



Norwegian University of
Science and Technology

Towards joint disease mapping using penalised complexity priors

Maxime Conjard

Master of Science in Mathematics (for international students)

Submission date: June 2016

Supervisor: Andrea Riebler, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

NTNU

Towards Joint Disease Mapping Using Penalized Complexity Priors

by

Maxime Conjard

Submitted for the degree of Master of Science

in the

Faculty of Information Technology, Mathematics and Electrical Engineering.
Department of Mathematical Sciences

June 2016

"If you are careful, " Garp wrote, "if you use good ingredients, and you don't take any shortcuts, then you can usually cook something very good. Sometimes it is the only worthwhile product you can salvage from a day; what you make to eat. With writing, I find, you can have all the right ingredients, give plenty of time and care, and still get nothing. Also true of love. Cooking, therefore, can keep a person who tries hard sane."

The World According To Garp, John Irving

NTNU

Abstract

Faculty of Information Technology, Mathematics and Electrical Engineering.

Department of Mathematical Sciences

Master in Mathematical Sciences

by Maxime Conjard

This thesis goal is to assess whether penalized complexity priors can be used successfully in a joint disease modeling framework and have similar properties as when used for modeling only one disease. Penalized complexity priors have never been used in models considering more than one disease at once and dealing with this is the core of this thesis. This thesis also has a didactic purpose, and is made so that the reader sees the progress that have been made throughout the ten months this thesis took to complete. This thesis shows that under certain simplifying assumptions, and with an educated choice of model that penalized complexity priors provide encouraging results when jointly modeling two diseases. Further work would be required to assess the exact extent of these results.

Acknowledgements

First I would like to express my sincere gratitude to my thesis supervisor Prof. Andrea Riebler for her patience and continuous support throughout this year.

I would also like to thanks Prof. Håvard Rue for his help and ever precious advice.

I would also like to thank my parents and my brother for the support they gave me throughout my studies, for letting me go to Norway, and for letting me stay.

My gratitude also goes to Guillaume Pierre and Quentin for holding the door.

Finally I would like to thank Aiga for bringing balance to the Force.

Contents

Abstract	2
Acknowledgements	3
Abbreviations	6
1 Introduction	1
1.1 Project Aim	1
1.2 Cancer data	2
1.2.1 Standardized mortality ratio	3
2 First Models and PC priors	7
2.1 First Model	7
2.1.1 Model results and comparison to R-INLA for oral cancer	9
2.2 Second model : the Besag York Mollie(BYM) model	11
2.2.1 Without the hyperpriors	12
2.2.2 With the hyper-priors	14
2.2.2.1 Derivation of the posterior marginal	15
2.3 Disease mapping with penalized complexity priors	19
2.3.1 Preliminary Definitions	19
2.3.2 Penalized complexity priors	20
2.3.3 New model with PC priors	22
2.3.4 Derivation of the penalized prior for τ	23
2.3.5 Derivation of the PC prior for ϕ	24
2.3.6 Derivation of the posterior marginal $\pi(u, v, \mu z)$	26
2.3.7 Implementation tricks and difficulties	27
2.3.8 Results and comparison to R-INLA	29
3 Joint disease mapping with penalized complexity priors	31
3.1 First proposed model with PC priors	32
3.1.1 Penalized complexity priors derivation	33
3.1.2 Posterior marginal $\pi(u, v, \mu z)$	35
3.1.3 Scaling and computing the priors	37
3.1.4 Results	39
3.1.4.1 Tests and analysis	41

3.2	Second Model with PC priors	43
3.2.1	Priors	44
3.2.2	Prior for ϕ_1	44
3.2.3	Posterior marginal $\pi(u_1^*, v_1, u_2^*, v_2, \mu_1, \mu_2)$	48
3.2.4	Results	50
3.3	Tests with variable mesh size	53
3.3.1	Test with a constant risk surface	53
3.3.2	Test with no disease specific component	54
3.3.3	Test Overview	54
4	Further developments	56
4.1	Extension to Poisson	56
4.2	Posterior marginal of $u_1, u_2, v_1, v_2, \mu_1, \mu_2$	57
4.3	Mean and original scale	58
4.3.1	Mean rather than mode	58
4.3.2	Original scale vs internal scale	58
4.4	Tobacco- alcohol synergy	58
4.5	Rank deficiency of the precision matrix	59
4.6	Extension to more than two diseases	59
5	Discussion	60
A	Code excerpts	62
A.1	Model 1	62
A.1.1	Implementation	62
A.2	Model 2	62
A.2.1	Implementation	62
A.3	3rd Model	63
A.3.1	Maximizing the posterior marginal for κ and τ	63
A.3.2	Computing the mode of the posterior marginal for u and v	64
A.3.3	Implementation	65
A.4	Scaling the priors	67
A.4.1	prior for ϕ_1 and midpoint formula	68
A.4.2	prior for γ and midpoint formula	69
A.4.3	Posterior marginal for $u_1, v_1, u_2, v, \mu_1, \mu_2$	70
A.4.4	Code for the second model with the PC priors	75
A.4.5	Code for the results of the second model	83
A.4.6	code for the test using the increased mesh size	84
B	Additional explanations	90
B.1	Precision matrix	90
B.2	Precision matrix First model with PC priors	91
B.3	Precision matrix Second model with PC priors	92
	Bibliography	93

Abbreviations

SMR	S tandardized M ortality R atio
MLE	M aximum L ikelihood E stimator
PC	P enalized C omplexity
GMRF	G aussian M arkov R andom F ield
KLD	K ullback L eibler D ivergence
BYM	B esag Y ork M ollie

Chapter 1

Introduction

Disease mapping refers to the analysis of the spatial patterns in disease risk across several areal units. Interest lies for example in detecting regions with elevated risks so that suitable measures can be taken [Lawson, 2013]. Often related diseases such as cancers of respiratory organs are likely to be affected by the same risk factors such as smoked tobacco and alcohol so that it seems suitable to analyse the data jointly. Thinking of two diseases Knorr-Held and Best [2001] proposed to introduce a shared spatial component in the linear predictor of both diseases weighted by a disease-specific gradient and an additional component for the residual variation, specific to only one of them. Recently, a new framework for specifying hyperpriors in hierarchical models, the penalised complexity (PC) approach, has been proposed [Simpson et al., 2015]. This approach allows to consider a model where both diseases have the same spatial field that depicts the least assuming spatial structure e.g. a very simple one with for instance a constant risk surface across the considered area called a base model, and define consequently priors to allow to blend in disease specific variations if required.

1.1 Project Aim

The project aims to investigate the use of PC priors when modeling jointly more than one disease and evaluate their relevance in that framework. Joint disease mapping has been widely covered in past publications in Knorr-Held and Best [2001] and in Held et al. [2005], however the use of penalized complexity priors in this framework is yet to be studied. Data from oral cavities, lung, oesophagus, and larynx cancers in Germany from 1986 to 1990 [Held et al., 2005] will be relied upon to assess the relevance of penalized complexity priors when applied to joint disease mapping.

Project Plan

The project is divided essentially into two parts. In the first part (chapter 1 and 2), models with one disease are studied. The idea is to start with a rather simple model and progressively extend and refine it to deal with different situations. Extending and refining the model requires more and more skills and knowledge of the field. This process is meant to help gain experience in the field of disease mapping, and ultimately to get the required perspective and skills to successfully carry out the framework's extension to two diseases in the second part of the project. As this first part has a didactic purpose, each model is implemented in R and the results are compared to R-INLA [Rue et al., 2009] to assess their correctness. The last model considered in that part is the one that is going to be relied upon to extend the use of penalized complexity priors to joint disease mapping in the second part. In the second part, the use of PC priors is extended to joint disease modeling with two diseases. The goal is to assess whether the PC priors can have the same attractive properties that motivated their development for one disease [Simpson et al., 2015]. As approximations were made to obtain the results, it is important that the latter are analyzed taking the approximations into account.

1.2 Cancer data

The data used in this project considers disease counts for male from oral, lung, larynx, and oesophagus cancers in the 544 districts of Germany from 1986 to 1990 . In order to come up with plausible models and be able to analyze the results, being aware of what increases the chances of a person getting a specific disease is paramount. These risk factors as they are called are different for each disease. The following paragraphs detail, according to www.cancer.org, the risk factors for each of the four cancers in the data set.

- **Oral cancer** The strongest risk factors for oral cancer are tobacco and alcohol use. The risk is even greater in people who combine both. Heavy smokers and drinkers face the highest risk.
- **Oesophagus cancer** Tobacco product is one of the major risk factors. The more and the longer they are used, the higher the risk. Alcohol also increases the risk of getting oesophagus cancer, and the risk increases with the consumption. Similarly to oral cancer, the combination of both alcohol and tobacco rather multiplies than adds the risks.

- **Larynx cancer** Tobacco use is the strongest risk factor for larynx cancer. Moderate and heavy drinking also increase the risk of getting larynx cancer. The combination of tobacco and alcohol is, similarly to oral and oesophagus cancer, synergetic, their combined use multiplies the risk of getting larynx cancer.
- **Lung cancer** Smoking tobacco is the strongest risk factor by quite a comfortable margin.

It appears then tobacco is a risk factor shared by all four cancers, and is also the main risk factor. Oral oesophagus and larynx cancer also have alcohol consumption as risk factor. It is important to bear in mind the main risk factors for each cancer since the goal is to assess whether the penalized complexity priors can help identify a geographical pattern induced by either of the risk factors. The data consists of the number of cases y_i and the expected number of cases e_i for each of the 544 german districts. The e_i 's are computed as follows : it is assumed that the number of disease cases y_i are independent and follow a binomial distribution $y_i \sim Bin(\pi_i, n_i)$, n_i denoting the number of person-years at risk in the i -th district. Then it is assumed that the π_i 's are small $\pi_i \ll 1$, which is not unrealistic since cancers are quite common but the odds of getting one specific cancer are quite low overall. Let $M_{y_i}(t)$ denote the moment generating function of the random variable y_i . Then it follows that:

$$\begin{aligned}
 M_{y_i}(t) &= (\pi_i e^t + (1 - \pi_i))^{n_i}, \\
 &= e^{n_i \ln(\pi_i e^t + (1 - \pi_i))}, \\
 &= e^{n_i \ln(1 + \pi_i(e^t - 1))}, \\
 &\approx e^{n_i \pi_i (e^t - 1)} \quad \text{with } \pi_i \ll 1.
 \end{aligned}$$

Hence, under the assumption that $\pi_i \ll 1$, $y_i \sim Poisson(n_i \pi_i)$. Let p denote the probability of getting the disease in all of 544 districts. The mean of the poisson is often considered as the product of two terms [Knorr-Held and Becker, 1999]:

- the expected number of death $e_i = n_i p$.
- the relative risk $\lambda_i = \frac{\pi_i}{p}$.

Thus the $y_i | \lambda_i$'s follow a Poisson distribution $y_i | \lambda_i \sim Poisson(e_i \lambda_i)$.

1.2.1 Standardized mortality ratio

The standardized mortality ratio (SMR) is the maximum likelihood estimator (MLE) of λ_i . Let f denote the density of $y_i | \lambda_i$. The MLE is obtained by computing the maximum

of the log-likelihood:

$$\frac{\partial(\ln(f(y_i|\lambda_i)))}{\partial\lambda_i} = 0.$$

Let $\hat{\lambda}_i$ denote the MLE of $y_i|\lambda_i$. Then it follows that:

$$\hat{\lambda}_i = y_i/e_i.$$

Thus $\hat{\lambda}_i$ is the standardized mortality ratio, and is often used in disease mapping but has some limitations [Wakefield, 2007], namely when the counts are low it does not perform too well. Indeed it can be shown that the variance of the estimator $\hat{\lambda}_i = y_i/e_i$ is inversely proportional to population count n_i in the i -th district:

$$\text{Var}(\hat{\lambda}_i) = \frac{\hat{\lambda}_i}{n_i p}.$$

Thus the estimate can be quite far off if n_i is small causing then the variance to be large. Besides the SMR alone cannot account for any underlying spatial structure.

This is rendered in Figures 1.1,1.2,1.3,1.4. There are not any discernible spatial patterns except for Figure 1.4, which can be explained since lung cancer is far more common and there are seldom low counts, as is shown in Table 1.1.

	disease cases
Oral	15466
Oesophagus	12835
Larynx	7283
Lung	134820

TABLE 1.1: Dead counts for each cancer type in Germany, 1986-1990

Besides, when having a close look at the count distribution for each disease, the count distributions show for oral, larynx and oesophagus in Figures 1.5, 1.6, 1.7 many very lows counts, trend that is not found for lung cancer, see Figure 1.8. The low counts make the SMR's poor indicators as said previously and explain the lack of spatial structure in Figure 1.1, 1.2, 1.3. The above consideration led to models that smooth the SMR using random effect that allow spatial variations to provide better spatial pattern identification. Models like that, allowing spatial and non spatial variability, will be studied in chapter two.

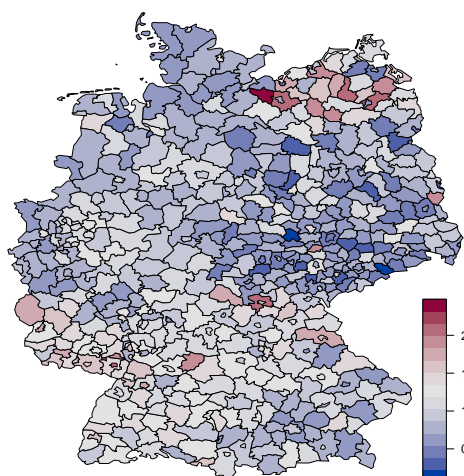


FIGURE 1.1: Oral cancer SMR

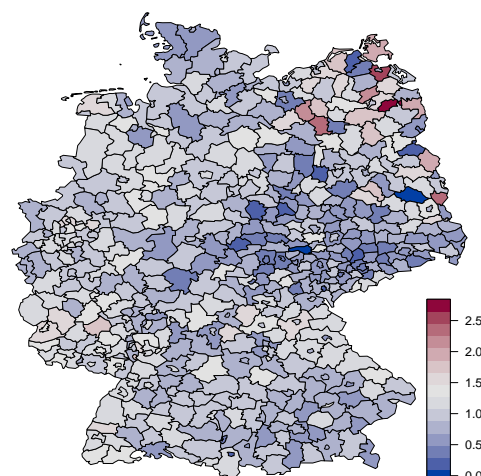


FIGURE 1.2: Oesophagus cancer SMR

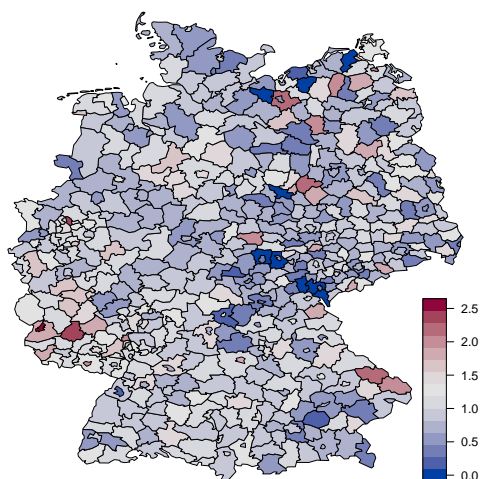


FIGURE 1.3: Larynx cancer SMR

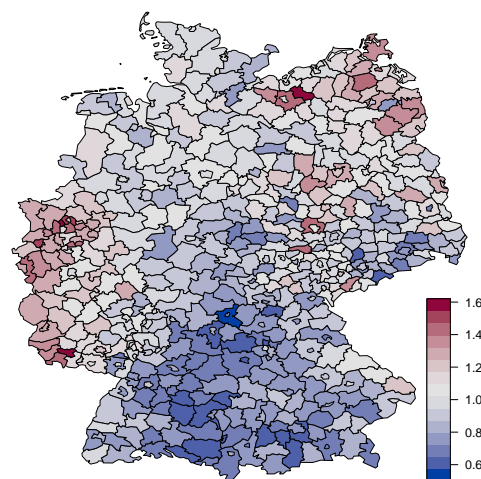


FIGURE 1.4: Lung SMR

Normal Approximation

The point of this thesis is not to work on which joint disease mapping model to adopt but rather how to use PC prior in these models. This is why, for simplicity's sake, the square root of the SMR $z_i = \sqrt{y_i/e_i}$ are assumed to be independent and follow a normal distribution $z_i|\mu_i \sim N(\mu_i, \sigma^2)$ where $\sigma^2 = \frac{1}{4}$. The models below mainly differ by the way μ_i is built.

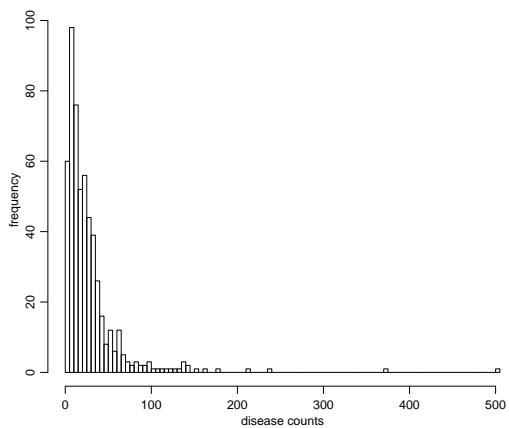


FIGURE 1.5: Oral cancer count distribution

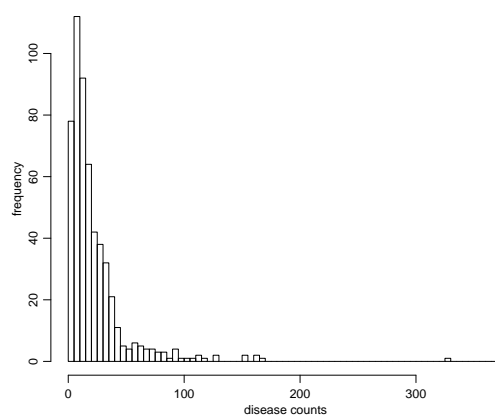


FIGURE 1.6: Oesophagus cancer count distribution

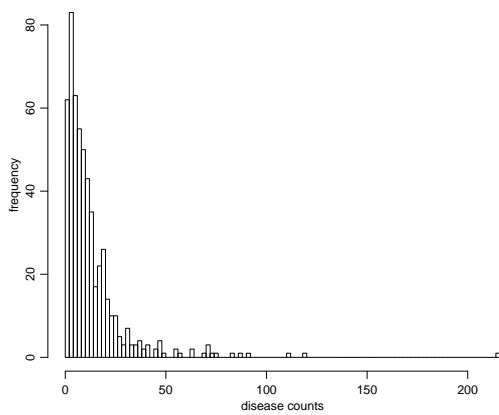


FIGURE 1.7: Larynx cancer count distribution

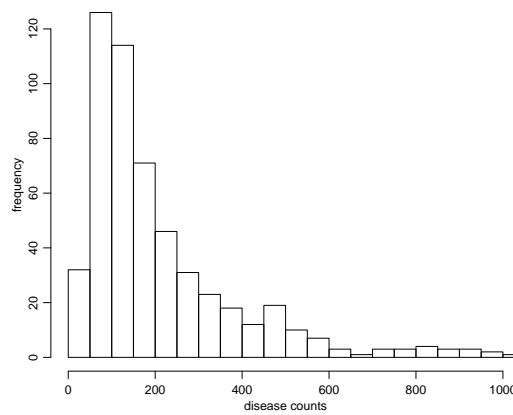


FIGURE 1.8: Lung cancer count distribution

Chapter 2

First Models and PC priors

The project is built such that the first models are very simple in order to understand what these models do, and why they are built the way they are. The goal is to learn by doing so that the leap in complexity is not too important when going from one model to the next one. It is not so much that the models gain in complexity but rather that they progressively require more and more skills and an understanding of what is going on in the previous models. Eventually successfully using the penalized complexity priors is what this part of the project aims for. The methodology used is very simple here. Since what is covered has been implemented on R-INLA [Rue et al., 2009], the results obtained are simply compared to the ones provided by INLA. Coding even the simplest results is a stance, taken mostly to get the hang of the coding skills required to carry out the more complex coding to come, and also as it has been said before, because this project has a didactic purpose, that is to understand some more about disease mapping and the PC priors.

2.1 First Model

The first model considered is one of the most used one. As said before the z_i 's are assumed to be independent and follow a normal distribution with mean μ_i and variance $\sigma^2 = \frac{1}{4}$. Besides, it is reasonable to assume that the disease has a more similar impact on close by districts than on remote ones. Closeness is defined as sharing a border $i \sim j$, in other words districts with a common border are hit by the disease more similarly than districts than do not share a common border. In order to allow for spatial structure, the $\mu_i = u_i$'s are assumed to follow a first order intrinsic Gaussian Markov Random Field (GMRF) first introduced by Besag et al. [1991]. The conditional distribution for the

u_i 's is:

$$u_i|u_{-i}, \tau \sim N\left(\frac{1}{n_{\partial i}} \sum_{j \in \partial i} u_j, \frac{1}{n_{\partial i} \tau}\right), \quad (2.1)$$

with ∂i denoting the set of neighboring district of district i , and $n_{\partial i}$ the number of neighbor district of district i . The mean of u_i is the mean of the effects over the neighbors and the precision is proportional the the number of neighbors. The joint distribution for $u|\tau$ is then:

$$\pi(u|\tau) \propto \tau^{\frac{n-1}{2}} e^{-\frac{\tau}{2} u^T R u}. \quad (2.2)$$

Let $i \sim j$ denote that district i and district j are neighbors e.g. they share a border together. Then the precision matrix R in Equation 2.2 is:

$$\begin{aligned} R_{ij} &= n_i \quad \text{if } i = j, \\ R_{ij} &= -1 \quad \text{if } i \sim j, \\ R_{ij} &= 0 \quad \text{else.} \end{aligned}$$

Note that:

$$\sum_i R_{ij} = 0 \quad \forall j,$$

thus the precision matrix R has a non empty kernel, namely the 1-vector, hence the density is invariant to adding a constant to u . This is why the precision τ in Equation 2.2 is to the power $\frac{n-1}{2}$ instead of $\frac{n}{2}$. It accounts for the rank deficiency of the precision matrix. The rank deficiency would be even higher if islands were included in the graphs. The likelihood is then as follows:

$$\pi(z|u) = \prod_{i=1}^n \left(\frac{1}{2\pi\sigma}\right)^{\frac{n}{2}} e^{\frac{1}{2\sigma^2} (u_i - z_i)^T (u_i - z_i)}.$$

The precision parameter τ is for now set constant, later it will be assigned a prior distribution, but the first model is meant to be as simple as possible. The quantity of interest is the posterior marginal for u , that is:

$$\pi(u|z) \propto \pi(z|u, \tau) \pi(u|\tau).$$

The probability density function of the posterior marginal of u , $\pi(u|z)$, is proportional to the product of two multivariate normal probability density functions and as such also follows a multivariate normal distribution. The mean and the variance of the posterior

marginal of u are derived using the following formulas:

$$\begin{aligned} u|\tau &\sim N(\mu_1, \Sigma_1), \\ z|u, \sigma^2 &\sim N(\mu_2, \Sigma_2), \\ u|z &\sim N(\mu_3, \Sigma_3). \end{aligned}$$

Where:

$$\begin{aligned} \Sigma_3 &= (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}, \\ \mu_3 &= \Sigma_3 \Sigma_1^{-1} \mu_1 + \Sigma_3 \Sigma_2^{-1} \mu_2. \end{aligned} \tag{2.3}$$

With:

$$\begin{aligned} \mu_1 &= 0, \\ \mu_2 &= z, \\ \Sigma_1 &= (\tau R)^{-1}, \\ \Sigma_2 &= \sigma^2 I, \\ \Sigma_3 &= \left(\tau R + \frac{I}{\sigma^2} \right)^{-1}, \\ \mu_3 &= \left(\tau R + \frac{I}{\sigma^2} \right)^{-1} \frac{I}{\sigma^2} z. \end{aligned}$$

Note that for this model, the results are exact since no approximation were made to compute the posterior marginal for u . Considering the simplicity of the model studied, the computation (code in Appendix A.1.1) are done only for oral cancer.

