c)
  {
     cont_c=1
     return(list(psi_candidato_c,cont_c))
  } else {
           cont_c=0
           return(list(psi_c,cont_c))
         }
}
#############  GIBS  zi
alocar=function(lambda_a,zi_a,psi_a,k_a,E,y,vizinhos,beta1,localização,jj,cont)
{
   aux_componente_a=zi_a[jj[cont]]
   n_areas_a=sum(zi_a==aux_componente_a)
   if (n_areas_a==1)
   {
    aux2_a=aux_componente_a
   } else {  prob_z_a=rep(0,k_a)
             for(i_a in 1:k_a)
             {
              aux_a=vizinhos[[jj[cont]]]
              prob_z_a[i_a]= -lambda_a[i_a]*E[jj[cont]]+y[jj[cont]]*log(lambda_a[i_a])+
              psi_a*sum(zi_a[aux_a]==i_a)
             }
             prob_z2_a=prob_z_a-max(prob_z_a)
             proba_a=exp(prob_z2_a)/sum(exp(prob_z2_a))
             aux2_a=which(rmultinom(1,n=1,prob=proba_a) == 1)
          }
 return(aux2_a)
}
############ REVERSIBLE JUMP
reversible_jump=function(k_j,lambda_j,psi_j,zi_j,E,y,vizinhos,k_max,ce_j,população)
{
  aux_j=rbinom(1,1,0.5) #sorteia se divide ou une
```

```
  resul_nasce=nasce_morre(aux_nm=aux_j,k_nm=k_j,lambda_nm=lambda_j,k_max=k_max,
  ce_nm=ce_j) #chama a função
  k_candidato_j=resul_nasce[[1]]
  componente_candidato_j=resul_nasce[[2]]
  lambda_mais_j=resul_nasce[[3]]
  lambda_menos_j=resul_nasce[[4]]
  u_j=resul_nasce[[5]]
  areas_do_componente_j= which(zi_j==componente_candidato_j)
  if (k_candidato_j>k_j)   #atualizando os zi's
  {    #zi_e_PALLOC=divisão(psi,areas_do_componente,componente_candidato,lambda,zi,
vizinhos,lambda_menos,lambda_mais,E,y,)
    if (length(areas_do_componente_j)==1)
    {
      k_candidato_j=k_j-1
      zi_e_PALLOC=união(psi_u=psi_j,areas_do_componente_u=areas_do_componente_j,
          componente_candidato_u=componente_candidato_j,lambda_u=lambda_j,zi_u=zi_j,
          vizinhos=vizinhos,E=E,y=y,k_u=k_j,ce_u=ce_j)
      u_j=zi_e_PALLOC[[3]]
      PALLOC= zi_e_PALLOC[[1]]
      aux_zi=zi_e_PALLOC[[2]]
      zi_candidato_j=zi_e_PALLOC[[2]]
      aux_zi1=which(aux_zi==-100)
      aux_zi2=which(aux_zi==-22)
      aux_zi3=which(aux_zi > componente_candidato_j)
      zi_candidato_j[aux_zi1]= componente_candidato_j-1
      zi_candidato_j[aux_zi2]= componente_candidato_j
      zi_candidato_j[aux_zi3]= aux_zi[aux_zi3]-1
    ####################probabilidade do RJ ####################
      b_k=1/k_candidato_j
      d_k=ifelse(k_j==k_max,1,1/k_j)
      mapa_U_j=matrix(ncol=n,nrow=2)
      mapa_U_j[1,]=seq(1,n)
      mapa_U_j[2,]=zi_candidato_j
      U_j=sapply(1:n,função_U,vizinhos,mapa2=mapa_U_j)
      U_j=sum(U_j)/2
      mapa_U_novo=matrix(ncol=n,nrow=2)
      mapa_U_novo[1,]=seq(1,n)
      mapa_U_novo[2,]=zi_j
      U_novo=sapply(1:n,função_U,vizinhos,mapa2=mapa_U_novo)
      U_novo=sum(U_novo)/2
      if (componente_candidato_j!=1 && componente_candidato_j!=k_j)
      {
        n_prod_menos=sum(aux_zi==-100)
        areas_prod_menos=which(aux_zi==-100)
        if (length(areas_prod_menos)==0) {areas_prod_menos=0}
        prod_menos=rep(0,n_prod_menos)
        lambda_menos_j=lambda_j[componente_candidato_j-1]
        lambda_mais_j=lambda_j[componente_candidato_j+1]
```

```
       for (j_j in 1:n_prod_menos)
       {
         prod_menos[j_j]= (-(lambda_menos_j-lambda_j[componente_candidato_j])*