2.1.1 Model results and comparison to R-INLA for oral cancer

Figure 2.1 displays a smooth spatial pattern with areas of increased risk in the south west and north east of Germany, however, according to Riebler et al. [2016] the model does not allow for spatially unstructured variability. In other word, $\mu_i = u_i$ does not have a multivariate normal component with mean 0 and a precision matrix proportional to the identity. As a consequence unstructured random error in each district that has no spatial structure is identified as spatial correlation, leading to misinterpretation. For simplicity's sake, and for consistency with the methodology for the models to come, we only compare the mode of the posterior marginal given by the implementation (code in Appendix A.1.1) to the one provided by R-INLA. The code using R-INLA is found below:

```
1 g = system.file("demodata/germany.graph", package="INLA")
```

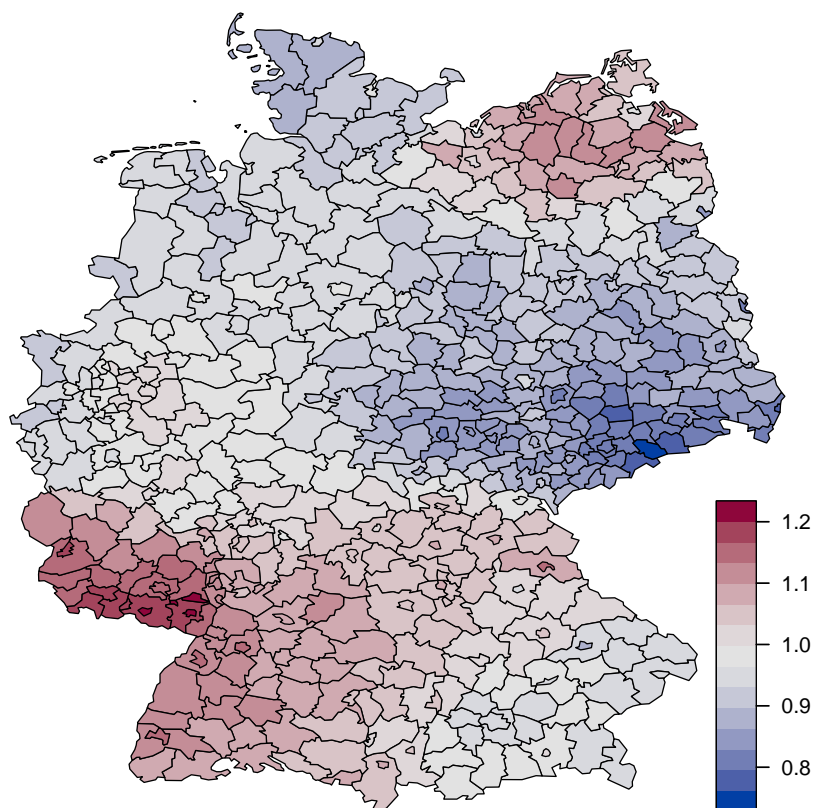


FIGURE 2.1: Mode of the posterior marginal for u for oral cancer

```

2 # Generating the precision matrix R
3 g=inla.read.graph(g)
4 myQ=inla.graph2matrix(g)
5 myQ = -myQ
6 diag(myQ) = g$nnbs
7 # Getting the data
8 Germany = or
9 Germany$z = sqrt(or$V3)
10 # structured random effect
11 formula = z ~ -1 + f(region, model="besag", graph=g, constr=FALSE,
12   hyper=list(prec=list(initial=2, fixed=TRUE)))
13 # The intercept is removed
14 result = inla(formula, family="gaussian", data=Germany,
15   control.family=list(hyper=list(prec=list(initial=log(4),
16     fixed=TRUE))))
17 # the log precision is fixed since tau is constant
18 # INLA plot check
19 a=u=result$mode$x[1:544] # u is the result of the implementation
   printfig('check1') # printfig is a function that store the plot in
   the figure folder

```

```

20 plot(a,xlab = 'district index',ylab='Difference ')
21 invisible(dev.off())
22

```

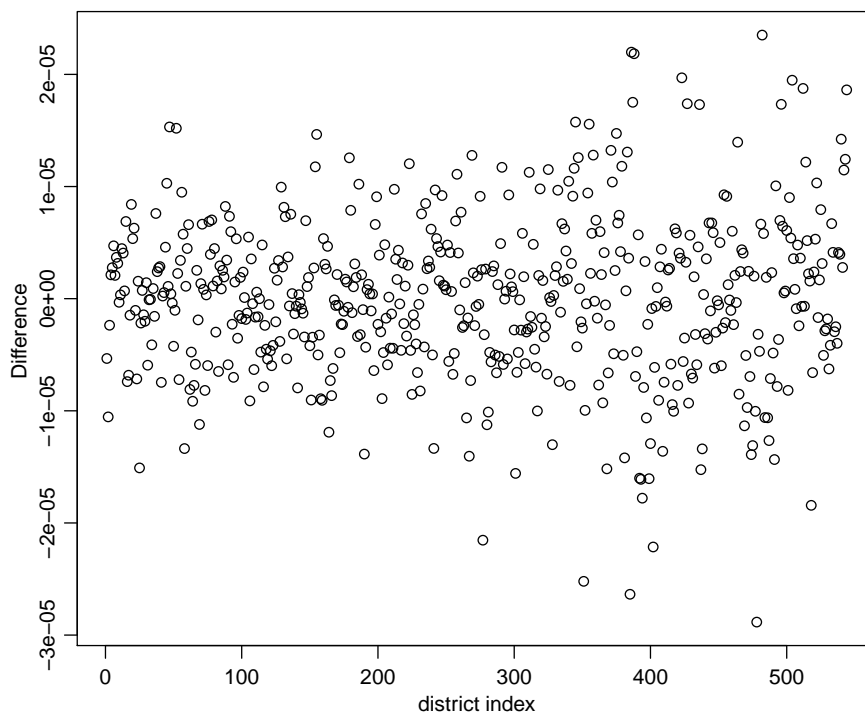


FIGURE 2.2: Difference at the mode between R-INLA and the simulation

As is seen in Figure 2.2, the difference between the implementation results and the ones produced by R-INLA is not significant.

2.2 Second model : the Besag York Mollie(BYM) model

Still assuming that $z_i \sim N(\mu_i, \sigma^2)$, Besag et al. [1991] proposed a model that accounts for spatial variations using a spatially structured effect, see Equation 2.1 and an unstructured effect. The point of the project is not to explain why this model is relevant, but to implement a model that has proved its relevance, thus for more detail see Besag et al. [1991]. The model is as follow:

$$\mu_i = u_i + v_i, \quad (2.5)$$

with $u \sim N(0, R^{-1})$ the structured effect from the previous model, see Equation 2.1, here R^{-1} is actually the generalized inverse of R since R has not full rank, and $v|\kappa \sim N(0, \kappa^{-1}I)$. The conditional distribution for u is then the same as is shown in Equation

2.1 and is as follows:

$$\pi(u|\tau) \propto \tau^{\frac{n-1}{2}} e^{-\frac{\tau}{2} u^\top R u}.$$

2.2.1 Without the hyperpriors

Hyper-priors will be taken into account later on, let $\kappa = 0.5$ and $\tau = e^2$ be constant for now. The quantity of interest is still the posterior marginal, but this time for u and v :

$$\pi(u, v|z) \propto \pi(z|u, v)\pi(u|\tau)\pi(v|\kappa),$$

the probability density function of posterior marginal is proportional to a product of multivariate normal probability density function and as such will also have a multivariate normal probability density function. As a consequence, the mode of the latter is derived to obtain its mean. Thus, the natural logarithm of the posterior marginal is differentiated with respect to u and v :

$$\begin{aligned} \frac{\partial \log(\pi(u, v|z))}{\partial u} &= -\frac{u - (z - v)}{\sigma^2} - \tau R u, \\ \frac{\partial \log(\pi(u, v|z))}{\partial v} &= -\frac{v - (z - u)}{\sigma^2} - \kappa v. \end{aligned}$$

After setting both equations to 0, it follows that:

$$\left(\frac{I}{\sigma^2} + \tau R\right)u = \frac{z - v}{\sigma^2}, \quad (2.6)$$

$$\left(\kappa + \frac{1}{\sigma^2}\right)v = \frac{z - u}{\sigma^2}. \quad (2.7)$$

Thus:

$$v = \frac{z - u}{\sigma^2\left(\kappa + \frac{1}{\sigma^2}\right)}. \quad (2.8)$$

Hence, after substituting the latter expression for v in 2.6, it follows that:

$$\begin{aligned} \left(\frac{I}{\sigma^2} + \tau R\right)u &= \frac{z - \frac{z - u}{\sigma^2\left(\kappa + \frac{1}{\sigma^2}\right)}}{\sigma^2}, \\ \left(\left(\frac{I}{\sigma^2} + \tau R\right) - \frac{I}{\sigma^4\left(\kappa + \frac{1}{\sigma^2}\right)}\right)u &= \left(\frac{1}{\sigma^2} - \frac{1}{\sigma^4\left(\kappa + \frac{1}{\sigma^2}\right)}\right)z. \end{aligned}$$

Equation 2.9 can be rewritten as:

$$u = \frac{\left(\frac{1}{\sigma^2} - \frac{1}{\sigma^4\left(\kappa + \frac{1}{\sigma^2}\right)}\right)z}{\left(\left(\frac{I}{\sigma^2} + \tau R\right) - \frac{I}{\sigma^4\left(\kappa + \frac{1}{\sigma^2}\right)}\right)}.$$

Once u is computed, it is plugged in Equation 2.8, and the result for v follows. There is no need to derive the algebraic formula for v , considering how big the formulas are getting. Since the mode of a multivariate normal distribution is its mean, the mean of the posterior marginal has been derived. In principle, the variance could be derived using twice Equation 2.3, but it proves to be tedious, and the comparison to R-INLA is only made at the mode. Similarly, rather than using Equation 2.4 twice to obtain the mode, which is tedious, it seemed easier to look for a maximum, considering the properties of the multivariate normal distribution. It is interesting to note that the results obtained through implementation are once again exact. The computations are only done for oral cancer (code in Appendix A.2.1).

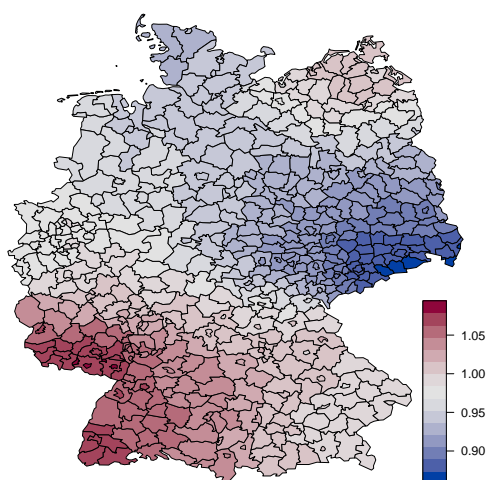


FIGURE 2.3: posterior marginal mode for u for Oral cancer

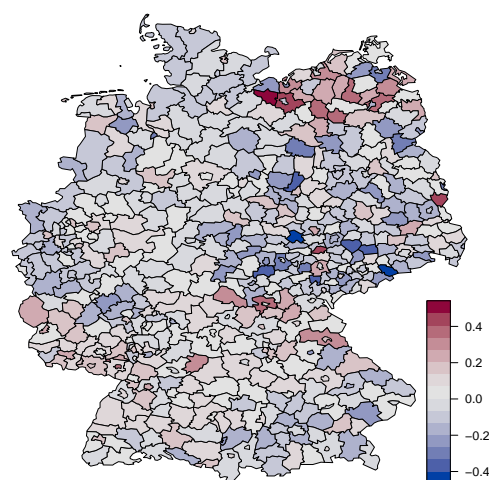


FIGURE 2.4: posterior marginal mode for v for Oral cancer

It can be seen in Figure 2.3 that there is an area of increased risk in the western part of Germany, while eastern Germany is rather an area of low risk. Figure 2.4 shows the unstructured effect, and it seems that at the mode of the posterior marginal this effect is quite significant even its scale is not as big as the one of the structured effect. It makes sense with what had been said on the previous model that did not allow for unstructured noise. Indeed with this new model, a clearer spatial pattern is obtained possibly because the unstructured effect “gets the randomness” out of the structure effect to leave the latter only with the underlying pattern. Figures 2.5, 2.6 show the difference between the results of the implementation and the results produced by R-INLA. The results obtained with the implementation check out with R-INLA. Indeed there are not any significant differences. The code used to generate the results with R-INLA is given below:

```

1 g = system.file("demodata/germany.graph", package="INLA")
2 # Get the neighbors matrix
3 g=inla.read.graph(g)
4 myQ=inla.graph2matrix(g)

```

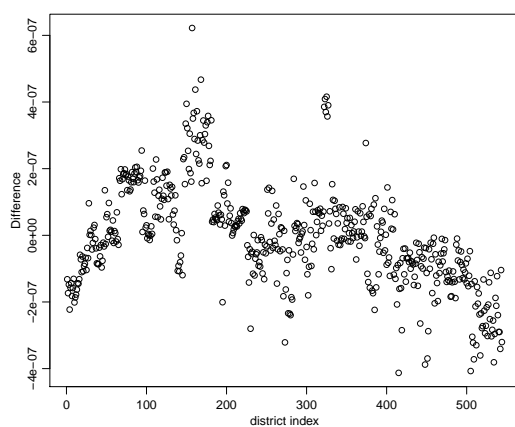


FIGURE 2.5: Difference between the posterior marginal mode from R-INLA and the implementation for the structured effect

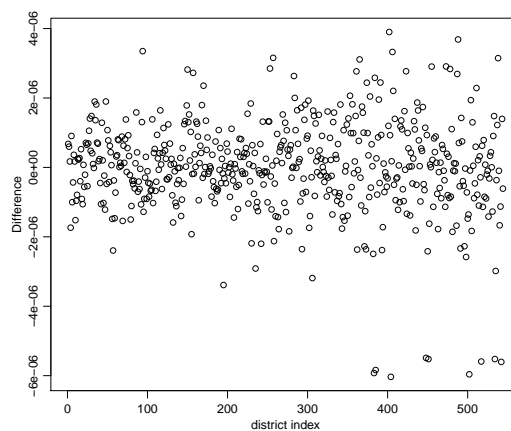


FIGURE 2.6: Difference between the posterior marginal mode from R-INLA and the implementation for the unstructured effect

```

5 myQ = -myQ
6 diag(myQ) = g$nnbs
7 # Structure for the unstructured effect
8 Oral["region.unstruc"] ← 1:544
9 # Getting the data
10 Germany = or
11 Germany$z = sqrt(or$V3)# Structured and unstructured effect
12 formula1 = z ~ -1 + f(region,model="besag",graph=g, constr=FALSE,
13   hyper=list(prec=list(initial=2, fixed=TRUE)))+f(Oral$region.unstruc
14   ,model="iid",
15   hyper=list(prec=list(initial=log(0.5), fixed=TRUE)))
16 # fixed precision
17 result1 = inla(formula1,family="gaussian",data=Germany,
18   control.family=list(hyper=list(prec=list(initial=log(4),
19   fixed=TRUE))))

```

2.2.2 With the hyper-priors

Now κ and τ are not considered to be constants anymore, it is assumed that they both follow a gamma distribution. Following Besag et al. [1991]:

$$\kappa \sim \gamma(\alpha_\kappa, \beta_\kappa),$$

$$\tau \sim \gamma(\alpha_\tau, \beta_\tau),$$

with $\alpha_\kappa = \alpha_\tau = 3$ and $\beta_\kappa = \beta_\tau = 1$ as default settings.

2.2.2.1 Derivation of the posterior marginal

The posterior marginal for u and v has now a different expression that accounts for the hyper-priors, and it can be derived as directly as before, the method used below is the one that is going to be used throughout the rest of the project to approximate the posterior marginal for the parameters of interest, here u and v , let $\theta = (\kappa, \tau)$:

$$\pi(u, v|z) \propto \int \pi(u, v|z, \theta)\pi(\theta|z)d\theta, \quad (2.9)$$

$$\pi(u, v|z) \approx \sum_k \pi(u, v|z, \theta_k)\pi(\theta_k|z)\Delta_k, \quad (2.10)$$

where $\theta_k, k = (1, \dots, K)$ correspond to representative points of $\theta|z$ and Δ_k are the corresponding weights (equal to one if points are equidistant). Equation 2.10 is rather time consuming to implement and compute. Besides, this method is going to be used later on for jointly modeling two diseases using PC priors, which has never quite been done before and, developing something new is something one feels more comfortable doing in a simplified set up rather than working with a more accurate approximation in order to focus on the novelty and not be stuck with anterior technicalities. Thus the posterior marginal is approximated as follows:

$$\pi(u, v|z) \approx \pi(u, v|z, \theta_{mode}). \quad (2.11)$$

It might seem dire, but it is actually reasonable since most of the mass of $\pi(\theta|z)$ is at its mode. Note that this is similar to an empirical Bayes strategy. Note also that Equation 2.11 can only be used for so long and if the results in later stages of the project turned out to be promising, the models would need to be implemented using Equation 2.10. Now, in order to compute Equation 2.11, the mode of the posterior marginal for θ has to be computed. In order to do so, note that:

$$\pi(\theta|z) \propto \frac{\pi(u, v, z|\theta)\pi(\theta)}{\pi(u, v|z, \theta)}. \quad (2.12)$$

Here the upsides of using z_i 's that follow a normal distribution rather than Poisson appear quite clearly. Indeed both $\pi(u, v, z|\theta)$ and $\pi(u, v|z, \theta)$ are multivariate normal which would not be the case if the y_i 's followed a Poisson distribution, for more details see Rue and Held [2005]. Equation 2.12 can be rewritten as:

$$\pi(\kappa, \tau|z) \propto \frac{\pi(z|u, v, \kappa, \tau)\pi(u|\kappa)\pi(v|\tau)\pi(\kappa)\pi(\tau)}{\pi(u, v|z, \kappa, \tau)}. \quad (2.13)$$

While $\pi(u, v, z|\kappa, \tau) \propto \pi(z|u, v, \kappa, \tau)\pi(u|\kappa)\pi(v|\tau)$ clearly follows a multivariate normal distribution, it is not as obvious for $\pi(u, v, z|\kappa, \tau)$. To prove it, the $\pi(u, v, z|k, t)$'s distribution has to be derived as in [Rue and Held \[2005\]](#):

$$\pi(u, v, z|\kappa, \tau) \propto \pi(z|u, v, \kappa, \tau)\pi(u|\tau)\pi(v|\kappa)$$

$\pi(u, v, z|\kappa, \tau)$ is a GMRF with precision Q and mean 0. The precision matrix Q is defined as follows:

$$Q = \begin{bmatrix} Q_{uu} & Q_{uv} & Q_{uz} \\ Q_{vu} & Q_{vv} & Q_{vz} \\ Q_{zu} & Q_{zv} & Q_{zz} \end{bmatrix}. \quad (2.14)$$

The expression for each block of the block matrix Q in Equation 2.14 is then derived using:

$$\begin{aligned} \pi(z|u, v, \kappa, \tau) &\propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (z_i - u_i - v_i)^2\right), \\ \pi(u|\tau) &\propto \tau^{\frac{n-1}{2}} e^{-\frac{\tau}{2} u^\top R u}, \\ \pi(v|\kappa) &\sim N(0, \kappa^{-1}I). \end{aligned}$$

Thus:

$$\begin{aligned} Q_{uu} &= \tau R + \frac{I}{\sigma^2}, \\ Q_{vv} &= \kappa I + \frac{I}{\sigma^2}, \\ Q_{uv} &= \frac{I}{\sigma^2}, \\ Q_{vu} &= \frac{I}{\sigma^2}, \\ Q_{uz} &= -\frac{I}{\sigma^2}, \\ Q_{vz} &= -\frac{I}{\sigma^2}. \end{aligned}$$

Now that the distribution for $u, v, z|\kappa, \tau$, the only thing left to do is to use it to derive $u, v|z, \kappa, \tau$.

Theorem 2.1. Let $x \sim N_c(b, Q)$, with $x = \begin{pmatrix} x_A \\ x_B \end{pmatrix}$ and $Q = \begin{bmatrix} Q_{AA} & Q_{AB} \\ Q_{BA} & Q_{BB} \end{bmatrix}$

and $x = \begin{pmatrix} b_A \\ b_B \end{pmatrix}$ then, $x_A|x_B \sim N_c(b_A - Q_{AB}x_B, Q_{AA})$. See [Rue and Held \[2005\]](#) for more details.