         E[areas_prod_menos[j_j]])+(y[areas_prod_menos[j_j]]*
log(lambda_menos_j/lambda_j[componente_candidato_j]))
       }
       prod_menos=exp(prod_menos)
       prod_menos=ifelse(length(prod_menos)==0,1,prod_menos)
       prod_menos=ifelse(prod(prod_menos)==0,1,prod_menos)
       n_prod_mais=sum(aux_zi==-22)
       areas_prod_mais=which(aux_zi==-22)
       if (length(areas_prod_mais)==0) {areas_prod_mais=0}
       prod_mais=rep(0,n_prod_mais)
       for (j_j in 1:n_prod_mais)
       {
         prod_mais[j_j]= (-(lambda_mais_j-lambda_j[componente_candidato_j])*
         E[areas_prod_mais[j_j]])+(y[areas_prod_mais[j_j]]*log(lambda_mais_j/
lambda_j[componente_candidato_j]))
       }
       prod_mais=exp(prod_mais)
       prod_mais=ifelse(length(prod_mais)==0,1,prod_mais)
       prod_mais=ifelse(prod(prod_mais)==0,1,prod_mais)
     }
     if(componente_candidato_j==1)
     {
       lambda_menos_j=0
       prod_menos=1
       lambda_mais_j=lambda_j[componente_candidato_j+1]
       n_prod_mais=sum(aux_zi==-22)
       areas_prod_mais=which(aux_zi==-22)
       if (length(areas_prod_mais)==0) {areas_prod_mais=0}
       prod_mais=rep(0,n_prod_mais)
       for (j_j in 1:n_prod_mais)
       {
         prod_mais[j_j]= (-(lambda_mais_j-lambda_j[componente_candidato_j])*
         E[areas_prod_mais[j_j]])+(y[areas_prod_mais[j_j]]*
log(lambda_mais_j/lambda_j[componente_candidato_j]))
       }
       prod_mais=exp(prod_mais)
       prod_mais=ifelse(prod(prod_mais)==0,1,prod_mais)
     }
     if(componente_candidato_j==k_j)
     {
       lambda_mais_j=0
       prod_mais=1
       n_prod_menos=sum(aux_zi==-100)
       areas_prod_menos=which(aux_zi==-100)
       if (length(areas_prod_menos)==0) {areas_prod_menos=0}
```

```
        prod_menos=rep(0,n_prod_menos)
        lambda_menos_j=lambda_j[componente_candidato_j-1]
        for (j_j in 1:n_prod_menos)
        {
          prod_menos[j_j]= (-(lambda_menos_j-lambda_j[componente_candidato_j])*
          E[areas_prod_menos[j_j]])+(y[areas_prod_menos[j_j]]*
log(lambda_menos_j/lambda_j[componente_candidato_j]))
        }
        prod_menos=exp(prod_menos)
        prod_menos=ifelse(prod(prod_menos)==0,1,prod_menos)
      }
      #aux_psi_rj=seq(0,1,by=0.1)
      #theta=theta_k[which(aux_psi_rj== psi_j),k_j]
      #theta_novo=theta_k[which(aux_psi_rj== psi_j),k_candidato_j]
      ## p(k+1) e P(k) coloquei como 1/2 aí eles se cancelam
      aux_psi_j=seq(0,1,by=0.1)
      theta=theta_k[which(round(aux_psi_j,2)== round(psi_j,2)),k_candidato_j]
      theta_novo=theta_k[which(round(aux_psi_j,2)== round(psi_j,2)),k_j]
      R_j=prod(prod_menos)*prod(prod_mais)*((beta1^alfa)/gamma(alfa))*exp(-beta1*
(lambda_menos_j+lambda_mais_j-lambda_j[componente_candidato_j]))*
(k_candidato_j+1)*exp(psi_j*(U_novo-U_j)+theta-theta_novo)*
      +((d_k*PALLOC)/b_k)*(1/(2*ce_j*lambda_j[componente_candidato_j]/u_j))
    }  else{
      zi_e_PALLOC=divisão(psi_d=psi_j,componente_candidato_d=
componente_candidato_j,zi_d=zi_j,vizinhos=vizinhos,
lambda_menos_d=lambda_menos_j, lambda_mais_d=lambda_mais_j,E=E,
y=y,população=população)
            PALLOC=zi_e_PALLOC[[1]]
            aux_zi=zi_e_PALLOC[[2]]
            zi_candidato_j=zi_e_PALLOC[[2]]
            aux_zi1=which(aux_zi==-100)
            aux_zi2=which(aux_zi==-22)
            aux_zi3=which(aux_zi > componente_candidato_j)
            t1=sum(aux_zi==-100)
            t2=sum(aux_zi==-22)
            zi_candidato_j[aux_zi3]= aux_zi[aux_zi3]+1
            zi_candidato_j[aux_zi1]= componente_candidato_j
            zi_candidato_j[aux_zi2]= componente_candidato_j+1
          ####################probabilidade do RJ ####################
            b_k=ifelse(k_j==1,1,1/k_j)
            d_k=1/(k_j+1)
            mapa_U=matrix(ncol=n,nrow=2)
            mapa_U[1,]=seq(1,n)
            mapa_U[2,]=zi_j
            U_j=sapply(1:n,função_U,vizinhos,mapa2=mapa_U)
            U_j=sum(U_j)/2
            mapa_U_novo=matrix(ncol=n,nrow=2)
            mapa_U_novo[1,]=seq(1,n)
```

```
            mapa_U_novo[2,]=zi_candidato_j
            U_novo=sapply(1:n,função_U,vizinhos,mapa2=mapa_U_novo)
            U_novo=sum(U_novo)/2
            n_prod_menos=sum(aux_zi==-100)
            areas_prod_menos=which(aux_zi==-100)
            if (length(areas_prod_menos)==0) {areas_prod_menos=0}
            prod_menos=rep(0,n_prod_menos)
            for (j_j in 1:n_prod_menos)
            {
              prod_menos[j_j]= (-(lambda_menos_j-lambda_j[componente_candidato_j])*
              E[areas_prod_menos[j_j]])+(y[areas_prod_menos[j_j]]*
  log(lambda_menos_j/lambda_j[componente_candidato_j]))
            }
            prod_menos=exp(prod_menos)
            #prod_menos=ifelse(length(prod_menos)==0,1,prod_menos)
            prod_menos=ifelse(prod(prod_menos)==0,1,prod_menos)
            n_prod_mais=sum(aux_zi==-22)
            areas_prod_mais=which(aux_zi==-22)
            if (length(areas_prod_mais)==0) {areas_prod_mais=0}
            prod_mais=rep(0,n_prod_mais)
            for (j_j in 1:n_prod_mais)
            {
              prod_mais[j_j]= (-(lambda_mais_j-lambda_j[componente_candidato_j])*
              E[areas_prod_mais[j_j]])+(y[areas_prod_mais[j_j]]*
  log(lambda_mais_j/lambda_j[componente_candidato_j]))
            }
            prod_mais=exp(prod_mais)
            #prod_mais=ifelse(length(prod_mais)==0,1,prod_mais)
            prod_mais=ifelse(prod(prod_mais)==0,1,prod_mais)
            #aux_psi_rj=seq(0,1,by=0.1)
            #theta=theta_k[which(aux_psi_rj== psi_j),k_j]
            #theta_novo=theta_k[which(aux_psi_rj== psi_j),k_candidato_j]
            ## p(k+1) e P(k) coloquei como 1/2 aí eles se cancelam
            aux_psi_j=seq(0,1,by=0.1)
            theta=theta_k[which(round(aux_psi_j,2)== round(psi_j,2)),k_j]
            theta_novo=theta_k[which(round(aux_psi_j,2)
== round(psi_j,2)),k_candidato_j]
          R_j=prod(prod_menos)*prod(prod_mais)*((beta1^alfa)/gamma(alfa))*exp(-beta1*
             (lambda_menos_j+lambda_mais_j-lambda_j[componente_candidato_j]))*
(k_j+1)*exp(psi_j*(U_novo-U_j)+theta-theta_novo)
*(d_k/(b_k*PALLOC))*(2*ce_j*lambda_j[componente_candidato_j]/u_j)
           }
  } else {
           zi_e_PALLOC=união(psi_u=psi_j,areas_do_componente_u=areas_do_componente_j,
           componente_candidato_u=componente_candidato_j,lambda_u=lambda_j,zi_u=zi_j,
           vizinhos=vizinhos,E=E,y=y,k_u=k_j,ce_u=ce_j)