From there using Theorem 2.1 allows rather easily to obtain the distribution of the posterior marginal:

$$\pi(u, v|z, \kappa, \tau) \sim N_c(b = - \begin{bmatrix} Q_{uz} \\ Q_{vz} \end{bmatrix} \begin{pmatrix} z \\ z \end{pmatrix}, Q_1 = \begin{bmatrix} Q_{uu} & Q_{uv} \\ Q_{vu} & Q_{vv} \end{bmatrix}).$$

Now in order to make finding $\kappa_{mode}, \tau_{mode} = \arg \max_{\kappa, \tau} \pi(\kappa, \tau|z)$ easier, it helps noticing that $\pi(\kappa, \tau|z)$ does not depend on u and v . Thus let $u = 0$ and $v = 0$. This choice of u and v makes the full conditional of κ and τ as easy to implement as possible. Then Equation 2.13 becomes:

$$\pi(\kappa, \tau|z) \propto \frac{\pi(z|u=0, v=0, \kappa, \tau)\pi(u=0|\kappa)\pi(v=0|\tau)\pi(\kappa)\pi(\tau)}{\pi(u=0, v=0|z, \kappa, \tau)}, \quad (2.15)$$

$$\pi(\kappa, \tau|z) \propto \frac{\tau^{\frac{n-1}{2}} \left(\frac{\kappa}{2\pi}\right)^{\frac{n}{2}}}{|Q_1| e^{-\frac{1}{2}\mu_1^T Q_1 \mu_1}}, \quad (2.16)$$

with $\mu_1 = Q_1^{-1}b$. Now $\kappa_{mode}, \tau_{mode} = \operatorname{argmax}_{\kappa, \tau} \pi(\kappa, \tau|z)$ can be obtained using the R built in function `optim`(code in Appendix A.3.1). The results obtained in Table 2.1

	Implementation	R-INLA
τ_{max}	18.02	18.02
κ_{max}	28.38	28.39

TABLE 2.1: argmax of the posterior marginal $\pi(\kappa, \tau|z)$

clearly show that the implementation's results are corroborated by R-INLA. Now using Equation 2.11, it is easy to obtain the mode of the posterior marginal for u and v , and then compare it to R-INLA. It seems (code in Appendix A.3.2) that oral cancer in Figure 2.7, oesophagus cancer in Figure 2.8, and larynx cancer in Figure 2.9 have a rather similar structured effect component with area of increased risk in western and north eastern Germany. As has been said in Section 1.2, tobacco and alcohol are the main risk factors for these three cancers, it only makes sense that these three diseases have a similar structured effect. The differences between these three spatial patterns show essentially that tobacco and alcohol are not the only risk factor at play. Lung cancer, however, as is seen in Figure 2.10, seems to have an altogether different structured effect. It makes sense for oral oesophagus and larynx are mostly caused both alcohol and tobacco consumption while tobacco is by far the risk factor to blame for most lung cancer as has been said in Section 1.2. The obtained results seem to add up, feeling confirmed when looking at the difference between the results from R-INLA and the implementation for oral cancer in Figures 2.11, 2.12, it all but fits. The code used to generate the R-INLA results is given below and commented:

```
1 # Precision matrix
```

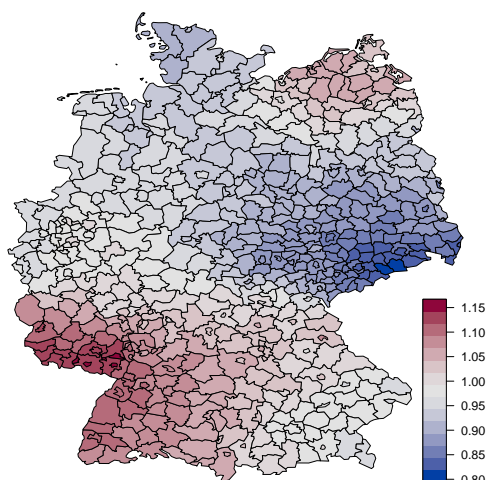


FIGURE 2.7: spatially structured effect at the mode for oral cancer

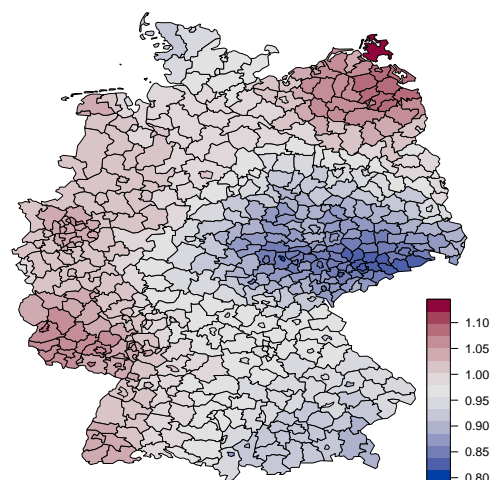


FIGURE 2.8: spatially structured effect at the mode for oesophagus cancer

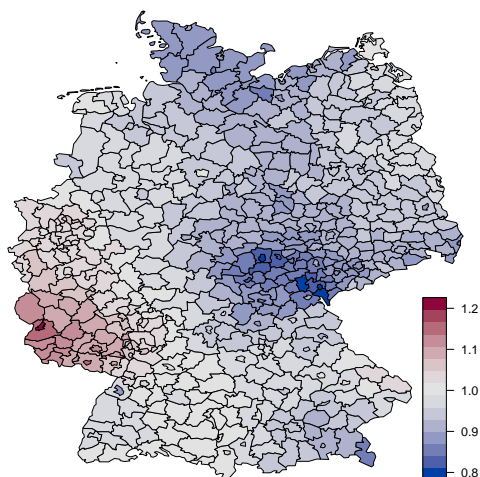


FIGURE 2.9: spatially structured effect at the mode for larynx cancer

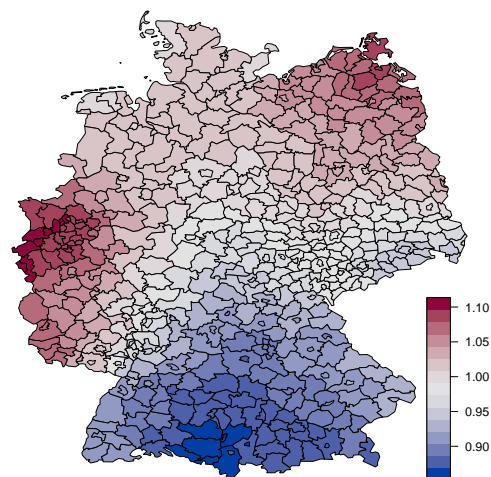


FIGURE 2.10: spatially structured effect at the mode for lung cancer

```

2 g = system.file("demodata/germany.graph", package="INLA")
3 g=inla.read.graph(g)
4 myQ=inla.graph2matrix(g)
5 myQ = -myQ
6 diag(myQ) = g$nnbs
7 Oral["region.unstruc"] ← 1:544 #v
8 # hyperprior for the structured effect
9 hyper_u = list( prec = list ( prior = "loggamma", param = c(3 , 1)))
10 # hyperprior for the unstructured effect
11 hyper_v = list( prec = list ( prior = "loggamma", param = c(3 , 1)))
12 Germany = or # data for oral cancer
13 Germany$z = sqrt(or$V3) # sqrt of the SMR

```

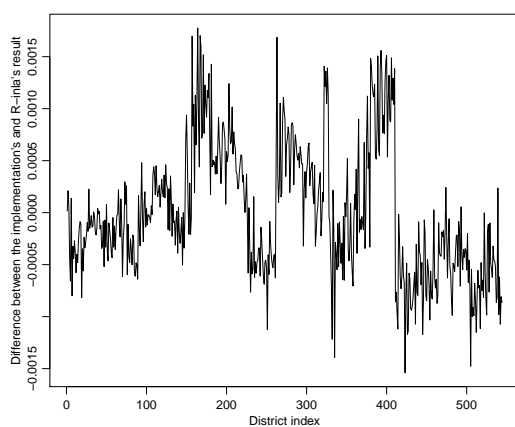


FIGURE 2.11: Difference between R-INLA and the implementation for the spatially structured component

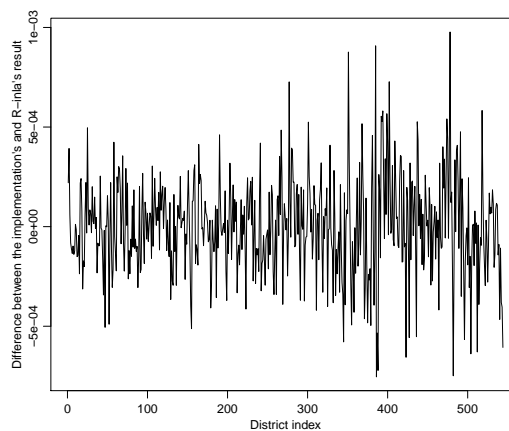


FIGURE 2.12: Difference between R-INLA and the implementation for the unstructured component

```

14 formula2 = z ~ -1 + f(region, model="besag", graph=g, constr=FALSE,
15 # besag for the structured effect
16 hyper=hyper_u)+f(Oral$region.unstruc, model="iid",
17 # iid for the unstructured effect
18 hyper=hyper_v)
19 result2= inla(formula2, family="gaussian", data=Germany,
20 control.inla = list(int.strategy="eb"),
21 # empirical bayes strategy to use the same method as in the
22 implementation
23 control.predictor = list(compute=TRUE),
24 control.family=list(hyper=list(prec=list(initial=log(4)),
25 fixed=TRUE)))

```

2.3 Disease mapping with penalized complexity priors

Some of the most popular disease mapping models have now been covered, and it is time to introduce penalized complexity priors in the models to come. In order to do so, a short introduction to PC priors is given together with a four principled method to derive them.

2.3.1 Preliminary Definitions

To understand fully what the PC priors are about it is paramount to comprehend the notions of base and flexible model.

Definition 2.2. Considering a model component with a probability density function $\pi(x|\xi)$ controlled by a flexibility parameter ξ , the **base model** denote the simplest model possible, in other word the value of ξ that makes the model as simple as possible. For clarity sake, the base model is the model for which $\xi = 0$. In terms of interpretation, the flexibility parameter ξ is chosen so that the base model “happens” when $\xi = 0$, not the other way around.

Now when $\xi \neq 0$, $\pi(x|\xi)$ can be interpreted as a flexible extension of the base model : a **flexible model**. As ξ increases, so does the deviation from the base model. For a more comprehensive definition see [Simpson et al. \[2015\]](#).

2.3.2 Penalized complexity priors

As explained in [Simpson et al. \[2015\]](#), penalized complexity priors have been designed for additive hierarchical model and built so that they satisfy the following criteria:

- The prior should not be non informative.
- Model structure aware prior.
- Changes in the model should impact the prior.
- An over specified model should see it’s flexibility limited by the prior.
- The prior should control what a parameter does, rather than its numerical value.
- Computational feasibility of the prior.
- Theory consistent prior.

Then again the focus of this project is not to explain why these priors are built the way they are, but rather to try and extend their use to joint disease mapping given that they have show promising results when using models with one disease and that they are designed for the very thing this project actually focuses on: additive models. The analysis of the underlying motivation for the list of criteria given above that led to the four principled constructive definition given in [Simpson et al. \[2015\]](#) is beyond the scope of this project, for more details see [Simpson et al. \[2015\]](#). The aforementioned four principled constructive definition (from [Simpson et al. \[2015\]](#)) is as follows:

- **First Principle: Occam’s razor.** Following the principle of parsimony, a simpler model should be preferred over a more complex one until there is enough

support, which means that the prior should penalize deviations from base model, see Definition 2.2, since $\pi(x, \xi)$ extends $\pi(x|\xi = 0)$, the prior for ξ should penalize the increase in complexity brought in by $\xi \geq 0$. The mode of the prior should be at the base model. In other words, the prior, without support, should favor the base model and the prior should decrease as a function of a measure of increased complexity between the flexible and the base model.

- **Second Principle: Measure of complexity.** In order to measure the increased complexity between the flexible model and the base model, the Kullback-Leibler divergence (KLD) defined in Kullback and Leibler [1951] is used. The KLD between the flexible model density f and the base model density g is defined as:

$$KLD(f||g) = \int_{-\infty}^{+\infty} f(x) \log\left(\frac{f(x)}{g(x)}\right) dx.$$

The distance between the two model is then $d(f||g) = \sqrt{2KLD(f||g)}$. As said in Definition 2.2, for a model component \mathbf{x} with a flexibility parameter ξ , the density base model is defined as $f(x) = \pi(x|\xi = 0), \forall x$, while the density of the flexible model is defined as $g(x) = \pi(x|\xi), \forall x$ for an arbitrary ξ . Hence $d(f||g)$ is a function of ξ and will be further referred to as $d(\xi)$.

- **Third Principle: Constant rate penalization.** To penalize the deviation from the base model that is parametrized with d , a constant decay rate r is used and satisfies:

$$\pi_d(d + \delta) = r\delta, \quad 0 \leq r \leq 1, \delta \geq 0.$$

Besides, the mode of the prior is at $d = 0$, i.e. the base model which implies $\pi(d) = \lambda e^{-\lambda d}$ with $r = e^{-\lambda}$. This corresponds to the following prior on the original space :

$$\pi(\xi) = \lambda e^{-\lambda d(\xi)} \left| \frac{\partial d(\xi)}{\partial \xi} \right|.$$

When d does not go to infinity as ξ goes to its upper bound, a truncated exponential is used.

- **Fourth Principle: User-defined scaling.** The fourth and final principle required to fully construct the prior selects λ and uses the user's idea of a sensible size for the parameter. It can be done by controlling the prior mass in the tail:

$$Prob(Q(\xi) \geq U) = \alpha.$$

where $Q(\xi)$ is an interpretable transformation of the parameter, U is an user defined upper bound that specifies what is meant by “tail event”, and α is the weight that is put on this “tail event”. This basically indicates how informative the prior is.

2.3.3 New model with PC priors

The previous model raises two main concerns according to [Simpson et al. \[2015\]](#), if the priors are to agree with Section 2.3.2:

- the marginal variance R_{ii}^{-1} of the spatial structured effect is not standardized which prevents from transferring the prior from one graph (say Germany’s graph) to another one (an other country) since the aforementioned marginal variance depends on the graph as said in [Simpson et al. \[2015\]](#).
- the structured u and unstructured v components should not be dealt with independently. Indeed according to [Simpson et al. \[2015\]](#), in the event that there are not any spatial dependency, it turns out that the unstructured effect is included in the structured effect.

The proposed re-parametrization that solve the two aforementioned main issues is found in [Simpson et al. \[2015\]](#):

$$\mu_i = \mu + \frac{1}{\sqrt{\tau}}(\sqrt{1 - \phi}v_i + \sqrt{\phi}u_i^*).$$

Let μ denote the intercept that is assumed to follow a normal distribution with a large variance ($\sigma_{max}^2 = 100$). Let v denote the unstructured effect that follows a multivariate normal distribution with mean 0 and precision I : $v \sim N(0, I)$. The precision matrix R of the structured effect in Equation 2.2 has a non empty kernel, namely the 1-vector, hence the density is invariant to adding a constant to u . The spatially structured effect needs a sum to zero constraint $1^T u = 0$ to prevent confounding it with the intercept μ . Finally let u^* denote the intrinsic gaussian markov random field with a sum to zero constraint and a scaled covariance matrix Q^* . Q^* scaled according to [Sorbye and Rue \[2014\]](#) to facilitate prior assignment, for more details refer to [Sorbye and Rue \[2014\]](#). The spatially structured component u^* is an intrinsic GMRF, it is going to be referred to as

$$u^* \sim N(0, Q^{*-1}),$$

even though it is rather improper because of the rank deficiency of the precision matrix Q^* . The INLA built in function `inla.scale.model` does the scaling, scaling that allows

prior interpretation to remain the same even if the graph is different, for more details see [Riebler et al. \[2016\]](#). This way, τ represents the precision of the marginal deviation from the null space while ϕ and $1 - \phi$ represent respectively the part of the variance explained by the spatially structured effect u^* and the unstructured effect v (for more details, see [Simpson et al. \[2015\]](#)). According to [Simpson et al. \[2015\]](#), κ and τ are orthogonal in terms of interpretation, which allows for an independent specification of their priors.

2.3.4 Derivation of the penalized prior for τ

The following derivation is done in [Simpson et al. \[2015\]](#).

Theorem 2.3. *The Kullback-Leibler divergence from $N_p^1(\mu_1, \Sigma_1)$ to the base case $N_p^0(\mu_0 = 0, \Sigma_0)$ is*

$$KLD(N_p^1 || N_p^0) = \frac{1}{2} (tr(\Sigma_0^{-1} \Sigma_1) + (\mu_0 - \mu_1)^\top \Sigma_0^{-1} (\mu_0 - \mu_1) - p - \ln(\frac{|\Sigma_1|}{|\Sigma_0|})). \quad (2.17)$$

The prior for τ is derived using the four principled definition in Section 2.3.2. The base model is defined to be when there is no spatial effect, that is when $\tau \rightarrow \infty$. Thus τ_0 , the precision for the base model is set such that $\tau_0 \gg 1$, and later on brought to $+\infty$. The flexible model has finite τ . For notation clarity, let A denote $\phi Q^{*-1} + (1 - \phi)I$ where ϕ is arbitrary. Then using Theorem 2.3 the Kullback-Leibler divergence between the flexible model $N_p^1(\mu_1 = 0, \Sigma_1 = A/\tau)$ and the base case $N_p^0(\mu_0 = 0, \Sigma_0 = A/\tau_0)$ is:

$$\begin{aligned} KLD(N_p^1 || N_p^0) &= \frac{1}{2} (tr(\Sigma_0^{-1} \Sigma_1) + -p - \ln(\frac{|\Sigma_1|}{|\Sigma_0|})), \\ &= \frac{1}{2} (\frac{\tau_0}{\tau} tr(I) + -p - \ln(\frac{\tau}{\tau_0})), \\ &= \frac{p}{2} \frac{\tau_0}{\tau} (1 - \frac{\tau}{\tau_0} - \frac{\tau}{\tau_0} \log(\frac{\tau}{\tau_0})). \end{aligned}$$

Now let $\frac{\tau_0}{\tau} \gg 1$, then:

$$KLD(N_p^1 || N_p^0) = \frac{p}{2} \frac{\tau_0}{\tau}.$$

Thus:

$$d(\tau) = \sqrt{\frac{p\tau_0}{\tau}}.$$

With a constant penalization rate, it follows that:

$$\pi(d) = \lambda \exp(-\lambda d).$$

Using the change of variable formula:

$$\begin{aligned}\pi(\tau) &= \lambda \exp(-\lambda \sqrt{\frac{p\tau_0}{\tau}}) \left| \frac{\partial d}{\partial \tau} \right|. \\ \pi(\tau) &= \frac{\lambda}{2} \sqrt{p\tau_0} \tau^{-\frac{3}{2}} \exp(-\lambda \sqrt{\frac{p\tau_0}{\tau}}).\end{aligned}$$

Let $\theta = \lambda \sqrt{p\tau_0}$, then:

$$\pi(\tau) = \frac{\theta}{2} \tau^{-\frac{3}{2}} \exp(-\theta \tau^{-\frac{1}{2}}). \quad (2.18)$$

The prior for τ thus follows a type-2 Gumbel distribution. Now is time to apply the user defined scaling principle to get a value for λ . The parameters (U, α) are to be chosen such that large deviations are unlikely:

$$\begin{aligned}P\left(\frac{1}{\sqrt{\tau}} \geq U\right) &= \alpha, \\ P\left(\frac{\sqrt{p\tau_0}}{\sqrt{\tau}} \geq U \sqrt{p\tau_0}\right) &= \alpha, \\ P(d \geq U \sqrt{p\tau_0}) &= \alpha, \\ 1 - F_d(U \sqrt{p\tau_0}) &= \alpha.\end{aligned}$$

Thus $\theta = -\ln(\alpha)/U$. Eventually:

$$\pi(\tau) = \frac{\theta}{2} \tau^{-\frac{3}{2}} \exp(-\theta \tau^{-\frac{1}{2}}), \quad (2.19)$$

with $\theta = -\ln(\alpha)/U$. It is useful, and eventually time saving to notice that the prior's nature does not depend on A . Thus further derivations of such spatial variance priors should be fairly similar. As a rule of thumb [Simpson et al., 2015], if $\alpha = 0.01$ then the standard deviation of the random component is approximately $0.31U$, it thus explains the choice of $\alpha = 0.01$ and $U = 0.2/0.31$.

2.3.5 Derivation of the PC prior for ϕ

The following derivation is done in Simpson et al. [2015]. The base case $\phi = 0$ is when all the variance is explained by the unstructured effect, that is when no spatial pattern can be found. The KLD between the base model $\mu_0 = 0$, $\Sigma_0 = I$ and the flexible model $\mu_1 = 0$, $\Sigma_1 = (1 - \phi)I + \phi R^{-1}$. It then follows from Theorem 2.3 that:

$$2KLD(\phi) = n\phi \left(\frac{1}{n} \text{tr}(Q^{*-1}) - 1 \right) - \log \left| (1 - \phi)I + \phi Q^{*-1} \right|.$$

Let the γ_i 's be the eigenvalues of Q^* . Let $\eta_i = \frac{1}{\gamma_i}$. Then it can be seen that:

$$|(1 - \phi)I + \phi Q^{*-1}| = \prod_{i=1}^n (1 - \phi - \phi \eta_i).$$

Note that if Q^* is not invertible, we just consider the non zero eigenvalues! Then applying the third principle and its exponential decay rate principle, and the change of variable formula:

$$\pi(\phi) = \lambda \exp(-\lambda d(\phi)) \left| \frac{\partial d(\phi)}{\partial \phi} \right|.$$

Since ϕ has to lie between 0 and 1, the exponential distribution has to be truncated as follows:

$$\pi(\phi) = \frac{\lambda e^{-\lambda d(\phi)}}{\Pi(1) - \Pi(0)} \left| \frac{\partial d(\phi)}{\partial \phi} \right|.$$

Thus:

$$\pi(\phi) = \frac{\lambda \exp(-\lambda d(\phi))}{1 - \exp(-\lambda d(1))} \left| \frac{\partial d(\phi)}{\partial \phi} \right|.$$

To determine λ , we set the degree of penalization to be $P(\phi \leq U) = \alpha$, thus:

$$\begin{aligned} P(\phi \leq U) &= \alpha, \\ \frac{1}{1 - \exp(-\lambda d(1))} \int_0^U \lambda \exp(-\lambda d(\phi)) \left| \frac{\partial d(\phi)}{\partial \phi} \right| d\phi &= \alpha, \\ \frac{1}{1 - \exp(-\lambda d(1))} \int_{d(0)}^{d(U)} \lambda \exp(-\lambda z) dz &= \alpha, \\ \frac{1 - \exp(-\lambda d(U))}{1 - \exp(-\lambda d(1))} &= \alpha. \end{aligned}$$

In practice however, the truncated exponential is used only if needed, that is only if $1 - \exp(-\lambda d(1)) \neq 1$. Then if $1 - \exp(-\lambda d(1)) = 1$ then the truncation is not required, and λ is easier to compute:

$$\pi(\phi) = \lambda \exp(-\lambda d(\phi)) \left| \frac{\partial d(\phi)}{\partial \phi} \right|, \quad (2.20)$$

with $\lambda = -\frac{\log(1-\alpha)}{d(U)}$.

2.3.6 Derivation of the posterior marginal $\pi(u, v, \mu|z)$

The quantity of interest is the posterior marginal $\pi(u^*, v, \mu|z)$ as in Section 2.2.2.1. Out of convenience, the method used to derive it is also the same as in Section 2.2.2.1, that is:

$$\pi(u^*, v, \mu|z) \approx \pi(u^*, v, \mu|z, (\phi, \tau)_{mode}),$$

where $(\phi, \tau)_{mode}$ denotes the mode of the posterior marginal $\pi(\phi, \tau|z)$. Out of convenience, the sum to zero constraint is omitted for now, the obtained formula will be adjusted a posteriori. The posterior marginal for ϕ, τ is then:

$$\pi(\phi, \tau|z) \propto \frac{\pi(z|u, v, \mu, \phi, \tau)\pi(u|\phi, \tau)\pi(v|\phi, \tau)\pi(\phi)\pi(\tau)\pi(\mu)}{\pi(u, v, \mu|z, \phi, \tau)}.$$

Then similarly to what has been done in Section 2.2.2.1, the numerator is known and the denominator $\pi(u, v, \mu|z, \phi, \tau)$ is a GMRF with a mean and a precision yet to be determined. It is going to be done using the following:

$$\begin{aligned} \pi(u, v, \mu, z|\phi, \tau) &\propto \pi(z|u, v, \mu, \phi, \tau)\pi(u|\phi, \tau)\pi(v|\phi, \tau)\pi(\mu|\sigma_{max}^2), \\ \pi(z|u, v, \mu, \kappa, \tau) &\propto e^{(-\frac{1}{2\sigma^2} \sum_{i=1}^n (z_i - \frac{\sqrt{\phi}}{\sqrt{\tau}} u_i - \frac{\sqrt{1-\phi}}{\sqrt{\tau}} v_i - \mu)^2)}, \\ \pi(u|\tau) &\propto e^{-\frac{u^T R u}{2}}, \\ \pi(v|\phi) &\sim N(0, I), \\ \pi(\mu|\sigma_{max}^2) &\sim N(0, \sigma_{max}^2). \end{aligned}$$

Thus $\pi(u, v, \mu, z|\phi, \tau)$ is as in Section 2.2.2.1 a GMRF, with mean 0 precision Q_2 . The precision matrix Q_2 is defined as follows:

$$Q_2 = \begin{bmatrix} Q_{uu} & Q_{uv} & Q_{u\mu} & Q_{uz} \\ Q_{vu} & Q_{vv} & Q_{v\mu} & Q_{vz} \\ Q_{\mu u} & Q_{\mu v} & Q_{\mu\mu} & Q_{\mu z} \\ Q_{zu} & Q_{zv} & Q_{z\mu} & Q_{zz} \end{bmatrix}.$$

The content of Q_2 's block matrices is found in Appendix B.1 From there, using Theorem 2.1, one can derive $\pi(u, v, \mu|z, \phi, \tau)$:

$$u, v, \mu|z, \phi, \tau \sim N_c(b = - \begin{bmatrix} Q_{uz} \\ Q_{vz} \\ Q_{\mu z} \end{bmatrix} \begin{pmatrix} z \\ z \\ z \end{pmatrix}, Q_1 = \begin{bmatrix} Q_{uu} & Q_{uv} & Q_{u\mu} \\ Q_{vu} & Q_{vv} & Q_{v\mu} \\ Q_{\mu u} & Q_{\mu v} & Q_{\mu\mu} \end{bmatrix}).$$

However, the sum to zero constraint was omitted from the proof for simplicity. $u, v, \mu, 1^T u = 0|z, \kappa, \tau$ actual mean and precision is obtain using the following theorem from [Rue and Held \[2005\]](#):

Theorem 2.4. *Let $x \sim N(\mu, Q_1^{-1})$, then $x|Ax = e \sim N(\mu^*, Q_1^{*-1})$ where*

$$\begin{aligned}\mu^* &= \mu - Q_1^{-1} A^T (A Q_1^{-1} A^T)^{-1} A \mu, \\ \Sigma^* &= Q_1^{-1} - Q_1^{-1} A^T (A Q_1^{-1} A^T)^{-1} A Q_1^{-1},\end{aligned}$$

where $\mu^* = Q_1^{*-1} b$ and $\mu = Q_1^{-1} b$.