           PALLOC= zi_e_PALLOC[[1]]
           aux_zi=zi_e_PALLOC[[2]]
```

```
          zi_candidato_j=zi_e_PALLOC[[2]]
          u_j=zi_e_PALLOC[[3]]
          aux_zi1=which(aux_zi==-100)
          aux_zi2=which(aux_zi==-22)
          aux_zi3=which(aux_zi > componente_candidato_j)
          zi_candidato_j[aux_zi1]= componente_candidato_j-1
          zi_candidato_j[aux_zi2]= componente_candidato_j
          zi_candidato_j[aux_zi3]= aux_zi[aux_zi3]-1
        ###################probabilidade do RJ ####################
          b_k=1/k_candidato_j
          d_k=ifelse(k_j==k_max,1,1/(k_j))
          mapa_U=matrix(ncol=n,nrow=2)
          mapa_U[1,]=seq(1,n)
          mapa_U[2,]=zi_candidato_j
          U_j=sapply(1:n,função_U,vizinhos,mapa2=mapa_U)
          U_j=sum(U_j)/2
          mapa_U_novo=matrix(ncol=n,nrow=2)
          mapa_U_novo[1,]=seq(1,n)
          mapa_U_novo[2,]=zi_j
          U_novo=sapply(1:n,função_U,vizinhos,mapa2=mapa_U_novo)
          U_novo=sum(U_novo)/2
          if (componente_candidato_j!=1 && componente_candidato_j!=k_j)
          {
            n_prod_menos=sum(aux_zi==-100)
            areas_prod_menos=which(aux_zi==-100)
            if (length(areas_prod_menos)==0) {areas_prod_menos=0}
            prod_menos=rep(0,n_prod_menos)
            lambda_menos_j=lambda_j[componente_candidato_j-1]
            lambda_mais_j=lambda_j[componente_candidato_j+1]
            for (j_j in 1:n_prod_menos)
            {
              prod_menos[j_j]= (-(lambda_menos_j-lambda_j[componente_candidato_j])*
              E[areas_prod_menos[j_j]])+(y[areas_prod_menos[j_j]]*
log(lambda_menos_j/lambda_j[componente_candidato_j]))
            }
            prod_menos=exp(prod_menos)
            prod_menos=ifelse(length(prod_menos)==0,1,prod_menos)
            prod_menos=ifelse(prod(prod_menos)==0,1,prod_menos)
            n_prod_mais=sum(aux_zi==-22)
            areas_prod_mais=which(aux_zi==-22)
            if (length(areas_prod_mais)==0) {areas_prod_mais=0}
            prod_mais=rep(0,n_prod_mais)
            for (j_j in 1:n_prod_mais)
            {
              prod_mais[j_j]= (-(lambda_mais_j-lambda_j[componente_candidato_j])*
              E[areas_prod_mais[j_j]])+(y[areas_prod_mais[j_j]]*
log(lambda_mais_j/lambda_j[componente_candidato_j]))
            }
```

```
            prod_mais=exp(prod_mais)
            prod_mais=ifelse(length(prod_mais)==0,1,prod_mais)
            prod_mais=ifelse(prod(prod_mais)==0,1,prod_mais)
          }
          if(componente_candidato_j==1)
          {
            lambda_menos_j=0
            prod_menos=1
            lambda_mais_j=lambda_j[componente_candidato_j+1]
            n_prod_mais=sum(aux_zi==-22)
            areas_prod_mais=which(aux_zi==-22)
            if (length(areas_prod_mais)==0) {areas_prod_mais=0}
            prod_mais=rep(0,n_prod_mais)
            for (j_j in 1:n_prod_mais)
            {
              prod_mais[j_j]= (-(lambda_mais_j-lambda_j[componente_candidato_j])*
              E[areas_prod_mais[j_j]])+(y[areas_prod_mais[j_j]]*
log(lambda_mais_j/lambda_j[componente_candidato_j]))
            }
            prod_mais=exp(prod_mais)