Since $\pi(\phi, \tau|z)$ does not depend on u, v, μ . Then u, v, μ are chosen in a way that simplifies the function to maximize, namely $u = 0, v = 0, \mu = 0$. The multiplicative constants that will not change the result of the optimization are taken out of the equation. It then simplifies to:

$$\pi(\phi, \tau|z) \propto \frac{\pi(\phi)\pi(\tau)}{\pi(0, 0, 0|z, \phi, \tau)}.$$

The R `optim` function then compute $(\phi, \tau)_{mode}$. Now since:

$$\pi(u^*, v, \mu|z) \approx \pi(u^*, v, \mu|z, (\phi, \tau)_{mode}),$$

then $u^*, v, \mu|z \sim N(\mu^*, \Sigma^*)$ where μ^*, Σ^* are evaluated at the mode. Since the goal of this project is not to focus on the user define scaling choice, the values for α and U are taken from [Riebler et al. \[2016\]](#):

$$\begin{aligned}\alpha_\tau &= 0.01, & U_\tau &= 0.2/0.31, \\ \alpha_\phi &= 0.5, & U_\phi &= 2/3.\end{aligned}$$

2.3.7 Implementation tricks and difficulties

When implementing this model in R, two issues were encountered:

- When using the `optim` function in R, the BFGS method was used as the go to method, however with larger problem in later models, the use of BFGS causes the machine to overflow. Hence the use of the L-BFGS-B method, that is a limited memory quasi Newton method that essentially stores only a few vectors that describe the Hessian implicitly rather than storing a dense Hessian matrix as said in [Nocedal and Wright \[2006\]](#), that does not cause the machine to overflow. It was

quite surprising to find out that that 5 variables to optimize caused the machine to overflow, but since L-BFGS-S brought an efficient solution to the problem, the problem was not looked further into since optimization is not in the scope of the project.

- When trying to maximize the posterior marginal $\pi(\phi, \tau|z)$, the parameter space is constrained as follows:

$$\tau \in]0, +\infty[,$$

$$\phi \in [0, 1].$$

Besides a constrained parameter space requires constrained optimization technique rather than plain optimization ones. This is a rather unnecessary complication. To get past it, let:

$$x = \log(\tau) \quad \forall \tau,$$

$$y = \text{logit}(\phi) \quad \forall \phi,$$

then:

$$\log(\tau) \in]-\infty, +\infty[,$$

$$\text{logit}(\phi) \in]-\infty, +\infty[.$$

With this change of variables, the parameter space is not constraint anymore. Then obviously the change of variable formula has to be applied to probability density function of the parameters as seen in Appendix [A.3.3](#).

$$\pi_X(x) = \pi_\tau(\exp x) \left| \frac{\partial \exp x}{\partial x} \right|,$$

$$\pi_Y(y) = \pi_\phi\left(\frac{\exp y}{1 + \exp y}\right) \left| \frac{\partial \frac{\exp y}{1 + \exp y}}{\partial y} \right|.$$

However when analyzing results, the internal scale does not provide easily interpretable results, only a basis for comparison between the implementation and R-INLA whereas the original scale does, so the latter is use for analysis. Even though the original scale is used, the values at the mode are those of x and y . They are just transformed back to the original scale. In other words the values are those given by $\arg \max \pi(\log(\tau), \text{logit}(\phi)|z)$ and not $\arg \max \pi(\tau, \phi|z)$, but they are often transformed back to the original scale to be more easily interpreted. In later models, this method is also used, so whenever a parameter has a domain comprised

between 0 and 1 a logit transformation is applied and whenever a parameter has for domain the positive real numbers, a log transformation is applied.

2.3.8 Results and comparison to R-INLA

	R-INLA	Implementation
Log(τ)	7.15	7.15
Logit(ϕ)	-0.64	-0.64

TABLE 2.2: argmax of the posterior marginal $\pi(\phi, \tau|z)$ for oral cancer

When comparing the results from the maximization (code in Appendix A.3.3) of the posterior marginal $\pi(\text{logit}(\phi), \log(\tau)|z)$ to R-INLA's result, see Table 2.2, it is seen that they are identical. The code used to obtain the R-INLA results is given below.

```

1 Germany = or #data for oral cancer
2 Germany$z ← sqrt(or$V3)
3 z = sqrt(or$V3)
4 formula3 = z ~ f(region, model="bym2", graph=g,
5 scale.model = TRUE, # scaled matrix
6 rankdef = 1, # rank deficiency set to 1
7 hyper = list(prec = list(prior="pc.prec", param=
  prec.par, initial=5),
8 theta2 = list(prior = "pc", param=
  phi.par,
9 initial = -3)), constr
  = TRUE), # constrained spatial effect
10 result3=inla(formula3, family="gaussian", data=Germany,
11 control.inla = list(strategy="gaussian",
12 int.strategy="eb", # empirical bayes strategy
13 # to fit the method that's been used in the
  implementation.
14 tolerance = 1e-12, h=1e-4),
15 control.family=list(hyper=list(prec=list(initial=log(4),
  fixed=TRUE))),
16 control.fixed=list(prec.intercept=(prec.mean)))

```

Similarly, it is seen in Figure 2.13 that the difference at the mode of the posterior marginal $\pi(u, v, \mu|z)$ between R-INLA and the implementation is not significant, and thus R-INLA's results check out the implementation's. As in shown in Figure 2.13, there is an area of high risk in southern Germany and an area of moderate risk in north eastern Germany.

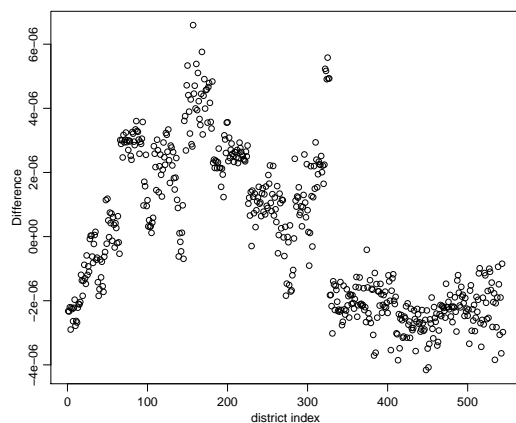


FIGURE 2.13: $\frac{1}{\sqrt{\tau}}(\sqrt{1-\phi}v_i + \sqrt{\phi}u_i^*)$ difference at the mode between R-INLA and the simulation for oral cancer

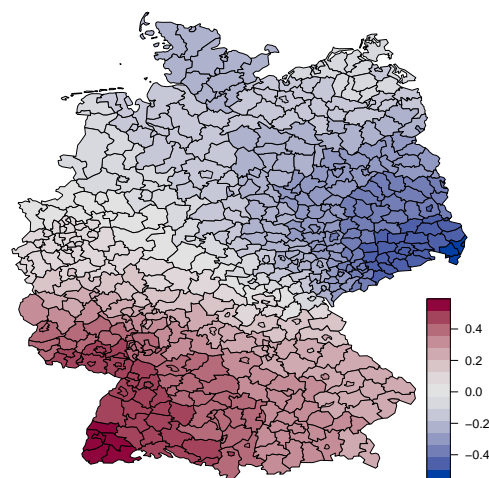


FIGURE 2.14: spatial effect at the mode of the posterior marginal for oral cancer

Chapter 3

Joint disease mapping with penalized complexity priors

In the previous chapter, models taking only one disease into account have been reviewed. In this chapter, the goal is to model two diseases jointly. Following [Held et al. \[2005\]](#), a shared spatial component is introduced together with disease specific component. The shared spatial component acts as a surrogate for an unobserved spatially structured covariate that affects both diseases, while the disease specific component acts as a surrogate for an unobserved structured covariate that affects only one of the disease. In practice, when considering oral cancer and lung cancer jointly, the shared component represents the common risk factors of these two diseases, which is mainly smoked tobacco, see [Section 1.2](#), and any other risk factor they have in common. Besides, as alcohol consumption seems to be a risk factor only for lung cancer, the disease specific component of lung cancer would account for alcohol consumption. PC priors have not yet been used in such a setting, and the underlying idea is to try and assess whether PC priors can be successfully used for joint disease mapping. [Riebler et al. \[2016\]](#) shows that the univariate disease mapping model presented in [Section 2.3](#) (under the assumption that the y_i 's follow a poisson distribution) behaves at least as well as the other popular models and possesses an ability to shrink to a simpler model when the data requires it, in other words, the mean value of the posterior marginal of a parameter can go to zero if the data suggests that the component this parameter scales should not be there. Finding evidence that a model using PC priors and models jointly two diseases can behave as convincingly and possesses such a shrinking property is what this chapter ideally aims for.

For all the models considered in this chapter it is assumed that:

$$\begin{aligned} z_{1,i} &\sim N(\eta_{1,i}, \sigma_1^2), \\ z_{2,i} &\sim N(\eta_{2,i}, \sigma_2^2), \end{aligned}$$

where $z_{1,i}$ denotes the squared root of the SMR for the i -th disease. As said before, since the work done on the use of PC priors when jointly modeling two disease has not been done before, it is assumed that the square root of the SMR follow a normal distribution. If the results under that assumption show encouraging promises, modeling using Poisson distributions should then be done.

3.1 First proposed model with PC priors

The following model combines the model presented in Section 2.3 and the shared-specific components mentioned in Held et al. [2005]. This model is designed for two diseases who share a common risk factor, with one of the diseases having an additional risk factor that the other does not have. Thus it would be well suited to jointly model lung and any of the other cancers, but not to model diseases that have two common risk factors, namely any pairing of larynx oral and oesophagus cancer, not that it would not work but it would be impossible to obtain the spatial patterns corresponding the each of the two diseases. Rather the spatial pattern of the two disease combined would be obtained. Spatial pattern that the model in Section 2.3 already provides. The model is the result of a discussion whose point was to determine how to build a model that would allow the joint modeling of two diseases and the use of the PC priors. It was not designed to be perfect but to provide a good basis on which to further elaborate. The parameters τ_1 and τ_2 model the marginal deviation from the null space for each of the two considered diseases. The parameter ϕ_1 models the part of the variance explained by the disease specific structured effect for the first disease while $1 - \phi_1$ represents the part of the variance explained by the shared component. The parameters γ and $1 - \gamma$ represent the part of the variance explained by respectively the structured effect of the shared component and the unstructured effect of the shared component. As for ϕ_2 and $1 - \phi_2$, they represent respectively the part of the variance explained by the structured and unstructured effect for the second disease. As in Section 2.3, the precision parameters and the weights are considered to be independent in interpretation and which allows for

independent specifications of the priors. The model is thus as follows:

$$\eta_1 = \mu_1 + \frac{1}{\sqrt{\tau_1}} \left(\sqrt{1 - \phi_1} (\sqrt{\gamma} u_1^* + \sqrt{1 - \gamma} v_1) + \sqrt{\phi_1} u_2^* \right), \quad (3.1)$$

$$\eta_2 = \mu_2 + \frac{1}{\sqrt{\tau_2}} \left(\sqrt{\phi_2} u_1^* + \sqrt{1 - \phi_2} v_1 \right), \quad (3.2)$$

where μ_1 and μ_2 are the disease specific intercepts, and follow a normal distribution with a large variance and mean zero, u_1^* , the shared component, is a scaled intrinsic GMRF with a sum to zero constraint, u_2^* , the disease specific component, is a scaled intrinsic GMRF with a sum to zero constraint. The component $v_1 \sim N(0, I)$ is the unstructured component for both diseases.

3.1.1 Penalized complexity priors derivation

When deriving the density of a prior, the other parameters are set arbitrarily.

PC prior for τ_1

The parameter τ_1 represents the deviation from the null space for the first disease and can be derived independently of the second disease. The prior for τ_1 is derived using the four principled definition in Section 2.3.2. The base model is defined as the least complex model, that is when $\tau_0 \rightarrow \infty$. The flexible model has finite τ_1 . Let A denote $((1 - \phi_1)\gamma + \phi_1 Q^{*-1}) + (1 - \phi_1)(1 - \gamma)I$. The choice of the other parameters, here γ and ϕ_1 , do not change the formula for the prior as the prior is derived considering an arbitrary γ and ϕ_1 . As said in Section 2.3.4, the prior specification does not depend on A , thus the prior for τ_1 is:

$$\pi(\tau_1) = \frac{\theta_1}{2} \tau_1^{-\frac{3}{2}} e^{(-\theta_1 \tau_1^{-\frac{1}{2}})},$$

with $\theta_1 = -\log(\alpha_1)/U_1$. The user define scaling is chosen identical to that of τ in Section 2.3.4 with $\alpha_1 = 0.01$ and $U_1 = 0.2/0.31$.

PC prior for τ_2

The precision parameter τ_2 represents the deviation from the null space for the second disease and can be derived independently of the first disease. What was said for τ_1 is also true for τ_2 , thus:

$$\pi(\tau_2) = \frac{\theta_2}{2} \tau_2^{-\frac{3}{2}} e^{(-\theta_2 \tau_2^{-\frac{1}{2}})},$$

with $\theta_2 = -\log(\alpha_2)/U_2$ where $\alpha_2 = 0.01$ and $U_2 = 0.2/0.31$.

Penalized complexity prior for ϕ_1

When $\phi_1 = 0$, the base model assumption is that both diseases share the same risk factor. When $\phi_1 \neq 1$, the flexible model allow for disease 1 to have spatial structured specific risk factors. The prior is derived following the method used in Section 2.3.2:

- **Base case** $N(0, \Sigma_0 = \gamma Q^{*-1} + (1 - \gamma)I)$.
- **Flexible case** $N(0, \Sigma_1 = ((1 - \phi_1)\gamma + \phi_1)Q^{*-1} + (1 - \phi_1)(1 - \gamma)I)$.

Where γ is arbitrary. Using Equation 2.3, the KLD between the base model and the flexible one is then:

$$KLD(\phi_1) = \frac{1}{2}(tr(\Sigma_0^{-1}\Sigma_1) - n - \ln(\frac{|\Sigma_1|}{|\Sigma_0|})).$$

Hence:

$$d(\phi_1) = \sqrt{(tr(\Sigma_0^{-1}\Sigma_1) - n - \ln(\frac{|\Sigma_1|}{|\Sigma_0|}))}.$$

The rest is the same as in Section 2.3.5, thus :

$$\pi(\phi_1) = \frac{\lambda_3 \exp(-\lambda_3 d(\phi))}{1 - \exp(-\lambda_3 d(1))} \left| \frac{\partial d(\phi)}{\partial \phi} \right|.$$

where $\frac{1 - \exp(-\lambda_3 d(U_3))}{1 - \exp(-\lambda_3 d(1))} = \alpha_3$ with some probability α_3 and an upper bound U_3 . Practice has shown that this prior needs to be truncated, e.g $1 - \exp \lambda_3 d(1) \neq 1$, however for computational reasons, namely computational speed, rather than solving $\frac{1 - \exp(-\lambda_3 d(U_3))}{1 - \exp(-\lambda_3 d(1))} = \alpha_3$, λ_3 is computed as if the prior did not need to be truncated as in Section 2.3.5 and then a posteriori divided by $1 - \exp(-\lambda_3 d(1))$. This is further detailed in Section 3.1.3. The prior now depends on γ , and so does λ_3 and it thus cannot be overlooked anymore when computing the mode of the posterior marginal $\pi(\gamma, \phi_1, \phi_2, \tau_1, \tau_2 | z)$.

Penalized complexity prior for ϕ_2

The second disease with no disease specific component is the same as in Section 2.3, thus the penalized complexity prior for ϕ_2 is the same as for ϕ in Section 2.3:

$$\pi(\phi_2) = \lambda_4 \exp(-\lambda_4 d(\phi_2)) \left| \frac{\partial d(\phi_2)}{\partial \phi_2} \right|,$$

where $\lambda_4 = -\frac{\ln(1 - \alpha_4)}{d(U_4)}$ with some probability α_4 and an upper bound U_4 .

Penalized Complexity prior for γ

When $\gamma = 0$, the base model assumption is that the two diseases have no shared spatially structured component.

- **Base case** $N(0, \Sigma_0 = (1 - \phi_1)I + \phi_1 Q^{*-1})$.
- **Flexible case** $N(0, \Sigma_0 = ((1 - \phi_1)\gamma + \phi_1)Q^{*-1} + (1 - \phi_1)(1 - \gamma)I)$,

where ϕ_1 is arbitrary. Using Equation 2.3, the KLD from the base model to the flexible one is then :

$$KLD(\phi_2) = \frac{1}{2}(tr(\Sigma_0^{-1}\Sigma_1) - n - \ln(\frac{|\Sigma_1|}{|\Sigma_0|})).$$

Hence :

$$d(\phi_2) = \sqrt{(tr(\Sigma_0^{-1}\Sigma_1) - n - \ln(\frac{|\Sigma_1|}{|\Sigma_0|}))}.$$

Then :

$$\pi(\gamma) = \frac{\lambda_5 \exp(-\lambda_5 d(\gamma))}{1 - \exp(-\lambda_5 d(1))} \left| \frac{\partial d(\gamma)}{\partial \phi} \right|.$$

where $\frac{1 - \exp(-\lambda_5 d(U_5))}{1 - \exp(-\lambda_5 d(1))} = \alpha_3$ with some probability α_5 and an upper bound U_5 . To compute λ_5 the method used is the same as the one use for ϕ_1 .

3.1.2 Posterior marginal $\pi(u, v, \mu|z)$

The quantity of interest is the same as before $\pi(u_1^*, u_2^*, v_1, \mu|z)$, and its derivation is done using the material in Section 2.2.2.1. Let u_1 and u_2 denote the scaled intrinsic GMRF, and u_1^* and u_2^* the scale intrinsic GMRF with the sum to zero constraint. Let $z_{1,:}$ denote $(z_{1,1}, \dots, z_{1,544})$ and $z_{2,:}$ denote $(z_{2,1}, \dots, z_{2,544})$.

$$\pi(u_1^*, u_2^*, v_1, \mu|z) \approx \pi(u_1^*, u_2^*, v_1, \mu|z, (\tau_1, \tau_2, \phi_1, \phi_2, \gamma)_{mode}). \quad (3.3)$$

where $(\tau_1, \tau_2, \phi_1, \phi_2, \gamma)_{mode} = \arg \max_{\tau_1, \tau_2, \phi_1, \phi_2, \gamma} \pi(\tau_1, \tau_2, \phi_1, \phi_2, \gamma|z_1, z_2)$ with

$$\begin{aligned} \pi(\tau_1, \tau_2, \phi_1, \phi_2, \gamma|z_{1,:}, z_{2,:}) &\propto \pi(z_{1,:}|\dots)\pi(z_{2,:}|\dots) \\ &\times \frac{\pi(u_{1,2}^*|\dots)\pi(v_1|\dots)\pi(\mu_{1,2})\pi(\tau_1, \tau_2, \phi_1, \phi_2, \gamma)}{\pi(u_1^*, u_2^*, v_1, \mu|z, \tau_1, \tau_2, \phi_1, \phi_2, \gamma)}. \end{aligned}$$

Let u_1 and u_2 denote the scaled intrinsic GMRF without the sum to zero constraint. Then it follows that $\pi(u_1, u_2, v_1, \mu, z | \tau_1, \tau_2, \phi_1, \phi_2, \gamma)$ is a GMRF with mean 0 and precision Q . Then Theorem 2.1 gives the following result:

$$u_1, u_2, v_1 | z, \tau_1, \tau_2, \phi_1, \phi_2, \gamma \sim N_c(b, Q_{GMRF}),$$

where

$$Q_{GMRF} = \begin{bmatrix} Q_{u_1 u_1} & Q_{u_1 u_2} & Q_{u_1 v_1} & Q_{u_1 \mu_1} & Q_{u_1 \mu_2} \\ & Q_{u_2 u_2} & Q_{u_2 v_1} & Q_{u_2 \mu_1} & Q_{u_2 \mu_2} \\ & & Q_{v_1 v_1} & Q_{v_1 \mu_1} & Q_{v_1 \mu_2} \\ & & & Q_{\mu_1 \mu_1} & Q_{\mu_1 \mu_2} \\ & & & & Q_{\mu_2 \mu_2} \end{bmatrix}.$$

The content of the block matrix is detailed in Appendix B.2. The linear term is:

$$b = \begin{pmatrix} \tau_{\epsilon_1} \frac{\sqrt{(1-\phi_1)\gamma}}{\sqrt{\tau_1}} y_1 + \tau_{\epsilon_2} \frac{\sqrt{\phi_2}}{\sqrt{\tau_2}} y_2 \\ \tau_{\epsilon_1} \frac{\sqrt{\phi_1} y_1}{\sqrt{\tau_1}} \\ \tau_{\epsilon_1} \frac{\sqrt{(1-\gamma)(1-\phi_1)} y_1}{\sqrt{\tau_1}} + \tau_{\epsilon_2} \frac{y_2}{\sqrt{\tau_2}} \\ \tau_{\epsilon_1} \Sigma z_{1,i} \\ \tau_{\epsilon_2} \Sigma z_{2,i} \end{pmatrix}.$$

Now that the GMRF for the unconstrained effect has been derived, it is easy to derived the same GMRF with the sum to zero constraints on the spatially structured effect using Theorem 2.4. Let

$$A \begin{pmatrix} u_1 \\ u_2 \\ v_1 \\ \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} 1^T u_1 \\ 1^T u_2 \end{pmatrix} = 0,$$

and $\mu = Q_{GMRF}^{-1} b$. Then:

$$\pi(u_1^*, u_2^*, v_1 | z, \tau_1, \tau_2, \phi_1, \phi_2, \gamma) \sim N_c(\mu^*, Q_{GMRF}^*),$$

where

$$\begin{aligned} \mu^* &= \mu - Q_{GMRF}^{-1} A^T (A Q_{GMRF}^{-1} A^T)^{-1} A \mu, \\ Q_{GMRF}^{*-1} &= Q_{GMRF}^{-1} - Q_{GMRF}^{-1} A^T (A Q_{GMRF}^{-1} A^T)^{-1} A Q_{GMRF}^{-1}. \end{aligned}$$

The optimization is done by setting $u_1 = u_2 = v_1 = v_2 = \mu = 0$. Then the function to optimize to obtain $(\tau_1, \tau_2, \phi_1, \phi_2, \gamma)_{mode}$ reduce to :

$$f(\tau_1, \tau_2, \phi_1, \phi_2, \gamma) = \frac{\pi(\tau_1, \tau_2, \phi_1, \phi_2, \gamma)}{\pi(u_1^* = 0, u_2^* = 0, v_1 = 0, \mu = 0 | z, \tau_1, \tau_2, \phi_1, \phi_2, \gamma)}.$$

After maximizing f, the only thing left do to is to plug $(\tau_1, \tau_2, \phi_1, \phi_2, \gamma)_{mode}$ in Equation 3.3 to get the posterior marginal $\pi(u_1^*, u_2^*, v_1, \mu | z)$

3.1.3 Scaling and computing the priors

When the model considered only one disease at a time, truncating was not relevant because “ λ ” and $d(1)$ did not depend on any other parameter, so as far as the maximization of the posterior marginal for the parameters was concerned, dividing by $1 - \exp(-\lambda d(1))$ came down to dividing by a constant which did not impact on the result of the optimization. However now that the model considers two diseases, $1 - \exp(-\lambda d(1))$ is actually a function of γ (or ϕ_1 depending on which prior is considered) and as such cannot be overlooked.

Scaling the prior for ϕ_1

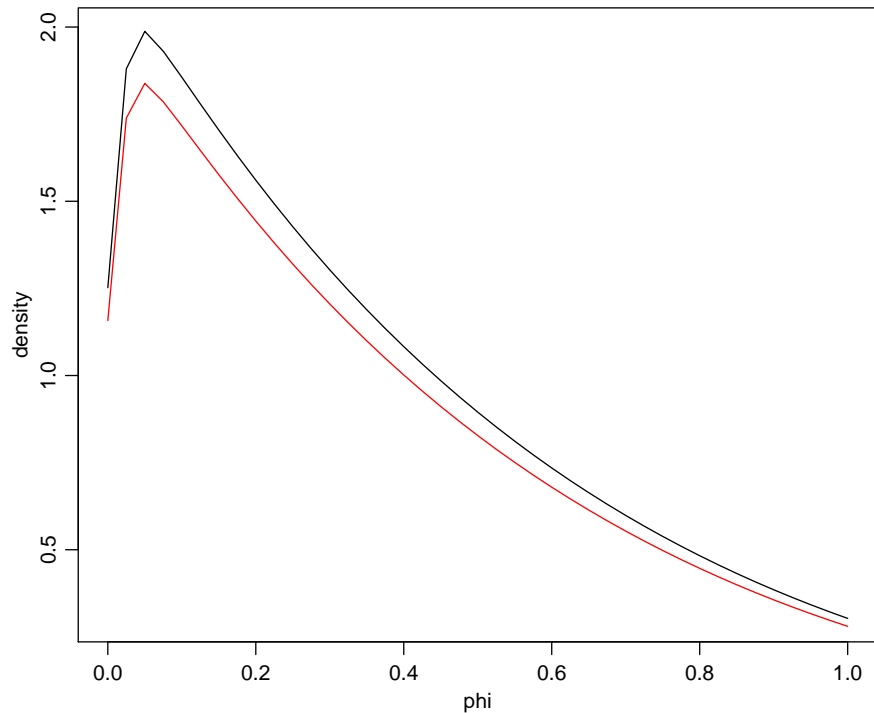


FIGURE 3.1: Distribution of ϕ before scaling (red) and after scaling (black) for $\gamma = 0.5$

As said in Section 3.1.1, the prior for ϕ_1 requires truncation since $1 - \exp(-\lambda d(1)) \neq 1$ for an arbitrary γ . Since the problem is not addressed by solving

$$\frac{1 - \exp(-\lambda d(U_3))}{1 - \exp(-\lambda d(1))} = \alpha_3 \quad (3.4)$$

numerically, but rather by dividing by $1 - \exp(-\lambda d(1))$ after having computed the value for λ using $\lambda = -\frac{\ln(1-\alpha)}{d(U)}$, it seems only natural to check whether the proposed method works. As seen in Figure 3.1 there is a substantial offset between the scaled and the unscaled prior. As a matter of fact, to check that the scaling does what it is meant to

	unscaled	scaled
value	0.91	0.99

TABLE 3.1: Integral of the prior for ϕ_1 from 0 to 1 with $\gamma = 0.5$

do, the integral of the prior is computed using the middle point method, see Appendix A.4.1, and as is shown by Table 3.1 scaling does force the integral of the prior to 1.

Scaling the prior for γ

	unscaled	scaled
value	0.91	0.99

TABLE 3.2: Integral of the prior for γ from 0 to 1 with $\phi_1 = 0.5$

Table 3.2 shows evidence that the scaling method that is used (see Appendix A.4.2) also produce satisfying result for the prior for γ . In further models, this will not be covered again. It just seemed important to at least show once that the method used was accurate enough.

Computing the priors

Computing the priors for ϕ_1 and γ is the most (and only) time consuming part of the computation mainly because for each prior density call, there are four calls of the KLD computing function and these prior densities have to be computed for each iteration of the optimization loop. And each KLD call requires to compute $\text{tr}(\Sigma_0^{-1}\Sigma_1)$. The prior for γ is taken as an example, roughly the same applies to the prior for ϕ_1 , and for ϕ_1 ϕ_2 and γ in the ulterior model in Section 3.2. In order to compute the prior for γ , where:

$$\begin{aligned} \Sigma_0 &= (1 - \phi_1)\gamma I + \phi_1 Q^{*-1}, \\ \Sigma_1 &= ((1 - \phi_1)\gamma + \phi_1)Q^{*-1} + (1 - \phi_1)(1 - \gamma) * I, \end{aligned}$$

Thus $\text{tr}(\Sigma_0^{-1}\Sigma_1)$ has to be computed. The problem is that Σ_0 is not sparse, and classical computationally efficient sparse matrix inversion method cannot be used. Thus the algorithm resorts to the all but usual `solve` built in function in R.

```

1 phi1 = 0.3
2 gamma = 0.6
3 # inv is the generalized inverse of Q
4 Ident = diag(544)
5 sigma_0 = (1-phi1)*Ident+phi1*inv
6 sigma_1 = (1-phi1)*(1-gamma)* Ident + ((1-phi1)*gamma+phi1)*inv
7 ptm <- proc.time()
8 tr_M = sum(diag(solve(sigma_0)%*%sigma_1))
9 proc.time() - ptm
10 user      system      time spent
11      0.441      0.004      0.450

```

As is seen in the above code excerpt, it takes roughly half a second which is far too much. If a method were to be found to compute this efficiently the code would become substantially faster.

3.1.4 Results

The code used in this section can be found in Appendix [A.4.3](#). The parameters are chosen as follows:

$$\alpha_{3:5} = 0.5,$$

$$U_{3:5} = 2/3.$$

Spatial pattern for the shared component

From Figures [3.2](#), [3.3](#), [3.4](#), it seems that the shared spatial components between lung cancer and the other three cancers are fairly similar which only makes sense, since all these cancers have tobacco as a risk factor as said in Section [1.2](#), and a major one at this. So it is expected that the shared components for all these pairings be similar. The shared component for oral and lung cancer, seen in Figure [3.2](#), presents with an area of increased risk in southern Germany, which the other pairings do not, see Figures [3.3](#), [3.4](#). As the similarity of the spatial patterns for Figures [3.2](#), [3.3](#), [3.4](#) are explained by the main risk factor they all have in common, namely tobacco, the differences of the spatial patterns can be explained by risk factors they do not all share. Thus this area

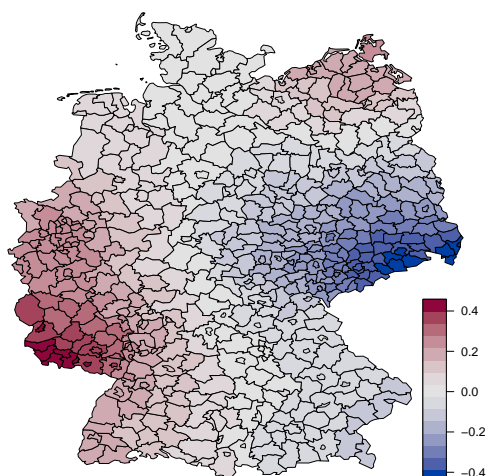


FIGURE 3.2: Shared spatial component for oral and lung cancers at the mode

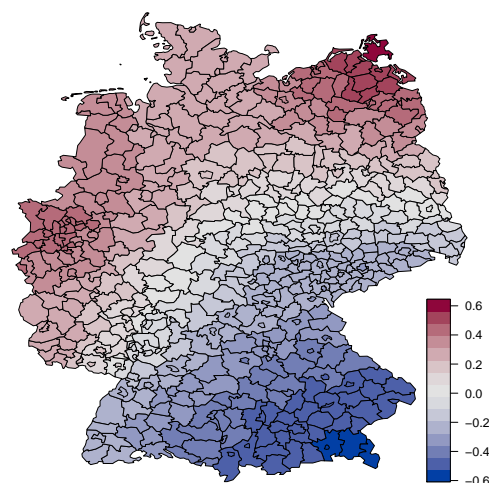


FIGURE 3.3: Shared spatial component for oesophagus and lung cancers at the mode

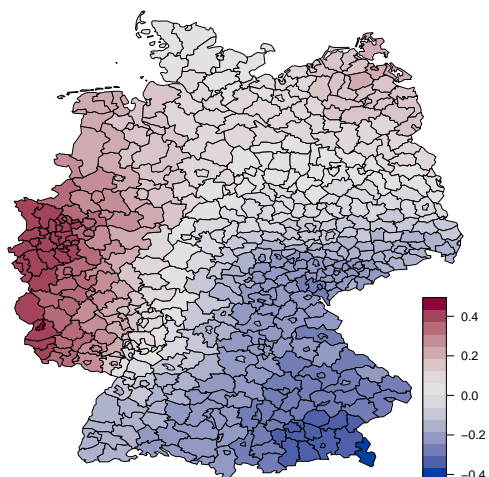


FIGURE 3.4: Shared spatial component for larynx and lung cancers at the mode

of increased risk might be evidence that there is a risk factor that only oral and lung cancer share.

Spatial pattern for the disease specific component

Alcohol consumption is said to be one of the two (tobacco is the other) main risk factors for oral, larynx and oesophagus cancer, but not for lung cancer as said in Section 1.2. From Figures 3.5, 3.6, 3.7, it is seen that all three disease specific component have fairly different patterns, which seems to suggest that alcohol consumption is not the only disease specific significant risk factor at play. If it were true, then Figures 3.5,

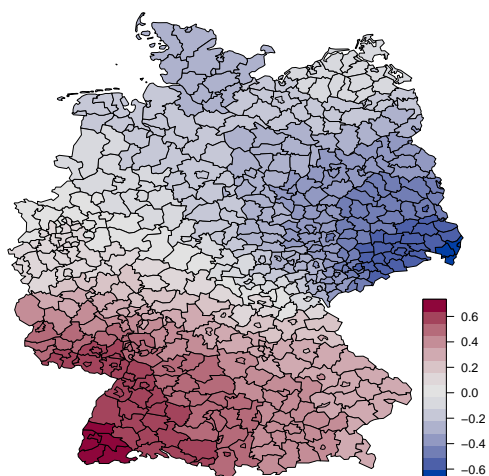


FIGURE 3.5: Disease specific component of oral cancer with lung cancer at the mode

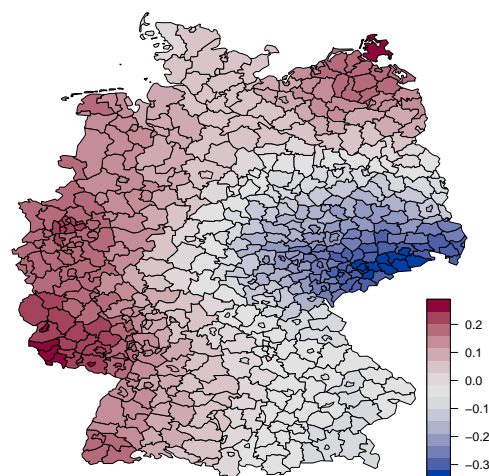


FIGURE 3.6: Disease specific component of oesophagus cancer with lung cancer at the mode

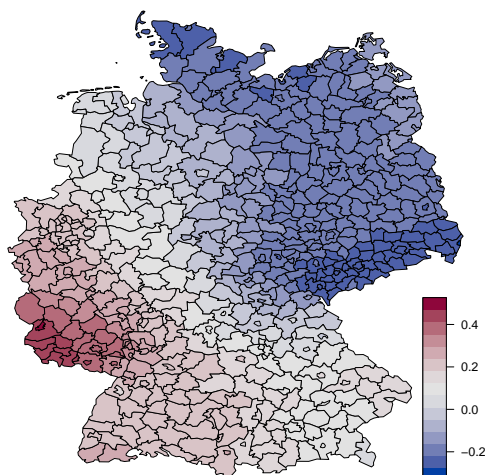


FIGURE 3.7: Disease specific component of larynx cancer with lung cancer at the mode

3.6, 3.7 would be fairly similar because then the disease specific component would only account for alcohol. These spatial patterns are quite similar to those obtained in [Held et al. \[2005\]](#). An in-depth analysis of these spatial patterns is beyond the scope of this project, however an epidemiologist might want to look further into it.

3.1.4.1 Tests and analysis

In order to further assess whether the model is capable of identifying correctly the component, a new data set is artificially created using the modes of the posterior marginals for $u_1, u_2, v_1, v_2, \mu_1, \mu_2$ and $\text{logit}(\phi_1), \text{logit}(\phi_2), \log(\tau_1), \log(\tau_2), \text{logit}(\gamma)$ for Oral-Lung cancer.

The observations are generated as follows:

$$z_{1,i,new} = \eta_{1,i,mode},$$

$$z_{2,i,new} = \eta_{2,i,mode},$$

where

$$\eta_{1,m} = \mu_{1,m} + \frac{1}{\sqrt{\tau_{1,m}}} \left(\sqrt{1 - \phi_{1,m}} (\sqrt{\gamma_m} u_{1,m}^* + \sqrt{1 - \gamma_m} v_{1,m}) + \sqrt{\phi_{1,m}} u_{2,m}^* \right),$$

$$\eta_{2,m} = \mu_{2,m} + \frac{1}{\sqrt{\tau_{2,m}}} \left(\sqrt{\phi_{1,m}} u_{1,m}^* + \sqrt{1 - \phi_{1,m}} v_{1,m} \right),$$

with:

$$(u_1, u_2, v_1, v_2, \mu_1, \mu_2)_m = \arg \max \pi(u_1, u_2, v_1, v_2, \mu_1, \mu_2 | z_1, z_2),$$

$$(\logit(\phi_1), \logit(\phi_2), \log(\tau_1), \log(\tau_2), \logit(\gamma))_m = \arg \max \pi(\logit(\phi_1), \logit(\phi_2), \log(\tau_1), \log(\tau_2), \logit(\gamma) | z_1, z_2).$$

Then, the posterior marginal for the parameters $\pi(\logit(\phi_1), \logit(\phi_2), \log(\tau_1), \log(\tau_2), \logit(\gamma) | z_1, z_2)$ is computed. It is expected that this ‘‘back estimation’’ produces results close to $(\logit(\phi_1), \logit(\phi_2), \log(\tau_1), \log(\tau_2), \logit(\gamma))_{mode}$ thus proving the capacity of the model to identify the underlying spatial structures. The results are displayed in Table 3.6. As is seen in the last column of Table 3.4, the back estimates underestimate

	argmax	back est	ratio back est / argmax
$\sqrt{\phi_1}$	0.56	0.64	0.87
$\sqrt{\phi_2}$	0.46	0.48	0.96
$\sqrt{\tau_1}$	50.98	29.41	1.73
$\sqrt{\tau_2}$	50.75	47.25	1.07
$\sqrt{\gamma}$	0.56	0.61	0.92

TABLE 3.3: Column 1 : Argmax of the posterior marginal Oral-Lung Cancer, Column 2 : argmax of the posterior marginal back estimation, Column 3 : Ratio of the latter columns

ϕ_1, ϕ_2, ϕ_3 , greatly overestimate τ_1 and overestimate τ_2 . More noticeably, the back estimate for the second disease, namely ϕ_2, τ_2 are rather good while the estimates for the first disease ϕ_1, τ_1, γ are quite off. This denote the model’s difficulties to identify properly what is going one for the first disease. This does not bide well for the chances of this model to produce conclusive results despite producing acceptable spatial patterns.

Besides, as in seen in Table 3.4, the weights of the structured effect at the mode for Oral-Lung, Oesophagus-Lung, and Larynx-Lung seem to suggest that the disease specific

	Oral	Oesophagus	Larynx	Lunglung
Shared $\gamma(1 - \phi_1)$	0.47	0.47	0.47	0.56
Specific ϕ_1	0.64	0.64	0.59	0.57

TABLE 3.4: Weights of the structured effects at the mode of the posterior marginal for the parameters. Oral oesophagus and larynx modeled jointly with lung cancer

component (alcohol consumption and potentially some others minor risk factors) has a stronger impact than the shared component that supposedly represent smoking which is expected to be the main risk cause. This does not corroborate what was said in Section 1.2. The last column represents the result from the posterior marginal when modeling lung cancer jointly with itself. It is then expected that smoking, the main risk factor, denoted by the shared component outweighs the disease specific component. It is not what is seen in Table 3.4. It is all the more suspicious since there is nothing in lung cancer that cannot be explained by lung cancer, thus it is expected that the disease specific component goes to 0. It thus seems that this model has serious shortcomings, and further work is required to successfully extend the use of penalized complexity priors to joint disease mapping. Besides, it seems unlikely that the shortcomings of this model are caused by the two major approximations made to derive the quantities of interest (normally distributed observations, method used to compute the posterior marginal) considering the discrepancies between what is expected and what is obtained.

3.2 Second Model with PC priors

This model is a modified version of the previous one. It was modified following suggestions from Patrick Brown from the University of Toronto who works with many researchers interested in disease mapping. There are essentially two main changes. First of all, the coefficient ϕ_1 for the shared disease component is now the same for both diseases. Besides the disease specific component is now given an unstructured effect. These changes are made in an effort to address the shortcomings of the previous model. Giving the same weight ϕ_1 to the shared component should make identifying the spatial pattern an easier task and better the shrinking property of the model, while giving the disease specific component an unstructured is meant to provide the first disease with unstructured error that is not shared with the second disease. The model is defined as follows:

$$\eta_1 = \mu_1 + \frac{1}{\sqrt{\tau_1}}(\sqrt{1-\gamma}(\sqrt{\phi_1}u_1^* + \sqrt{1-\phi_1}v_1) + \sqrt{\gamma}(\sqrt{\phi_2}u_2^* + \sqrt{1-\phi_2}v_2)), \quad (3.5)$$

$$\eta_2 = \mu_2 + \frac{1}{\sqrt{\tau_2}}(\sqrt{\phi_1}u_1^* + \sqrt{1-\phi_1}v_1), \quad (3.6)$$

where

- The structured effects u_1 and u_2 are scaled GMRF and a sum to zero constraint with mean 0 and precision Q^* .
- The intercept μ_1 and μ_2 follow a normal distribution with a large variance $\sigma^2 = 100$.
- The unstructured effects v_1 and v_2 follow a multivariate normal distribution $N(0, I)$.

3.2.1 Priors

When deriving the density of a prior, the other parameters are set arbitrarily.

Priors for τ_1 and τ_2

Since the precision matrix of the spatially structured effect is scaled, τ_1 and τ_2 represent the marginal deviation from the null space for each disease. The prior for τ_1 and τ_2 in Equation 3.5 are the same as for Equation 3.1. They follow a type 2 gumbel distribution with the same parameters as in Equation 3.5:

$$\pi(\tau_i) = \frac{\theta_i}{2} \tau_i^{-\frac{3}{2}} \exp(-\theta_i \tau_i^{-\frac{1}{2}}),$$

with $\theta_i = -\ln(\alpha_i)/U_i$.

3.2.2 Prior for ϕ_1

The parameter ϕ_1 represents the part of the variance of the shared component explained by the spatially structured effect. For this model the prior for ϕ_1 is different from before since ϕ_1 is present in both η_1 and η_2 . The two disease are then considered jointly in order to derive the prior for ϕ_1 .

Base case

The base model, when $\phi_1 = 0$, is when the shared component is only unstructured, in other words the shared risk factors do not have a spatial structure. It is still assumed that the precision priors and the weights can be derived independently. The base case is thus:

$$\begin{aligned} s_{1,0} &= \sqrt{1-\gamma}v_1 + \sqrt{\gamma}(\sqrt{\phi_2}u_2 + \sqrt{1-\phi_2}v_2), \\ s_{2,0} &= v_1. \end{aligned}$$

It can be rewritten as:

$$\begin{pmatrix} s_{1,0} \\ s_{2,0} \end{pmatrix} = \begin{bmatrix} \sqrt{1-\gamma} & \sqrt{\gamma} \\ 1 & 0 \end{bmatrix} \begin{pmatrix} v_1 \\ \sqrt{\phi_2}u_2 + \sqrt{1-\phi_2}v_2 \end{pmatrix}.$$

Let Q_{base} denote the precision of $\begin{pmatrix} v_1 \\ \sqrt{\phi_2}u_2 + \sqrt{1-\phi_2}v_2 \end{pmatrix}$:

$$Q_{base}^{-1} = \begin{bmatrix} I & 0 \\ 0 & \phi_2 Q^{-1} + (1-\phi_2)I \end{bmatrix}.$$

Let A denote $\begin{bmatrix} \sqrt{1-\gamma} & \sqrt{\gamma} \\ 1 & 0 \end{bmatrix}$ Then $\begin{pmatrix} s_{1,0} \\ s_{2,0} \end{pmatrix} \sim N(0, \Sigma_0^{-1} = A Q_{base} A^T)$.

Flexible case and derivation of the prior

Similarly we derive the distribution for the flexible model, eg. $\phi_1 \neq 0$. The flexible model is as follows:

$$\begin{aligned} s_{1,flex} &= (\sqrt{1-\gamma}(\sqrt{\phi_1}u_1 + \sqrt{1-\phi_1}v_1) + \sqrt{\gamma}(\sqrt{\phi_2}u_2 + \sqrt{1-\phi_2}v_2)), \\ s_{2,flex} &= (\sqrt{\phi_1}u_1 + \sqrt{1-\phi_1}v_1). \end{aligned}$$

Which can be rewritten as:

$$\begin{pmatrix} s_{1,flex} \\ s_{2,flex} \end{pmatrix} = \begin{bmatrix} \sqrt{1-\gamma} & \sqrt{\gamma} \\ 1 & 0 \end{bmatrix} \begin{pmatrix} \sqrt{\phi_1}u_1 + \sqrt{1-\phi_1}v_1 \\ \sqrt{\phi_2}u_2 + \sqrt{1-\phi_2}v_2 \end{pmatrix}.$$

Let Q_{flex} denote the precision of $\begin{pmatrix} \sqrt{\phi_1}u_1 + \sqrt{1-\phi_1}v_1 \\ \sqrt{\phi_2}u_2 + \sqrt{1-\phi_2}v_2 \end{pmatrix}$:

$$Q_{flex}^{-1} = \begin{bmatrix} \phi_1 Q^{-1} + (1-\phi_1)I & 0 \\ 0 & \phi_2 Q^{-1} + (1-\phi_2)I \end{bmatrix}.$$

Then:

$$\begin{pmatrix} s_{1,flex} \\ s_{2,flex} \end{pmatrix} \sim N(0, \Sigma_1^{-1} = A Q_{flex} A^T).$$

Then using:

$$KLD(N_n^{flex} || N_n^{base}) = \frac{1}{2} (tr(\Sigma_0^{-1} \Sigma_1) + (\mu_0 - \mu_1)^\top \Sigma_0^{-1} (\mu_0 - \mu_1) - n - \ln(\frac{|\Sigma_1|}{|\Sigma_0|})). \quad (3.7)$$

It follows that:

$$KLD(N_n^{flex} || N_n^{base}) = \frac{1}{2} (tr(\Sigma_0^{-1} \Sigma_1) - n - \ln(\frac{|\Sigma_1|}{|\Sigma_0|})),$$

$$KLD(N_n^{flex} || N_n^{base}) = \frac{1}{2} (tr(A Q_{base} Q_{flex}^{-1} A^{-1}) - n - \ln(\frac{|Q_{flex}^{-1}|}{|Q_{base}^{-1}|})).$$

Now using $Tr(AB) = Tr(BA)$, it follows that:

$$KLD(N_n^{flex} || N_n^{base}) = \frac{1}{2} (tr(Q_{flex}^{-1} Q_{base}) - n - \ln(\frac{|Q_{flex}^{-1}|}{|Q_{base}^{-1}|})).$$

Note then that the prior does not depend on γ . Then:

$$d(\phi_1) = \sqrt{tr(Q(\phi_1)_{flex}^{-1} Q_{base}) - n - \ln(\frac{|Q(\phi_1)_{flex}^{-1}|}{|Q_{base}^{-1}|})},$$

$$\pi(\phi_1) = \lambda_1 \frac{\exp(-\lambda_1 d(\phi_1))}{1 - \exp(-\lambda_1 d(1))} \left| \frac{\partial d(\phi_1)}{\partial \phi_1} \right|,$$

where $\frac{1 - \exp(-\lambda d(U_1))}{1 - \exp(-\lambda d(1))} = \alpha_1$ with $\alpha_1 = 0.5, U_1 = 2/3$. Note that A is not invertible when $\gamma = 0$, the base case is then:

$$s_{1,0} = v_1,$$

$$s_{2,0} = v_1.$$

And the flexible case is then:

$$\begin{aligned} s_{1,flex} &= (\sqrt{\phi_1}u_1 + \sqrt{1 - \phi_1}v_1), \\ s_{2,flex} &= (\sqrt{\phi_1}u_1 + \sqrt{1 - \phi_1}v_1). \end{aligned}$$

Thus the prior for ϕ_1 when $\gamma = 0$ is the same as the prior for ϕ in Section 2.3.3.