            prod_mais=ifelse(prod(prod_mais)==0,1,prod_mais)
          }
          if(componente_candidato_j==k_j)
          {
            lambda_mais_j=0
            prod_mais=1
            n_prod_menos=sum(aux_zi==-100)
            areas_prod_menos=which(aux_zi==-100)
            if (length(areas_prod_menos)==0) {areas_prod_menos=0}
            prod_menos=rep(0,n_prod_menos)
            lambda_menos_j=lambda_j[componente_candidato_j-1]
            for (j_j in 1:n_prod_menos)
            {
              prod_menos[j_j]= (-(lambda_menos_j-lambda_j[componente_candidato_j])*
              E[areas_prod_menos[j_j]])+(y[areas_prod_menos[j_j]]*
log(lambda_menos_j/lambda_j[componente_candidato_j]))
            }
            prod_menos=exp(prod_menos)
            prod_menos=ifelse(prod(prod_menos)==0,1,prod_menos)
          }
          #aux_psi_rj=seq(0,1,by=0.1)
          #theta=theta_k[which(aux_psi_rj== psi_j),k_j]
          #theta_novo=theta_k[which(aux_psi_rj== psi_j),k_candidato_j]
          ## p(k+1) e P(k) coloquei como 1/2 aí eles se cancelam
          aux_psi_j=seq(0,1,by=0.1)
          theta=theta_k[which(round(aux_psi_j,2)== round(psi_j,2)),k_candidato_j]
          theta_novo=theta_k[which(round(aux_psi_j,2)== round(psi_j,2)),k_j]
          R_j=prod(prod_menos)*prod(prod_mais)*((beta1^alfa)/gamma(alfa))
```

```
*exp(-beta1*(lambda_menos_j+lambda_mais_j-
lambda_j[componente_candidato_j]))*(k_candidato_j+1)*exp(psi_j*
(U_novo-U_j)+theta-theta_novo)*((d_k*PALLOC)/b_k)*
(1/(2*ce_j*lambda_j[componente_candidato_j]/u_j))
         }
  aux_probabilidade=min(1,R_j)
  up=runif(1,0,1)
  if (up < aux_probabilidade)
  {
     cont=1
     if (k_candidato_j<k_j)
     {
        lambda_escolhido=lambda_j[-componente_candidato_j]
     } else {
          lambda_escolhido=sort(c(lambda_j[-componente_candidato_j],lambda_mais_j,
lambda_menos_j))
#           cat("\n ",R, "\t", k_candidato, "\t", 1, "\t", zi_candidato)
            }
     erros_estimados=rep(0,n)
     for (ii_j in 1:n)
     {
        erros_estimados[ii_j]=lambda_escolhido[zi_candidato_j[ii_j]]
     }
return(list(k_candidato_j,cont,lambda_escolhido,zi_candidato_j,R_j,erros_estimados))
  } else {
           cont=0
           erros_estimados=rep(0,n)
           for (ii_j in 1:n)
           {
              erros_estimados[ii_j]=lambda_j[zi_j[ii_j]]
           }
           return(list(k_j,cont,lambda_j,zi_j,R_j,erros_estimados))
#           cat("\n ",R, "\t",k , "\t",0 ,"\t", zi)
         }
}
nasce_morre=function(aux_nm,k_nm,lambda_nm,k_max,ce_nm)
{
  if (k_nm<k_max && k_nm>1)
  {
    k_candidato_nm=aux_nm*(k_nm-1)+(1-aux_nm)*(k_nm+1)
    probabilidade_nm=rep(1/k_nm,k_nm)
    componente_candidato_nm=which(rmultinom(1,n=1,prob=probabilidade_nm) == 1)
    if (k_candidato_nm < k_nm)
    {
      lambda_mais_nm=NA
      lambda_menos_nm=NA
      u_nm=1
    } else
```

```
          {
            m1_nm=0
            m2_nm=0
            while ((m1_nm+m2_nm)<1.5) #os Lambdas devem estar no intervalo
            {
              u_nm=runif(1,0,1)
              lambda_mais_nm=lambda_nm[componente_candidato_nm]*(u_nm^(-ce_nm))
              lambda_menos_nm=lambda_nm[componente_candidato_nm]*(u_nm^ce_nm)