Prior for ϕ_2

The parameter ϕ_2 represents the part of the variance of the disease specific component explained by the spatially structured effect.

Base case: The base model, when $\phi_2 = 0$, is when the shared component is only unstructured, in other words the disease specific risk factors does not have any spatial structure. The base model is as follows:

$$s_{1,0} = \sqrt{1 - \gamma}(\sqrt{\phi_1}u_1 + \sqrt{1 - \phi_1}v_1) + \sqrt{\gamma}v_2.$$

Flexible model:

$$s_{1,flex} = \sqrt{1 - \gamma}(\sqrt{\phi_1}u_1 + \sqrt{1 - \phi_1}v_1) + \sqrt{\gamma}(\sqrt{\phi_2}u_1 + \sqrt{1 - \phi_2}v_2).$$

Thus:

$$d(\phi_2) = \sqrt{(tr(\Sigma_0^{-1}\Sigma_1) - n - \ln(\frac{|\Sigma_1|}{|\Sigma_0|}))},$$

where:

$$\begin{aligned} \Sigma_0 &= (1 - \gamma)\phi_1 Q^{-1} + (1 - \phi_1 + \gamma)I, \\ \Sigma_1 &= ((1 - \gamma)\phi_1 + \gamma\phi_2)Q^{-1} + ((1 - \gamma)(1 - \phi_1) + \gamma(1 - \phi_2))I, \end{aligned}$$

Hence:

$$\pi(\phi_2) = \frac{\lambda_2 \exp(-\lambda_2 d(\phi))}{1 - \exp(-\lambda_2 d(1))} \left| \frac{\partial d(\phi_2)}{\partial \phi_2} \right|,$$

where $\frac{1 - \exp(-\lambda d(U_2))}{1 - \exp(-\lambda d(1))} = \alpha_2$ with $\alpha_2 = 0.5, U_2 = 2/3$.

Prior for γ

The parameter γ is the part of the variance explained by the disease specific component. The base case $\gamma = 0$ is when there is not any disease specific component. Let $\gamma = 0$, then the base model is as follows:

$$s_{1,0} = (\sqrt{\phi_1}u_1 + \sqrt{1 - \phi_1}v_1),$$

and the flexible model:

$$s_{1,flex} = \sqrt{1 - \gamma}(\sqrt{\phi_1}u_1 + \sqrt{1 - \phi_1}v_1) + \sqrt{\gamma}(\sqrt{\phi_2}u_1 + \sqrt{1 - \phi_2}v_2).$$

Thus:

$$d(\gamma) = \sqrt{(tr(\Sigma_0^{-1}\Sigma_1) - n - \ln(\frac{|\Sigma_1|}{|\Sigma_0|}))},$$

where:

$$\Sigma_0 = \phi_1 Q^{-1} + (1 - \phi_1)I,$$

$$\Sigma_1 = ((1 - \gamma)\phi_1 + \gamma\phi_2)Q^{-1} + ((1 - \gamma)(1 - \phi_1) + \gamma(1 - \phi_2))I.$$

Hence:

$$\pi(\gamma) = \frac{\lambda_3 \exp(-\lambda_3 d(\gamma))}{1 - \exp(-\lambda_3 d(1))} \left| \frac{\partial d(\gamma)}{\partial \gamma} \right|,$$

where $\frac{1 - \exp(-\lambda d(U_3))}{1 - \exp(-\lambda_3 d(1))} = \alpha_3$ with $\alpha_3 = 0.5, U_3 = 2/3$.

3.2.3 Posterior marginal $\pi(u_1^*, v_1, u_2^*, v_2, \mu_1, \mu_2)$

The method to obtain the posterior marginal $\pi(u_1^*, v_1, u_2^*, v_2, \mu_1, \mu_2 | z_{1,:}, z_{2,:})$ has been done repeatedly throughout the project. Here are only displayed the results. The posterior marginal without the sum to zero constraints on u_1 and u_2 $\pi(u_1, v_1, u_2, v_2, \mu_1, \mu_2 | z_{1,:}, z_{2,:})$

is a GMRF such that $u_1, v_1, u_2, v_2, \mu_1, \mu_2 | z_{1,:}, z_{2,:} \sim N_c(b, Q_{GMRF})$ where:

$$Q_{GMRF}^* = \begin{bmatrix} Q_{u_1 u_1} & Q_{u_1 u_2} & Q_{u_1 v_1} & Q_{u_1 v_2} & Q_{u_1 \mu_1} & Q_{u_1 \mu_2} \\ & Q_{u_2 u_2} & Q_{u_2 v_1} & Q_{u_2 v_2} & Q_{u_2 \mu_1} & Q_{u_2 \mu_2} \\ & & Q_{v_1 v_1} & Q_{v_1 v_2} & Q_{v_1 \mu_1} & Q_{v_1 \mu_2} \\ & & & Q_{v_2 v_2} & Q_{v_2 \mu_1} & Q_{v_2 \mu_2} \\ & & & & Q_{\mu_1 \mu_1} & Q_{\mu_1 \mu_2} \\ & & & & & Q_{\mu_2 \mu_2} \end{bmatrix}.$$

Details about Q_{GMRF} are found in Appendix B.3 and :

$$b = \begin{pmatrix} \tau_{\epsilon_1} \frac{\sqrt{\phi_1(1-\gamma)}}{\sqrt{\tau_1}} y_1 + \tau_{\epsilon_2} \frac{\sqrt{\phi_1}}{\sqrt{\tau_2}} y_2 \\ \tau_{\epsilon_1} \frac{\sqrt{\gamma\phi_2} y_1}{\sqrt{\tau_1}} \\ \tau_{\epsilon_1} \frac{\sqrt{(1-\gamma)(1-\phi_1)} y_1}{\sqrt{\tau_1}} + \tau_{\epsilon_2} \sqrt{\frac{1-\phi_1}{\tau_2}} y_2 \\ \tau_{\epsilon_2} \frac{\sqrt{\gamma(1-\phi_2)} y_2}{\sqrt{\tau_2}} \\ \tau_{\epsilon_1} \Sigma y_{1,i} \\ \tau_{\epsilon_2} \Sigma y_{2,i} \end{pmatrix},$$

where $\mu = Q_{GMRF}^{*-1} b$. Thus the posterior marginal with the sum to zero constraints is $\pi(u_1, u_2, v_1, v_2, \mu_1, \mu_2 | \dots, 1^T u_1 = 0, 1^T u_2 = 0) \sim N(\mu^*, \Sigma^*)$ with:

$$\begin{aligned} \mu^* &= \mu - Q_{GMRF}^{*-1} A^T (A Q_{GMRF}^{*-1} A^T)^{-1} A \mu, \\ \Sigma^* &= Q_{GMRF}^{*-1} - Q_{GMRF}^{*-1} A^T (A Q_{GMRF}^{*-1} A^T)^{-1} A Q_{GMRF}^{*-1}, \end{aligned}$$

where:

$$A \begin{pmatrix} u_1 \\ u_2 \\ v_1 \\ v_2 \\ \mu_1 \\ \mu_2 \end{pmatrix} = 1^T u = 0.$$

With this it is easy to maximise $\pi(\phi_1, \phi_2, \tau_1, \tau_2 | z)$ and then compute the posterior marginal approximation for $\pi(u_1, v_1, u_2, v_2, \mu_1, \mu_2 | z_{1,:}, z_{2,:})$. The implementation is found in Appendix A.4.4.

3.2.4 Results

Results for oral oesphagus larynx and lung cancer

The code is found in Appendix A.4.5. Figures 3.8, 3.9, 3.10 show the shared components

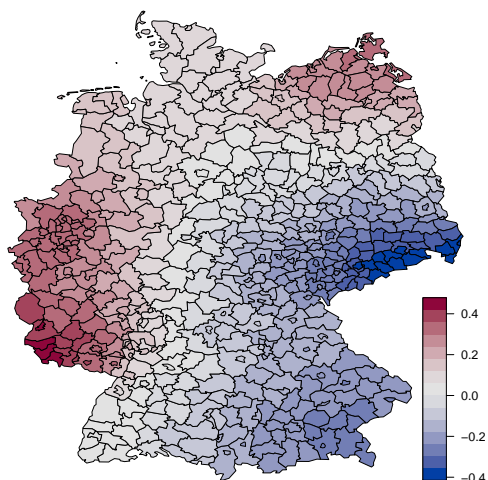


FIGURE 3.8: Shared spatial component for oral and lung cancers at the mode

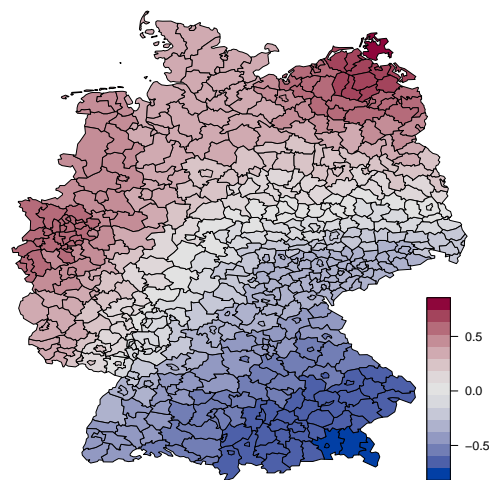


FIGURE 3.9: Shared spatial component for oesophagus and lung cancers at the mode

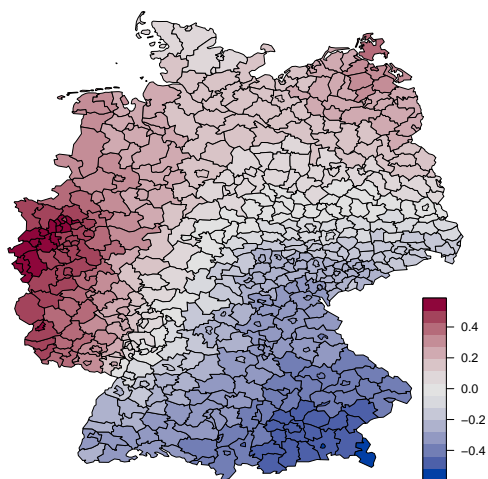


FIGURE 3.10: Shared spatial component for larynx and lung cancers at the mode

for Oral-Lung, Larynx - Lung, and Oesophagus-Lung. They are very similar to those obtained with the previous model which might mean that the spatial patterns for the disease specific and shared component are not too difficult to obtain accurately even if the model does not have the “shrinking” priors. There are two area of high risk in western and north eastern Germany for all three of them, which if the assumption

that the shared component represents mainly tobacco consumption is true, means that consumption of tobacco in these areas is high. Oral-Lung shows an area of increased risk in south western Germany probably accounting for a minor risk factor specific to lung and oral cancers since this area of increased risk is not found on the other plots. These plots are fairly similar to those obtain in [Held et al. \[2005\]](#) when jointly modeling these diseases which is an encouraging sign. Figures 3.11, 3.12, 3.13 show the disease

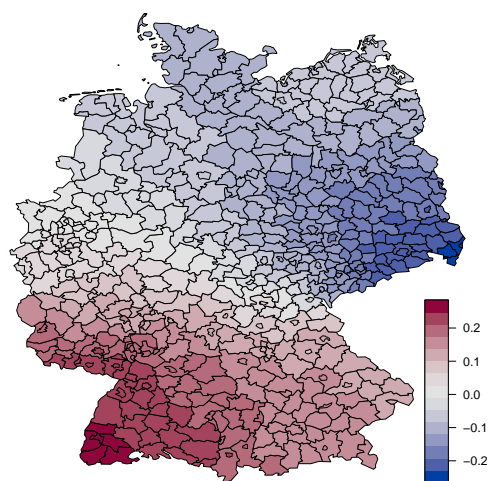


FIGURE 3.11: Disease specific component of oral cancer with lung cancer at the mode

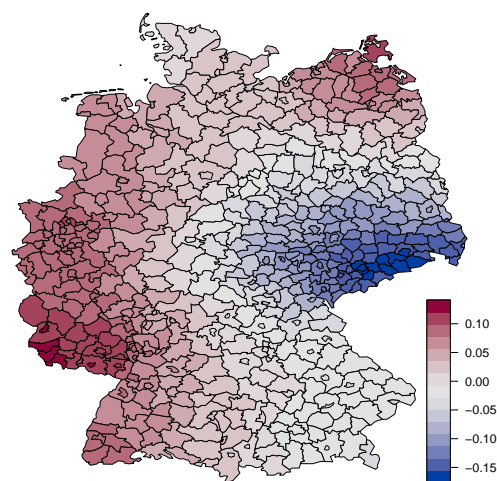


FIGURE 3.12: Disease specific component of oesophagus cancer with lung cancer at the mode

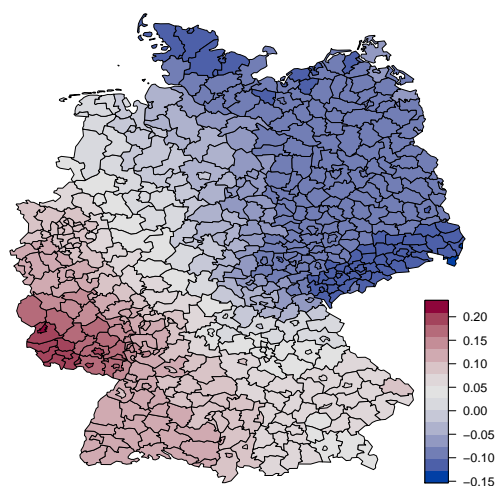


FIGURE 3.13: Disease specific component of larynx cancer with lung cancer at the mode

specific components for Oral-Lung, Larynx - Lung, and Oesophagus-Lung. They are very similar to those obtained with the previous model and are also quite comparable to those obtained in [Held et al. \[2005\]](#), and as such the analysis is rather similar to the one carried out for the previous model. Since the spatial patterns of the structured effect of

the disease specific components are rather different, it seems that alcohol is not the only risk factor at play, as was suggested in Section 1.2. Table 3.5 shows the weights for the shared and disease specific components for Oral-Lung, Larynx - Lung, and Oesophagus-Lung. It is seen that the shared component depicting smoking throughout Germany has a weight twice as large as the one for the disease specific component depicted alcohol consumption and others minor risk factors. It seems consistent with smoked tobacco being the main risk factor for all these cancer. The last column of Table 3.5 shows the weights for the shared and disease specific components when modeling jointly lung cancer with itself. It is somehow artificial but it is expected that the disease specific component goes to 0. While it does not quite exactly go to 0, the weight for the shared component is three times as large as the one for the disease specific component. If not perfect, this result is encouraging and far better than the result obtained for the previous model where both weights were about equal, see Table 3.4. Table 3.6 shows the result of the

	Oral	Oesophagus	Larynx	LungLung
Shared $\phi_1(1 - \gamma)$	0.56	0.63	0.58	0.74
Specific $\gamma\phi_2$	0.29	0.28	0.29	0.27

TABLE 3.5: Weights of the structured effect at the mode of the posterior marginal for the parameters. Oral oesophagus and larynx modeled jointly with lung cancer

same test carry out in Section 3.1.4.1 for Oral-Lung cancer for the current model. The explanation can be found in the aforementioned section. It is seen from the last column of Table 3.6 that ϕ_1 and ϕ_2 are slightly underestimated while τ_1, τ_2 are overestimated and γ only slightly. The estimates are nowhere as far off as the results from the previous model seen in Table 3.3. Although the precision terms are somewhat still overestimated, the model's approximations can probably account for the discrepancies.

	argmax	back estimate	ratio
$\sqrt{\phi_1}$	0.68	0.63	0.92
$\sqrt{\phi_2}$	0.51	0.48	0.93
$\sqrt{\tau_1}$	39.22	51.23	1.31
$\sqrt{\tau_2}$	42.61	51.13	1.20
$\sqrt{\gamma}$	0.57	0.58	1.02

TABLE 3.6: Column 1: Argmax of the posterior marginal Oral-Lung Cancer, Column 2: argmax of the posterior marginal Back estimation, Column 3: Ratio of the latter columns

Many reasons can explain that the results are not perfect. It can be because this model is somehow flawed and needs improvement, or because of the approximations made to allow for an easy derivation of the posterior marginals, or because the data is too scarce, or else because the mean should be considered as the quantity of interest as in Riebler et al. [2016]. However, the results obtained here is a huge leap in the right direction.

3.3 Tests with variable mesh size

It has been shown in the previous part that the results were convincing but not perfect. One possible shortcoming of the model is the lack of data. The following test only involve the second model described in Section 3.5, the first one having shown to be rather disappointing. The posterior distributions, due to the lack of data, rely on the prior distributions which possibly prevent the model from properly identifying the different components as in Table 3.6. It has been shown that despite these results being acceptable they are not quite perfect. To assess whether the lack of data might be responsible for it, replication of the data would be ideal, however it is not possible with the current implementation thus the following method to generate bigger mesh is proposed :

- Generate a mesh using the `cell2nb` R built in function
- Generate the precision matrix of the first order intrinsic GMRF for this mesh

```

1 a=cell2nb(p,p,torus = TRUE) # mesh
2 #Generate the precision matrix
3 Q_test= as.matrix(-nb2mat(a, style="B"))
4 diag(Q_test) = -rowSums(Q_test) # make sure
5 # that the diagonal term is minus the sum of the others
6 Q_test = INLA:::inla.scale.model.bym(Q_test) # scaling

```

- Generate data
- See how the estimates improve as the mesh size increases

If the estimates converge towards “the data” as the mesh size increases, then it is likely that the lack of data is responsible for at least part of the explanation as to why the estimates in Table 3.6 were a little off.

3.3.1 Test with a constant risk surface

The data is generated assuming:

$$\begin{aligned}\eta_{1,i} &= K_1, \\ \eta_{2,i} &= K_2.\end{aligned}$$

Then the posterior marginal for $\text{logit}(\phi_1)$, $\text{logit}(\phi_2)$, $\log(\tau_1)$, $\log(\tau_2)$, $\log(\gamma)$ is maximized using this data. First remember that the optimization is done on the internal scale and as $\log(\tau_1)$ and $\log(\tau_2)$ increase, so do τ_1 and τ_2 . It is then expected that τ_1 and τ_2 go

mesh size	100×100	$21^2 \times 21^2$	$29^2 \times 29^2$	$33^2 \times 33^2$
τ_1	560.64	2033.30	3622.99	4631.18
τ_2	563.90	2033.84	3643.21	4659.50

TABLE 3.7: Estimate of mode of the posterior marginal for different mesh sizes

to infinity as the mesh size increases. As is seen in Table 3.7, τ_1 and τ_2 both increase substantially as the mesh size increases which would tend to prove that as the amount of data increase the model gets more and more precise. The code can be found in Appendix A.4.6.

3.3.2 Test with no disease specific component

The data is generated assuming :

$$\eta_1 = \frac{1}{\sqrt{\tau_1}}((\sqrt{\phi_1}u_1^* + \sqrt{1 - \phi_1}v_1),$$

$$\eta_2 = \frac{1}{\sqrt{\tau_2}}(\sqrt{\phi_1}u_1^* + \sqrt{1 - \phi_1}v_1).$$

Then the posterior marginal for $\text{logit}(\phi_1)$, $\text{logit}(\phi_2)$, $\log(\tau_1)$, $\log(\tau_2)$, $\log(\gamma)$ is maximized using this data. So it is expected that $\text{logit}(\gamma)$ goes to $-\infty$ or on the original scale γ goes to 0. As is seen in Table 3.8, γ decreases as the mesh size increases, however it seems

mesh size	100×100	$21^2 \times 21^2$	$29^2 \times 29^2$	$33^2 \times 33^2$
ϕ_1	0.46	0.43	0.49	0.46
γ	0.50	0.37	0.31	0.30

TABLE 3.8: Estimate of mode of the posterior marginal for different mesh sizes

to plateau which would tend to show that despite a substantial decrease, the mode does not go to zero as the mesh size increase. On the other hand, ϕ_1 is rather close to its “true value” 0.5 chosen for the implementation that can be found in Appendix A.4.6. The results from this test are altogether encouraging for if the model does not exactly identify the data, it does go in the right direction.

3.3.3 Test Overview

The two previous tests showed the model ability to shrink at least to a certain extent towards simpler models. Even though the results are not perfect, it is at least encouraging for there are still many reasons that could explain why the results are what they are. First of all, the test was run for only one sample and would require to be run over substantially more samples to get a reliable estimate. Furthermore, the posterior

marginal is optimized on the internal scale rather than on the original scale which might be misleading. Besides, the approximation made to derive the posterior marginals only produce their modes. In [Riebler et al. \[2016\]](#), the shrinking properties of the PC priors are obtained using the mean of the posterior marginal. Unfortunately, the method used falls short to compute the mean of a posterior marginal. It is one of the thing that would have to be further explored to really assess whether the PC priors have, when jointly modeling disease, the same properties as when modeling just one disease.

Chapter 4

Further developments

This chapter is about further extending the project. It features possible extensions and improvement of the models and methods. There are, as a matter of fact, many aspects of this project that can be refined improved and extended. The nature of the project, itself, explains why so many things are still to be done. Since the work done on PC priors when jointly modeling disease is rather new, a lot of assumptions were made at the beginning, and the results being rather encouraging but not quite conclusive, the topic has to be further explored. Eventually, the model's results should be obtain using the theory behind R-INLA to compare the results, and to have a method that has proven to be efficient.

4.1 Extension to Poisson

Throughout the project, it was assumed that the square root of the SMR followed a normal distribution. The main motivation for switching to a poisson distribution is to have a better basis for comparison. Indeed, in [Riebler et al. \[2016\]](#), the results are obtained using a Poisson distribution on the diseases counts. It is assumed that :

$$y_{1,i} \sim \text{Poisson}(e_{1,i}\lambda_{1,i}),$$

$$y_{2,i} \sim \text{Poisson}(e_{2,i}\lambda_{2,i}),$$

where $\log(\lambda_{j,i}) = \eta_{j,i}$ and

$$\eta_1 = \mu_1 + \frac{1}{\sqrt{\tau_1}}(\sqrt{1-\gamma}(\sqrt{\phi_1}u_1^* + \sqrt{1-\phi_1}v_1) + \sqrt{\gamma}(\sqrt{\phi_2}u_2^* + \sqrt{1-\phi_2}v_2)), \quad (4.1)$$

$$\eta_2 = \mu_2 + \frac{1}{\sqrt{\tau_2}}(\sqrt{\phi_1}u_1^* + \sqrt{1-\phi_1}v_1), \quad (4.2)$$

where $u_1^*, u_2^*, v_1, v_2, \mu_1, \mu_2$ are the same as in the model detailed in Section 3.2. The posterior marginal of $\phi_1, \phi_2, \gamma, \tau_1, \tau_2$ is then :

$$\begin{aligned} \pi(\phi_1, \phi_2, \gamma, \tau_1, \tau_2 | y_{1,i}, y_{2,i}) &= \frac{\pi(u_1, u_2, v_1, v_2, \mu_1, \mu_2, \phi_1, \phi_2, \gamma, \tau_1, \tau_2 | y_{1,i}, y_{2,i})}{\pi(u_1, u_2, v_1, v_2, \mu_1, \mu_2 | y_{1,i}, y_{2,i}, \phi_1, \phi_2, \gamma, \tau_1, \tau_2)}, \\ \pi(\phi_1, \phi_2, \gamma, \tau_1, \tau_2 | y_{1,i}, y_{2,i}) &\propto \pi(\phi_1, \phi_2, \gamma, \tau_1, \tau_2) \\ &\quad \times \frac{\pi(u_1, u_2, v_1, v_2, \mu_1, \mu_2 | \phi_1, \phi_2, \gamma, \tau_1, \tau_2) \pi(y_{1,i}, y_{2,i} | \dots)}{\pi(u_1, u_2, v_1, v_2, \mu_1, \mu_2 | y_{1,i}, y_{2,i}, \phi_1, \phi_2, \gamma, \tau_1, \tau_2)}, \end{aligned}$$

where

$$\pi(y_{1,i}, y_{2,i} | \dots) \propto \exp \sum y_{1,i} \eta_{1,i} - \log(e_{1,i} \eta_{1,i}) \exp \sum y_{2,i} \eta_{2,i} - \log(e_{2,i} \eta_{2,i}),$$

and

$$\begin{aligned} \pi(u_1, u_2, v_1, v_2, \mu_1, \mu_2 | y_{1,i}, y_{2,i}, \phi_1, \phi_2, \gamma, \tau_1, \tau_2) &\propto \pi(y_{1,i}, y_{2,i} | \dots) \\ &\quad \times \pi(u_1, u_2, v_1, v_2, \mu_1, \mu_2 | \phi_1, \phi_2, \gamma, \tau_1, \tau_2), \\ &\propto \exp \sum y_{1,i} \eta_{1,i} - \log(e_{1,i} \eta_{1,i}) \\ &\quad \times \exp \sum y_{2,i} \eta_{2,i} - \log(e_{2,i} \eta_{2,i}) \\ &\quad \times \exp -\frac{1}{2}(u_1^T Q^{-1} u_1 + u_2^T Q^{-1} u_2) \\ &\quad \times \exp -\frac{1}{2}(v_1^T I v_1 + v_2^T I v_2 + \frac{\mu_1^2}{\sigma^2} + \frac{\mu_2^2}{\sigma^2}). \end{aligned}$$

It is obviously quite cumbersome to manipulate. Mostly because the full conditional of $u_1, u_2, v_1, v_2, \mu_1, \mu_2$ is not a GMRF as it was in the normal case. To remedy to that, Rue et al. [2009] proposes to use a gaussian approximation of the full conditional $\pi_G(u_1, u_2, v_1, v_2, \mu_1, \mu_2 | y_{1,i}, y_{2,i}, \phi_1, \phi_2, \gamma, \tau_1, \tau_2)$. Once this approximation is made the computations made in this project can basically be carried out for the poisson case the same way they have been in the project.

4.2 Posterior marginal of $u_1, u_2, v_1, v_2, \mu_1, \mu_2$

The following Equation 2.11

$$\pi(u, v | z) \approx \pi(u, v | z, \theta_{mode})$$

is used throughout the project as a simplification of Equation 2.10 given below

$$\pi(u, v | z) \approx \sum_k \pi(u, v | z, \theta_k) \pi(\theta_k | z) \Delta_k.$$

Although using Equation 2.10 should not change the result dramatically, because the approximation in Equation 2.11 is, as said before, reasonable, it is still necessary to implement it properly to be able to carry out a precise analysis of the results and have results that are comparable to Riebler et al. [2016].

4.3 Mean and original scale

4.3.1 Mean rather than mode

Throughout the project, the mode of the posterior marginals of $u_1, u_2, v_1, v_2, \mu_1, \mu_2$ and $\phi_1, \phi_2, \gamma, \tau_1, \tau_2$ were computed and used to analyze model performance. Even though the mode of the posterior marginal of $\phi_1, \phi_2, \gamma, \tau_1, \tau_2$ was needed to compute the density of the posterior marginal of $u_1, u_2, v_1, v_2, \mu_1, \mu_2$, the main reason why the mode was systematically computed is because it is the easiest quantity to compute. Easier than the mean. However, now that the second model produced encouraging results, having access to others quantities such as the mean is necessary. The mean, especially of the posterior marginals, would be a useful quantity to have access to for it is with the mean of the posterior marginals that the *shrinking to a simpler model* properties of the PC priors is shown in Riebler et al. [2016]. In this project, a similar analysis is done but using the mode of the posterior marginal. The obtained results would probably be more reliable if it had been done with the mean.

4.3.2 Original scale vs internal scale

Using the internal scale (log and logit) helped turn a constrained optimization problem into an unconstrained optimization problem. However parameter interpretation makes more sense on the original scale and if the "shrinking" capacity of the pc priors is to be seriously tested for joint disease mapping, having access to the mean (or the mode) on the original scale, as in Riebler et al. [2016] would be of great help.

4.4 Tobacco- alcohol synergy

There is a synergy between alcohol consumption and smoking that the models fails to account for. It is in nature non linear and the easiest way to account for it would be to use covariates that quantify the whether the patients drank or smoke or both. The problem is, more often than not, this kind of data is not available.

4.5 Rank deficiency of the precision matrix

The rank deficiency of the intrinsic GMRF precision matrix has been dealt with either by taking the generalized inverse of the precision matrix when the inverse matrix was required or by increasing slightly the diagonal. It also led to adjusting the KLD for the parameters. If the priors were to shrink towards simpler model, looking into a proper way to use this degenerate matrix would probably be necessary as problems might occur when studying the limiting cases.

4.6 Extension to more than two diseases

If after all that is suggested before gives convincing results then extending to more than two diseases could be the next step. It has been done in [Held et al. \[2005\]](#), and it can be used to compare results.

Chapter 5

Discussion

The goal of the project was to try and assess whether PC priors could be successfully used when jointly modeling disease. [Held et al. \[2005\]](#) showed that jointly modeling disease could be a good idea, and [Simpson et al. \[2015\]](#) and then [Riebler et al. \[2016\]](#) showed that PC priors could be interesting disease mapping tools.

This project did not produce results that would allow for a conclusive answer, however, pieces of evidence were found. Under simplifying assumption spatial patterns similar to those found in [Held et al. \[2005\]](#) were found together with a model aptitude to shrink towards simpler models similar albeit less convincing than the one found in [Riebler et al. \[2016\]](#).

The findings of this project seems to suggest that the PC priors could have very similar properties when jointly modeling disease as when modeling only one disease. The findings do not conflict with [Riebler et al. \[2016\]](#), rather they are not as strong and conclusive. At this stage it is hard to assess if the result are so because of the simplifying assumptions that were made or because some properties are lost when extending to two diseases. The results however showed, even for the most accomplished model that the precision priors tend to overestimate the value of the parameters. It would be however rather harsh not to give the results the benefit of the doubt because the results are overall going in the right direction. Besides, at this stage it is hard to assess whether the overestimation is caused by a flawed model or by the simplifications that were made.

The findings of this project seem encouraging enough to explore what would happen with fewer simplifications. Using R-INLA methodology would give a better basis for comparison and hopefully provide conclusive results. The amount of work required to do so is unfortunately too important to be done as a part of this project. However, it

would be really exciting to see whether further development would lead to results as exciting as expected.

Appendix A

Code excerpts

A.1 Model 1

A.1.1 Implementation

This code computes the analytical mode of the posterior marginal of u .

```
1 library(INLA)
2 data(Germany)
3 n = 544
4 to=exp(2)
5 s2=1/4
6 ID = diag.spam(n)
7 # SMR for oral cancer
8 z=sqrt(or$V3)
9 inv=solve(s2*(to*R+ID/s2))
10 u=inv%*%z
```

A.2 Model 2

A.2.1 Implementation

This code computes the analytical mode of the posterior marginal of u, v , first computing u , and then obtaining v by substituting u with its value at the mode in the system.

```
1 k= 0.5
2 n = 544
3 to=exp(2)
```

```

4 s2=1/4
5 lhs.m=ID/s2+to*R-ID/(s2*s2*(k+1/s2))
6 rhs.v=(1/s2-1/(s2*s2*(k+1/s2)))*z
7 inv.lhs=solve(lhs.m)
8 # solution
9 u1=inv.lhs%*%rhs.v
10 v1=(z-u1)/(1+s2*k)

```

A.3 3rd Model

A.3.1 Maximizing the posterior marginal for κ and τ

This code computes the arg max of the posterior marginal for κ and τ for the four different cancers.

```

1 alpha.k=3
2 alpha.to=3
3 beta.k=1
4 beta.to=1
5 s2=1/4
6 # Data set for Oral cancer
7 orl = read.table("germany.0.data")
8 # Data set for oesophagus cancer
9 oesoph = read.table("germany.1.data")
10 #Data set for larynx cancer
11 lar= read.table("germany.2.data")
12 # Data set for Lung cancer
13 lungcancer = read.table("germany.3.data")
14 # SMR for the for types of cancer
15 # SMR oral
16 z = sqrt(or$V3)
17 # SMR oesophagus
18 b =sqrt(oesoph$V3)
19 #SMR Larynx
20 c = sqrt(lar$V3)
21 # SMR Lung
22 d = sqrt(lungcancer$V3)
23 # Posterior Marginal for kappa and tau
24 try1=function(theta , z ,R, s2 , alpha.to , alpha.k , beta.to , beta.k) {
25   ID = diag.spam(544)
26   Q1=rBind(cBind(theta [1] *R+ID/s2 , ID/s2) , cBind((ID/s2) , theta [2] *ID+
    ID/s2))

```

```

27  Q2=as.matrix(rBind(cBind(theta[1]*R+ID/s2, ID/s2),
28  cBind((ID/s2), theta[2]*ID+ID/s2)))
29  logdet=as.numeric(determinant(Q2)$modulus)
30  b=rbind(cbind(z), cbind(z))/s2
31  mu=solve(Q1, b)
32  fin=(543/2+alpha.to-1)*log(theta[1])+(544/2+alpha.k-1)
33  *log(theta[2])-beta.to*theta[1]
34  -beta.k*theta[2]-logdet/2 + t(mu)%*%Q1%*%mu/2
35  }
36  # Argmax for Oral
37  prior.max12=optim(par=c(18.13, 28.38), fn=try1, z=z, R=R,
38  s2=s2, alpha.to = alpha.to, alpha.k = alpha.k,
39  beta.to = beta.to, beta.k = beta.k, control = list(fnscale=-1),
40  method="BFGS")
41  # Argmax for Oesophagus
42  prior.max13=optim(par=c(18.13, 28.38), fn=try1, z=b, R=R,
43  s2=s2, alpha.to = alpha.to, alpha.k = alpha.k,
44  beta.to = beta.to, beta.k = beta.k, control = list(fnscale=-1),
45  method="BFGS")
46  # Argmax for Larynx
47  prior.max14=optim(par=c(18.13, 28.38), fn=try1, z=b, R=R,
48  s2=s2, alpha.to = alpha.to, alpha.k = alpha.k,
49  beta.to = beta.to, beta.k = beta.k, control = list(fnscale=-1),
50  method="BFGS")
51  # Argmax for Lung
52  prior.max15=optim(par=c(18.13, 28.38), fn=try1, z=d, R=R,
53  s2=s2, alpha.to = alpha.to, alpha.k = alpha.k,
54  beta.to = beta.to, beta.k = beta.k,
55  control = list(fnscale=-1),
56  method="BFGS")

```

A.3.2 Computing the mode of the posterior marginal for u and v

This code computes the posterior marginal of u, v

```

1  # function to obtain the mode of the posterior marginal
2  get_max1 = function(k1, to1, R=R, s2=s2, z) {
3    n = 544
4    s2=1/4
5    ID = diag.spam(544)
6    lhs.ml=ID/s2+to1*R-ID/(s2*s2*(k1+1/s2))
7    rhs.v1=(1/s2-1/(s2*s2*(k1+1/s2)))*z
8    inv.lhs1=solve(lhs.ml)

```

```

9 # solution
10   u12=inv.lhs1%*%rhs.v1
11   v11=(z-u12)/(1+s2*k1)
12   result = data.frame(u12,v11)
13 }
14 # Mode of the posterior marginal for oral cancer
15 oral = get_max1(k1=prior.max12$par[2],tol= prior.max12$par[1],
16   R=R,s2=s2,z)
17 # Mode of the posterior marginal for oesophagus cancer
18 oesophagus = get_max1(k1=prior.max13$par[2],tol= prior.max13$par[1],
19   R=R,s2=s2,b)
20 # Mode of the posterior marginal for larynx cancer
21 larynx = get_max1(k1=prior.max14$par[2],tol= prior.max14$par[1],
22   R=R,s2=s2,c)
23 # Mode of the posterior marginal for lung cancer
24 lung = get_max1(k1=prior.max15$par[2],tol= prior.max15$par[1],
25   R=R,s2=s2,d)

```

A.3.3 Implementation

This code computes the mode of the posterior marginal of $\log(\phi)$, $\text{logit}(\tau)$ and the posterior marginal for u, v, μ

```

1 logdens_pcprior2 = function(par, prec.eps = 4,prec.par,
2   phi.par, y, Q,prec.mean=0.01){
3   require(mvtnorm)
4   # change of variable to have an unconstrained parameter space
5   prec = exp(par[1])
6   phi = exp(par[2])/(1+exp(par[2]))
7   n = dim(Q)[1]
8   # Compute lambda
9   lambda = -log(prec.par[2])/prec.par[1]
10   ld = 0
11   # log(prec) is added to the distribution for tau to satisfy
12   # the change of variable formula
13   ld = ld - (1/2)*par[1] - lambda*prec^(-1/2)
14   # log( exp(par[2]) / (1 + exp(par[2]))^2 ) is added to
15   #the distribution for phi to satisfy the change of #variable formula
16   ld = ld + INLA:::inla.pc.bym.phi(Q=Q, u= phi.par[1], alpha = phi
17   .par[2])(phi)
18   + log( exp(par[2]) / (1 + exp(par[2]))^2 )
19   # that we can keep
20   Ident = diag(n)

```

```

20   one = matrix(1,544,1)
21   QQ = (cbind(rbind(Ident + prec.eps * (1-phi)/prec * Ident ,
22               +prec.eps * sqrt((1-phi)*phi)/prec * Ident ,
23               prec.eps*sqrt((1-phi)/prec)*t(one)) ,
24               rbind(prec.eps*sqrt((1-phi)*phi)/prec*Ident ,
25               Q + prec.eps * phi/prec * Ident ,
26               prec.eps*sqrt(phi/prec)*t(one)) ,
27               rbind(prec.eps*sqrt((1-phi)/prec)*one ,
28               prec.eps*sqrt(phi/prec)*one ,
29               prec.eps*n+prec.mean*1)))
30   b = (c(prec.eps * sqrt((1-phi)/prec) * y ,
31         prec.eps * sqrt(phi/prec) * y,prec.eps*sum(y)))
32   QQ = INLA:::inla.as.sparse(QQ)
33   mu = c(inla.qsolve(QQ, matrix(b, 2*n+1, 1)))
34   ld = ld - c(inla.qsample(Q=QQ, mu=mu, sample = rep(0, 2*n+1),
35                   logdens=TRUE, constr = list(A = matrix(c(rep(0, n),
36                   rep(1,n),0), 1, 2*n+1), e = 0))$logdens)
37   return (ld)
38 }
39
40 g = system.file("demodata/germany.graph", package="INLA")
41 g=inla.read.graph(g)
42 Q=as.matrix(inla.graph2matrix(g))
43 Q = -Q
44 diag(Q) = g$nnbs
45 Q = INLA:::inla.scale.model.bym(Q)
46 diag(Q) = diag(Q) + 1e-8
47 data(Germany)
48 y = sqrt(Germany$Y/Germany$E)
49 prec.par = c(0.2/0.31, 0.01)
50 phi.par = c(0.5, 2/3)
51 prec.mean = 0.01
52
53 bss4= (optim(par=c(0.1, 0),fn=logdens_pcprior2 ,
54             prec.par = prec.par ,
55             phi.par = phi.par ,
56             y = y ,
57             Q = Q,
58             control = list(fnscale=-1, reltol = 1e-16),
59             method = "L-BFGS-B",lower = c(1e-3,-10)))
60
61 # posterior marginal u,v,mu
62 par1=bss4$par
63 prec = exp(par1[1])

```

```

64 phi = exp(par1[2])/(1+exp(par1[2]))
65 n = dim(Q)[1]
66 Ident = diag(n)
67 one=matrix(1,544,1)
68 QQ = (cbind(rbind(Ident + prec.eps * (1-phi)/prec * Ident ,
69                 +prec.eps * sqrt((1-phi)*phi)/prec * Ident ,
70                 prec.eps*sqrt((1-phi)/prec)*t(one)) ,
71         rbind(prec.eps*sqrt((1-phi)*phi)/prec*Ident ,
72         Q + prec.eps * phi/prec * Ident ,
73         prec.eps*sqrt(phi/prec)*t(one)) ,
74         rbind(prec.eps*sqrt((1-phi)/prec)*one ,
75         prec.eps*sqrt(phi/prec)*one ,
76         prec.eps*n+prec.mean*1)))
77 b = (c(prec.eps * sqrt((1-phi)/prec) * y ,
78        prec.eps * sqrt(phi/prec) * y, prec.eps*sum(y)))
79 mu=solve(QQ,b)
80 A = matrix(c(rep(0, n), rep(1,n), 0), 1, 2*n+1)
81 QQ.inv=solve(QQ)
82 c=solve((A%*%QQ.inv%*%t(A)))
83 mean= mu-QQ.inv%*%t(A)%*%c%*%(A%*%mu)
84

```

A.4 Scaling the priors

These are the settings used for the next two subsection of this appendix.

```

1 g = system.file("demodata/germany.graph", package="INLA")
2 g=inla.read.graph(g)
3 Q=as.matrix(inla.graph2matrix(g))
4 Q = -Q
5 diag(Q) = g$nnbs
6 Q = INLA:::inla.scale.model.bym(Q)
7 diag(Q) = diag(Q) + 1e-8
8 prec.mean.1 = 0.01
9 prec.mean.2 = 0.01
10 prec.eps.1 = 4
11 prec.eps.2 = 4
12 phi.par = matrix(rep(0), 3, 2)
13 phi.par[1,] = c(0.5, 2/3)# .5 for all
14 phi.par[2,] = c(0.5, 2/3)
15 phi.par[3,] = c(0.5, 2/3)
16 prec.par = matrix(rep(0), 2, 2)

```

```

17 prec.par[1,] = c(0.2/0.31, 0.01)
18 prec.par[2,] = c(0.2/0.31,0.01)
19 inv = ginv(as.matrix(Q))
20 eig_1 = eigen(Q)$value[1:543]
21 eig_1 = 1/eig_1

```

A.4.1 prior for ϕ_1 and midpoint formula

```

1 #Function that compute d(phi_1)
2 dist_phi1 = function (phi1,gamma,Ident,Q,n,inv = inv, eig_1 = eig_1){
3   sigma_0 = (1-gamma)*Ident+gamma*inv
4   sigma_1 = (1-phi1)*(1-gamma)* Ident + ((1-phi1)*gamma+phi1)*inv
5   tr_M = sum(diag(ginv(sigma_0)%*%sigma_1))
6   ln_S_0 = sum(log(1-gamma+gamma*eig_1)) # ok
7   ln_S_1 = sum(log( (1-phi1)*(1-gamma)+ ((1-phi1)*gamma+phi1)*eig_1
8   )
9   res = sqrt( (tr_M - (n) - ln_S_1 + ln_S_0)+ 1e-2)
10 }
11 # Function that compute the prior for phi_1
12 kld_dist = function (phi1,gamma,Ident,Q,n,lambda_3,
13   inv = inv, eig_1 = eig_1){
14   result = 0
15   result = result + log(lambda_3)
16   result = result - lambda_3*dist_phi1(phi1,gamma,Ident,Q,n,
17   inv = inv, eig_1 = eig_1)
18   delta =1e-4
19   phi_d = phi1 + delta
20   d = log(abs(dist_phi1(phi_d,gamma,Ident,Q,n,
21   inv = inv, eig_1 = eig_1)-
22   dist_phi1(phi1,gamma,Ident,Q,
23   n,inv = inv, eig_1 = eig_1))/delta)
24   result = result +d
25   result = result - log (1-
26   exp(-lambda_3*dist_phi1(phi1=1,gamma= gamma,
27   Ident=Ident,Q=Q,n=n,
28   inv = inv, eig_1 = eig_1)))
29 return(result)
30 }

```

Integral for the scaled and unscaled prior using the mid point formula :

```

1 # Compute the integral of the scale and unscaled prior
2 a=seq(0,1,0.05)

```

```

3 res = rep(0,length(a))
4 prod1 = rep(0,length(a))
5 # Lambda for the chosen value of gamma
6 lamb = - log (1 -phi.par[3,2])/dist_phi1(phi1 = phi.par[3,1],
7 gamma=0.5,Ident = Ident ,Q = Q,n=n,inv = inv , eig_1 = eig_1)
8 for (i in 1:length(a)){
9 # unscaled prior
10   res[i] = kld_dist_uns(phi1 = a[i],gamma = 0.5,Ident = Ident ,Q = Q,
11   n = n,lambda_3 = lamb,inv = inv , eig_1 = eig_1)
12 # scaled prior
13   prod1[i] = kld_dist(phi1 = a[i],gamma = 0.5,Ident = Ident ,Q = Q,
14   n = n,lambda_3 = lamb,inv = inv , eig_1 = eig_1)
15 }
16 sum = 0
17 sum_scale = 0
18 for (i in 1 :(length(a)-1))
19 {
20 # mid point method for the unscaled prior
21   sum = sum + 0.05 * (exp(res[i])+exp(res[i+1]))/2
22 # same method
23   sum_scale = sum_scale + 0.05 *(exp(prod1[i])+exp(prod1[i+1]))/2
24
25 }
26 sum = round(sum,digits =3)
27 sum_scale = round(sum_scale ,digits = 3)

```

A.4.2 prior for γ and midpoint formula

This code computes the prior for gamma both scaled and unscaled.

```

1 # Function that computes d(gamma)
2 dist_gamma = function (gamma,phi1 ,Ident ,Q,n,
3 inv = inv , eig_1 = eig_1){
4   sigma_0 = (1-phi1)*Ident+phi1*inv
5   sigma_1 = (1-phi1)*(1-gamma)* Ident + ((1-phi1)*gamma+phi1)*inv
6   tr_M = sum(diag(ginv(sigma_0)%*%sigma_1))
7   ln_S_0 = sum(log(1-phi1+phi1*eig_1)) # done
8   ln_S_1 = sum(log( (1-phi1)*(1-gamma)+ ((1-phi1)*gamma+phi1)*eig_1
9   )
10   res = sqrt( (tr_M - (n) - ln_S_1 + ln_S_0)+1e-2)
11 # 1-e2 is a corrective term for when gamma is small, numerical error
    in
12 # ln_S_0 and ln_S_1 produce very small negative number.