              if(componente_candidato_nm>1 && componente_candidato_nm<k_nm)
                {
                  m1_nm=ifelse (lambda_mais_nm > lambda_nm[componente_candidato_nm+1],0,1)
                  m2_nm=ifelse (lambda_menos_nm < lambda_nm[componente_candidato_nm-1],0,1)
                }  else
                       {
                         if (componente_candidato_nm==1)
                          {
                            m1_nm=ifelse (lambda_mais_nm >
lambda_nm[componente_candidato_nm+1],0,1)
                            m2_nm=1
                          }
                          if (componente_candidato_nm==k_nm)
                          {
                            m1_nm=1
                            m2_nm=ifelse (lambda_menos_nm <
lambda_nm[componente_candidato_nm-1],0,1)
                          }
                       }
            }
          }
  }  else
          {
            if (k_nm==1) {
                          k_candidato_nm=2
                          probabilidade_nm=rep(1/k_nm,k_nm)
 componente_candidato_nm=which(rmultinom(1,n=1,prob=
 probabilidade_nm) == 1)
 u_nm=runif(1,0,1)
                  lambda_mais_nm=lambda_nm[componente_candidato_nm]*(u_nm^(-ce_nm))
               lambda_menos_nm=lambda_nm[componente_candidato_nm]*(u_nm^ce_nm)
                          }
            if (k_nm==k_max) {
                                k_candidato_nm=k_max-1
                                probabilidade_nm=rep(1/k_nm,k_nm)
             componente_candidato_nm=which(rmultinom(1,n=1,
 prob=probabilidade_nm) == 1)
                                lambda_mais_nm=NA
                                lambda_menos_nm=NA
                                u_nm=NA
```

```
                                }
          }
return(list(k_candidato_nm,componente_candidato_nm,lambda_mais_nm,lambda_menos_nm,u_nm))
}
divisão=function(psi_d,componente_candidato_d,zi_d,vizinhos,lambda_menos_d,
lambda_mais_d,E,y,população)
{
  PALLOC_d=1
  matriz_aux_d=matrix(ncol=length(zi_d),nrow=5)
  matriz_aux_d[1,]=y/população
  matriz_aux_d[2,]=zi_d
  matriz_aux_d[3,]=localização
  matriz_aux_d[4,]=y
  matriz_aux_d[5,]=E
  matriz_aux2_d=matriz_aux_d[,which(matriz_aux_d[2,]==componente_candidato_d)]
  o_d=order(matriz_aux2_d[1,])
  matriz_aux2_d=matriz_aux2_d[,o_d]
  tamanho_matriz_d=length(matriz_aux2_d[1,])
  matriz_aux2_d[2,1]=-100 #lambda_menos
  matriz_aux2_d[2,tamanho_matriz_d]=-22  #lambda_mais
  if (tamanho_matriz_d > 2)
  {
    for (i in 2:(tamanho_matriz_d-1))
    {
      aux_vizinhos_d=vizinhos[[matriz_aux2_d[3,i]]]
      n_aux_vizinhos_d=length(aux_vizinhos_d)
      aux2_vizinhos_d=rep(0,n_aux_vizinhos_d)
     # aux2_vizinhos_d[j] = which(matriz_aux2_d[3, ] == aux_vizinhos_d[j])
      for (j in 1:n_aux_vizinhos_d)
      {
       aux2_vizinhos_d[j]=sum(which(matriz_aux2_d[3,]==aux_vizinhos_d[j]))
      }
      n_menos= sum(matriz_aux2_d[2,aux2_vizinhos_d]==-100)
      n_mais=  sum(matriz_aux2_d[2,aux2_vizinhos_d]==-22)
      numerador_menos=(psi_d*n_menos*(-lambda_menos_d)* matriz_aux2_d[5,i])+
      matriz_aux2_d[4,i]*log(lambda_menos_d)
      numerador_mais=(psi_d*n_mais*(-lambda_mais_d)*matriz_aux2_d[5,i])+
      matriz_aux2_d[4,i]*log(lambda_mais_d)
      prob_aux_d=c(numerador_menos,numerador_mais)
      prob_aux2_d=prob_aux_d-max(prob_aux_d)
      probabilidade_d=exp(prob_aux2_d)/sum(exp(prob_aux2_d))
      aux_escolha_d=rbinom(1,1,probabilidade_d[1])
      PALLOC_d=PALLOC_d*((aux_escolha_d*probabilidade_d[1])+((1-aux_escolha_d)*
probabilidade_d[2]))
      matriz_aux2_d[2,i]=-100*aux_escolha_d-22*(1-aux_escolha_d)
    }
  }
  for (i_d in 1:length(matriz_aux2_d[2,]))
```

```
  {
   for (j_d in 1:length(zi_d))
   {
    if (matriz_aux_d[3,j_d]==matriz_aux2_d[3,i_d]) {matriz_aux_d[,j_d]=matriz_aux2_d[,i_d]}
   }
  }
  zi_candidato_d=matriz_aux_d[2,]
 # PALLOC_d2=ifelse(PALLOC_d==0,0.00000000001,PALLOC_d)
return(list(PALLOC_d,zi_candidato_d))
}
união=function(psi_u,areas_do_componente_u,componente_candidato_u,lambda_u,zi_u,
vizinhos,E,y,k_u,ce_u)
{
  if(componente_candidato_u>1 && componente_candidato_u<k_u)
  {
    a_u=length(areas_do_componente_u)
    PALLOC_u=1
    zi_candidato_u=zi_u
    lambda_menos_u=lambda_u[componente_candidato_u-1]
    lambda_mais_u=lambda_u[componente_candidato_u+1]
    n_menos_u=rep(0,a_u)
    n_mais_u=rep(0,a_u)
    E_comp=rep(0,a_u)
    y_comp=rep(0,a_u)
    for (i_u in 1:a_u)
    {
      aux_viz_u=vizinhos[[areas_do_componente_u[i_u]]]
      n_menos_u[i_u]= sum(zi_candidato_u[aux_viz_u]==(componente_candidato_u-1))+
      sum(zi_candidato_u[aux_viz_u]==(-100))
      n_mais_u[i_u]= sum(zi_candidato_u[aux_viz_u]==(componente_candidato_u+1))+
      sum(zi_candidato_u[aux_viz_u]==(-22))
      E_comp[i_u]= E[areas_do_componente_u[i_u]]
      y_comp[i_u]=y[areas_do_componente_u[i_u]]
    }
    numerador_menos_u=(psi_u*sum(n_menos_u)*(-lambda_menos_u)*sum(E_comp))+sum(y_comp)*
lambda_menos_u
    numerador_mais_u=(psi_u*sum(n_mais_u)*(-lambda_mais_u)*sum(E_comp))+sum(y_comp)*
lambda_mais_u
    prob_aux_u=c(numerador_menos_u,numerador_mais_u)
    prob_aux2_u=prob_aux_u-max(prob_aux_u)
    probabilidade_u=exp(prob_aux2_u)/sum(exp(prob_aux2_u))
    aux_escolha_u=rbinom(1,1,probabilidade_u[1])
PALLOC_u=((aux_escolha_u*probabilidade_u[1])+((1-aux_escolha_u)*probabilidade_u[2]))
 zi_candidato_u[areas_do_componente_u]=(-100*aux_escolha_u)+(-22*(1-aux_escolha_u))
    u_u=aux_escolha_u*((lambda_u[componente_candidato_u-1]/lambda_u
[componente_candidato_u])^(1/ce_u))+(1-aux_escolha_u)*
((lambda_u[componente_candidato_u+1]/lambda_u[componente_candidato_u])^(-1/ce_u))
  } else{
```

```
           if (componente_candidato_u==1)
           {
              PALLOC_u=1
              zi_candidato_u=zi_u
              zi_candidato_u[areas_do_componente_u]=-22
            u_u=((lambda_u[componente_candidato_u+1]/
  lambda_u[componente_candidato_u])^(1/-ce_u))
           }
           if (componente_candidato_u==k_u){
                  PALLOC_u=1
                  zi_candidato_u=zi_u
                  zi_candidato_u[areas_do_componente_u]=-100
                  u_u=((lambda_u[componente_candidato_u-1]/
 lambda_u[componente_candidato_u])^(1/ce_u))
                 }
        }
return(list(PALLOC_u,zi_candidato_u,u_u))
}
função_U=function(p,vizinhos,mapa2)
{
   aux_u=vizinhos[[p]]
   u[p]=sum(mapa2[2,aux_u]==mapa2[2,p]) #vetor com o nº de vizinhos com a
mesma cor para cada área
}
```