```

```

12 }
13 # Function that computes the prior for gamma
14 kld_dist_gamma_uns = function (gamma, phi1, Ident, Q, n,
15 lambda_5, inv = inv, eig_1 = eig_1){
16     result = 0
17     result = result + log (lambda_5)
18     result = result - lambda_5*dist_gamma(gamma, phi1,
19 Ident, Q, n, inv = inv, eig_1 = eig_1)
20     delta = 1e-4
21     gamma_d = gamma + delta
22     result = result + log (abs(dist_gamma(gamma = gamma_d, phi1, Ident,
23 Q, n, inv = inv, eig_1 = eig_1)-dist_gamma(gamma = gamma, phi1,
24 Ident, Q, n, inv = inv, eig_1 = eig_1))/delta)
25     result = result
26     print(c(result, 'gamma'))
27 return(result)
28 }
29 kld_dist_gamma = function (gamma, phi1, Ident, Q,
30 n, lambda_5, inv = inv, eig_1 = eig_1){
31     result = 0
32     result = result + log (lambda_5)
33     result = result - lambda_5*dist_gamma(gamma, phi1, Ident,
34 Q, n, inv = inv, eig_1 = eig_1)
35     delta = 1e-4
36     gamma_d = gamma + delta
37     result = result + log (abs(dist_gamma(gamma = gamma_d, phi1, Ident,
38 Q, n, inv = inv, eig_1 = eig_1)-dist_gamma(gamma = gamma, phi1,
39 Ident, Q, n, inv = inv, eig_1 = eig_1))/delta)
40     result = result - log(1 - exp(-lambda_5*dist_gamma(gamma=1,
41 phi1=phi1, Ident=Ident, Q=Q, n=n, inv = inv, eig_1 = eig_1)))
42 return(result)
43 }

```

A.4.3 Posterior marginal for $u_1, v_1, u_2, v, \mu_1, \mu_2$

This code computes the mode of the posterior marginal of $\text{logit}(\phi_1)$, $\text{logit}(\phi_2)$, $\log(\tau_1)$, $\log(\tau_2)$, $\text{logit}(\gamma)$ and the mode of the posterior marginal of $u_1, v_1, u_2, v, \mu_1, \mu_2$. Note that the prior for ϕ_1 and γ 's code is in [A.4.2](#).

```

1 # Neighbor matrix
2 g = system.file("demodata/germany.graph", package="INLA")
3 g=inla.read.graph(g)
4 Q=as.matrix(inla.graph2matrix(g))

```

```

5 Q = -Q
6 diag(Q) = g$nnbs
7 Q = INLA:::inla.scale.model.bym(Q)
8 diag(Q) = diag(Q) + 1e-8
9 # Parameters
10 prec.mean.1 = 0.01
11 prec.mean.2 = 0.01
12 prec.eps.1 = 4
13 prec.eps.2 = 4
14 phi.par = matrix(rep(0),3,2)
15 phi.par[1,] = c(0.5, 2/3) # .5 for all
16 phi.par[2,] = c(0.5, 2/3)
17 phi.par[3,] = c(0.5, 2/3)
18 prec.par = matrix(rep(0),2,2)
19 prec.par[1,] = c(0.2/0.31, 0.01)
20 prec.par[2,] = c(0.2/0.31,0.01)
21 inv = ginv(as.matrix(Q))
22 eig_1 = eigen(Q)$value[1:543]
23 eig_1 = 1/eig_1
24 # Function to maximise to obtain the posterior marginal for the
   parameters
25 logdens_joint_v2 = function(par, prec.eps.1 = 4,prec.eps.2 = 4, y1,
   y2,
26 Q, prec.mean.2 = 0.01,prec.mean.1 = 0.01,
27 prec.par, phi.par,inv = inv, eig_1 = eig_1){
28 require(mvtnorm)
29 # Change of variables
30 phi1 = exp(par[1])/(1+exp(par[1]))
31 phi2 = exp(par[2])/(1+exp(par[2]))
32 t1 = exp(par[3])
33 t2 = exp(par[4])
34 gamma = exp(par[5])/(1+exp(par[5]))
35 n = dim(Q)[1]
36 # User defined scaling parameter
37 lambda_1=-log(prec.par[1,2])/prec.par[1,1]
38 lambda_2=-log(prec.par[2,2])/prec.par[2,1]
39 lambda_3= -log(1-phi.par[3,2])/dist_phi1(phi1 = phi.par[3,1],
40 gamma=gamma, Ident = Ident, Q = Q,n=n,inv = inv, eig_1 = eig_1)
41 lambda_5= -log(1-phi.par[1,2])/dist_gamma(gamma = phi.par[1,1],
42 phi1 = phi1, Ident = Ident, Q = Q,n=n,inv = inv, eig_1 = eig_1)
43 # Log density
44 ld = 0
45 ld = ld + kld_dist(phi1, gamma, Ident, Q,
46 n,lambda_3,inv = inv, eig_1 = eig_1)+ log( exp(par[1]) /

```



```

47 (1 + exp(par[1]))^2 )
48 ld = ld + INLA:::inla.pc.bym.phi(Q=Q, u= phi.par[2,1],
49 alpha = phi.par[2,2])(phi2) + log( exp(par[2]) /
50 (1 + exp(par[2]))^2 )
51 ld = ld + inla.pc.dprec(t1 , u=prec.par[1,1],
52 alpha = prec.par[1,2], log=TRUE) + log (t1)
53 ld = ld + inla.pc.dprec(t2 , u=prec.par[2,1],
54 alpha = prec.par[2,2], log=TRUE) + log (t2)
55 ld = ld + kld_dist_gamma(gamma, phi1 ,
56 Ident, Q, n, lambda_5, inv = inv , eig_1 = eig_1) +
57 log( exp(par[5]) / (1 + exp(par[5]))^2 )
58 # Building the precision matrix
59 Ident = diag(n)
60 one = matrix(1,544,1)
61 zerom = matrix(0,544,544)
62 zero = matrix(0,544,1)
63 R1 = cbind(prec.eps.1*(1-phi1)*gamma/t1*Ident+
64 prec.eps.2*phi2/t2*Ident+Q,
65 prec.eps.1*sqrt(gamma*(1-phi1)*phi1)/t1*Ident ,
66 prec.eps.1*sqrt(gamma*(1-gamma))*(1-phi1)/t1*Ident
67 +prec.eps.2*sqrt(phi2*(1-phi2))/t2*Ident ,
68 prec.eps.1*sqrt((1-phi1)*gamma/t1)*one ,
69 prec.eps.2*sqrt(phi2/t2)*one) #done
70 R2 = cbind(prec.eps.1*sqrt(gamma*(1-phi1)*phi1)/t1*Ident ,
71 prec.eps.1*phi1/t1*Ident+Q,
72 prec.eps.1*sqrt((1-phi1)*(1-gamma)*phi1)/t1*Ident ,
73 prec.eps.1*sqrt(phi1/t1)*one ,
74 zero)
75 R3 = cbind(prec.eps.1*sqrt(gamma*(1-gamma))*(1-phi1)/t1*Ident
76 +prec.eps.2*sqrt(phi2*(1-phi2))/t2*Ident ,
77 prec.eps.1*sqrt((1-phi1)*
78 (1-gamma)*phi1)/t1*Ident ,
79 (1+prec.eps.1*(1-phi1)*(1-gamma)/t1)
80 *Ident+prec.eps.2*(1-phi2)/t2*Ident ,
81 prec.eps.1*sqrt((1-phi1)*(1-gamma)/t1)*one ,
82 prec.eps.2*sqrt((1-phi2)/t2)*one)
83 R5 = cbind(prec.eps.1*sqrt((1-phi1)*gamma/t1)*t(one) ,
84 prec.eps.1*sqrt(phi1/t1)*t(one) ,
85 prec.eps.1*sqrt((1-phi1)*(1-gamma)/t1)*t(one) ,
86 prec.eps.1*n+prec.mean.1*1 ,
87 0)
88 R6 = cbind(prec.eps.2*sqrt(phi2/t2)*t(one) ,
89 t(zero) ,
90 prec.eps.2*sqrt((1-phi2)/t2)*t(one) ,

```

```

91         0,
92         prec.eps.2*n+prec.mean.2*1)
93 QQ = rbind(R1,R2,R3,R5,R6)
94 b = c(prec.eps.1*sqrt((1-phi1)*gamma/t1)*y1+
95       prec.eps.2*sqrt(phi2/t2)*y2,
96       prec.eps.1*sqrt(phi1/t1)*y1,
97       prec.eps.1*sqrt((1-gamma)*(1-phi1)/t1)*y1+
98       prec.eps.2*sqrt((1-phi2)/t2)*y2,
99       prec.eps.1*sum(y1),
100      prec.eps.2*sum(y2))
101 # GMRF precision
102 QQ = INLA::inla.as.sparse(QQ)
103 # GMRF mean
104 mu = c(inla.qsolve(QQ[1:1634,1:1634],
105                  matrix(b[1:1634], 3*n+2, 1)))
106 e=c(0,0) # constraint RHS
107 # constraint matrix
108 A=matrix(0,2,3*n+2)
109 A[1,1:544]=1
110 A[2,545:1088]=1
111 # compute density
112 ld = ld - c(inla.qsample(Q=QQ, mu=mu, sample = rep(0, 3*n+2),
113                  logdens=TRUE, constr = list(A = A, e = e))$logdens)
114 print(round(c(par,ld), digits=5))
115 return (ld)
116 }
117 # Function that compute the mode of the posterior marginal for u v mu
118 get_mode = function (par,y1,y2,Q,n){
119   phi1 = exp(par[1])/(1+exp(par[1]))
120   phi2 = exp(par[2])/(1+exp(par[2]))
121   t1 = exp(par[3])
122   t2 = exp(par[4])
123   gamma = exp(par[5])/(1+exp(par[5]))
124   Ident = diag(n)
125   one = matrix(1,544,1)
126   zerom = matrix(0,544,544)
127   zero = matrix(0,544,1)
128
129   R1 = cbind(prec.eps.1*(1-phi1)*gamma/t1*Ident+
130             prec.eps.2*phi2/t2*Ident+Q,
131             prec.eps.1*sqrt(gamma*(1-phi1)*phi1)/t1*Ident,
132             prec.eps.1*sqrt(gamma*(1-gamma))*(1-phi1)/t1*Ident+
133             prec.eps.2*sqrt(phi2*(1-phi2))/t2*Ident,
134             prec.eps.1*sqrt((1-phi1)*gamma/t1)*one,

```

```

135         prec.eps.2*sqrt(phi2/t2)*one) #done
136
137 R2 = cbind(prec.eps.1*sqrt(gamma*(1-phi1)*phi1)/t1*Ident ,
138           prec.eps.1*phi1/t1*Ident+Q,
139           prec.eps.1*sqrt((1-phi1)*(1-gamma)*phi1)/t1*Ident ,
140           prec.eps.1*sqrt(phi1/t1)*one ,
141           zero)
142
143 R3 = cbind(prec.eps.1*sqrt(gamma*(1-gamma))*(1-phi1)/t1*Ident+
144           prec.eps.2*sqrt(phi2*(1-phi2))/t2*Ident ,
145           prec.eps.1*sqrt((1-phi1)*(1-gamma)*phi1)/t1*Ident ,
146           (1+prec.eps.1*(1-phi1)*(1-gamma)/t1)*Ident+
147           prec.eps.2*(1-phi2)/t2*Ident ,
148           prec.eps.1*sqrt((1-phi1)*(1-gamma)/t1)*one ,
149           prec.eps.2*sqrt((1-phi2)/t2)*one)
150
151 R5 = cbind(prec.eps.1*sqrt((1-phi1)*gamma/t1)*t(one) ,
152           prec.eps.1*sqrt(phi1/t1)*t(one) ,
153           prec.eps.1*sqrt((1-phi1)*(1-gamma)/t1)*t(one) ,
154           prec.eps.1*n+prec.mean.1*1 ,
155           0)
156 R6 = cbind(prec.eps.2*sqrt(phi2/t2)*t(one) ,
157           t(zero) ,
158           prec.eps.2*sqrt((1-phi2)/t2)*t(one) ,
159           0 ,
160           prec.eps.2*n+prec.mean.2*1)
161 QQ = rbind(R1,R2,R3,R5,R6)
162 b = c(prec.eps.1*sqrt((1-phi1)*gamma/t1)*y1+
163       prec.eps.2*sqrt(phi2/t2)*y2 ,
164       prec.eps.1*sqrt(phi1/t1)*y1 ,
165       prec.eps.1*sqrt((1-gamma)*(1-phi1)/t1)*y1+
166       prec.eps.2*sqrt((1-phi2)/t2)*y2 ,
167       prec.eps.1*sum(y1) ,
168       prec.eps.2*sum(y2))
169
170 QQ = INLA:::inla.as.sparse(QQ) # GMRF precision
171 mu = c(inla.qsolve(QQ, matrix(b, 3*n+2, 1)))
172 e=c(0,0)
173 A=matrix(0,2,3*n+2)
174 A[1,1:544]=1
175 A[2,545:1088]=1
176 res=inla.qsample(Q=QQ, mu=mu, logdens=TRUE,
177                 constr = list(A = A, e = e), compute.mean = TRUE)
178 u1 = res$mean[1:544]

```

```

179   u2 = res$mean[545:1088]
180   v1 = res$mean[1089:1632]
181   mu1 = res$mean[1633]
182   mu2 = res$mean[1634]
183   eta1= mu1 + 1/sqrt(t1)*(sqrt(1-phi1)*(sqrt(gamma)*u1+
184   sqrt(1-gamma)*v1) + sqrt(phi1)* u2)
185   eta2 = mu2 + 1/sqrt(t2)*(sqrt(1-phi2)*v1
186   + sqrt(phi2)* u1)
187   result1 = data.frame(u1 =u1 ,u2=u2 ,v1=v1 ,mu1=mu1 ,
188   mu2=mu2,eta2 = eta2 , eta1 = eta1)
189   return(result1)
190 }

```

A.4.4 Code for the second model with the PC priors

This code computes the priors for $\phi_1, \phi_2, \tau_1, \tau_2, \gamma$, the mode of the posterior marginal of $\text{logit}(\phi_1), \text{logit}(\phi_2), \log(\tau_1), \log(\tau_2), \text{logit}(\gamma)$ and the mode of the posterior marginal of $u_1, v_1, u_2, v, \mu_1, \mu_2$.

```

1  g = system.file("demodata/germany.graph", package="INLA")
2  g=inla.read.graph(g)
3  Q=as.matrix(inla.graph2matrix(g))
4  Q = -Q
5  diag(Q) = g$nnbs
6  Q = INLA:::inla.scale.model.bym(Q)
7  diag(Q) = diag(Q) + 1e-8
8  prec.mean.1 = 0.01
9  prec.mean.2 = 0.01
10 prec.eps.1 = 4
11 prec.eps.2 = 4
12 phi.par = matrix(rep(0),3,2)
13 phi.par[1,] = c(0.5, 2/3)# .5 for all
14 phi.par[2,] = c(0.5, 2/3)
15 phi.par[3,] = c(0.5, 2/3)
16 prec.par = matrix(rep(0),2,2)
17 prec.par[1,] = c(0.2/0.31, 0.01) prec.par[2,] = c(0.2/0.31,0.01)
18 temp = as.matrix(Q)
19 inv = ginv(temp)
20 eig_1 = eigen(Q)$value[1:543]
21 eig_1 = 1/eig_1
22 Ident = diag(544)
23 n = 544
24 # KLD for phi2

```

```

25 kld_dist_phi2 = function (phi2, gamma, phi1 ,
26 eig_1 = eig_1, inv = inv, Ident = Ident, n = n){
27   Sigma_0 = ((1-gamma)*(1-phi1)+gamma)*Ident +(1-gamma)*phi1*inv
28   Sigma_1 = ((1-gamma)*(1-phi1)+gamma*(1-phi2))*Ident
29   + ((1-gamma)*phi1+gamma*phi2)*inv
30   ln_S_0 = n/(n-1)*sum(log(((1-gamma)*(1-phi1)
31   +gamma)+(1-gamma)*phi1*eig_1))
32   Tr_1 = sum(diag(solve(Sigma_0)%*%Sigma_1))
33   ln_S_0 = sum(log(((1-gamma)*(1-phi1)+gamma)
34   +(1-gamma)*phi1*eig_1))
35   ln_S_1 = sum(log( (1-gamma)*(1-phi1)+gamma*(1-phi2)
36   + ((1-gamma)*phi1+gamma*phi2)*eig_1)) # ok
37   res = sqrt( (Tr_1 - n - ln_S_1 + ln_S_0)+ 1e-1)
38 }
39 # Prior for phi2
40 prior_phi2 = function(phi2, gamma, phi1, lambda,
41 eig_1=eig_1, inv=inv, Ident=Ident, n=n){
42   result = 0
43   result = result + log(lambda)
44   result = result - lambda*kld_dist_phi2(phi2 = phi2, gamma = gamma,
45   phi1 = phi1, eig_1=eig_1, inv=inv, Ident=Ident, n=n)
46   delta = 1e-4
47   b = log(abs((kld_dist_phi2(phi2 = (phi2+delta), gamma = gamma,
48   phi1 = phi1, eig_1=eig_1, inv=inv, Ident=Ident, n=n)-
49   kld_dist_phi2(phi2 = phi2, gamma = gamma, phi1 = phi1,
50   eig_1=eig_1, inv=inv, Ident=Ident, n=n))/delta))
51   result = result + b
52   scalt = kld_dist_phi2(phi2 = 1, phi1 = phi1, gamma = gamma,
53   Ident = Ident, n=n, eig_1 = eig_1, inv = inv)
54   result = result - log(1-exp(-lambda*scalt))
55 return(result)
56 }
57 # KLD phi1
58 kld_dist_phi1 = function (phi1, phi2, inv, eig_1, n){
59   Ident = diag(n)
60   zerom = matrix(0, n, n)
61   Sigma_base = rbind(cbind(Ident, zerom), cbind(zerom, phi2*inv+(1-phi2
62   )*Ident))
63   Sigma_flex = rbind(cbind(phi1*inv+(1-phi1)*Ident, zerom),
64   cbind(zerom, phi2*inv+(1-phi2)*Ident))
65   Tr_1 = sum(diag(Sigma_flex)%*%solve(Sigma_base))
66   ln_det = - sum(log((1-phi1)+ phi1*eig_1))
67   result = sqrt(Tr_1 - 2*n + ln_det)
68 }

```

```

68 # PC prior for phi1
69 prior_phi1 = function(phi1, phi2, lambda, eig_1=eig_1, inv=inv, Ident=
    Ident, n=n){
70     result = 0
71     result = result + log(lambda)
72     result = result - lambda*kld_dist_phi1(phi1 = phi1, phi2 = phi2,
73     eig_1=eig_1, inv=inv, n=n)
74     delta = 1e-3
75     b = log(abs((kld_dist_phi1(phi1 = (phi1+delta), phi2 = phi2,
76     eig_1=eig_1, inv=inv, n=n)-kld_dist_phi1(phi1 = phi1,
77     phi2 = phi2, eig_1=eig_1, inv=inv, n=n))/delta))
78     result = result + b
79     scalt = kld_dist_phi1(phi1 = 1, phi2 = phi2,
80     n=n, eig_1 = eig_1, inv = inv)
81     result = result - log(1-exp(-lambda*scalt))
82 return(result)
83 }
84 # KLD for gamma
85 kld_dist_gamma = function (gamma, phi2, phi1, eig_1 = eig_1,
86 inv = inv, Ident = Ident, n = n){
87     Sigma_0 = (1-phi1)*Ident + phi1*inv
88     Sigma_1 = ((1-gamma)*(1-phi1)+gamma*(1-phi2))*Ident+
89     ((1-gamma)*phi1+gamma*phi2)*inv
90     Tr_1 = sum(diag(ginv(Sigma_0)%%Sigma_1))
91     ln_S_0 = sum(log(((1-phi1)+phi1*eig_1))) # ok
92     ln_S_1 = sum(log( (1-gamma)*(1-phi1)+gamma*(1-phi2)
93     + ((1-gamma)*phi1+gamma*phi2)*eig_1)) # ok
94     res = sqrt( (Tr_1 - n - ln_S_1 + ln_S_0+1e-1))
95 }
96 # Prior for gamma
97 prior_gamma = function(gamma, phi2, phi1, lambda, eig_1=eig_1,
98 inv=inv, Ident=Ident, n=n){
99     result = 0
100    result = result + log(lambda)
101    result = result - lambda*kld_dist_gamma(gamma = gamma, phi2 = phi2,
102    phi1 = phi1, eig_1=eig_1, inv=inv, Ident=Ident, n=n)
103    delta = 1e-3
104    b = log(abs((kld_dist_gamma(gamma = (gamma+delta), phi2 = phi2,
105    phi1 = phi1, eig_1=eig_1, inv=inv, Ident=Ident,
106    n=n)-kld_dist_gamma(gamma = gamma, phi2 = phi2, phi1 = phi1,
107    eig_1=eig_1, inv=inv, Ident=Ident, n=n))/delta))
108    result = result + b
109    scalt = kld_dist_gamma(gamma = 1, phi2 = phi2, phi1 = phi1,
110    Ident = Ident, n=n, eig_1 = eig_1, inv = inv)

```

```

111     result = result - log(1-exp(-lambda*scalt))
112 return(result)
113 }
114 # Function to maximise to obtain the mode of the
115 # posterior marginal for the parameters
116 logdens_joint_last = function(par, prec.eps.1 = 4, prec.eps.2 = 4,
117 y1, y2, Q, prec.mean.2 = 0.01, prec.mean.1 = 0.01,
118 prec.par, phi.par, inv, eig_1){
119     require(mvtnorm)
120 # Change of variables
121     phi1 = exp(par[1])/(1+exp(par[1]))
122     phi2 = exp(par[2])/(1+exp(par[2]))
123     t1 = exp(par[3])
124     t2 = exp(par[4])
125     gamma = exp(par[5])/(1+exp(par[5]))
126     n = dim(Q)[1]
127     Ident = diag(n)
128     one = matrix(1,n,1)
129     zerom = matrix(0,n,n)
130     zero = matrix(0,n,1)
131 # User defined scaling parameter
132 # lambdas
133     lambda_phi1= - log (1 -phi.par[1,2])/kld_dist_phi1(phi1 = phi.par
134 [1,1],
135     phi2 = phi2, eig_1=eig_1, inv=inv, n=n)
136     lambda_phi2= - log (1 -phi.par[2,2])/kld_dist_phi2(phi2 = phi.par
137 [2,1],
138     gamma=gamma, phi1 = phi1, Ident = Ident, n=n,
139     eig_1 = eig_1, inv = inv)
140     lambda_gamma= - log (1 -phi.par[3,2])/
141     kld_dist_gamma(gamma = phi.par[3,1],
142     phi2 = phi2, phi1 = phi1, Ident = Ident, n=n, eig_1 = eig_1, inv = inv)
143 # priors contribution
144     ld = 0
145     ld = ld + inla.pc.dprec(t1, u=prec.par[1,1],
146     alpha = prec.par[1,2], log=TRUE) + log (t1)
147     ld = ld + inla.pc.dprec(t2, u=prec.par[2,1],
148     alpha = prec.par[2,2], log=TRUE) + log (t2)
149     ld = ld + prior_phi1(phi1 = phi1, phi2 = phi2,
150     lambda = lambda_phi1, eig_1=eig_1, inv=inv,
151     Ident=Ident, n=n) + log( exp(par[1]) /
152     (1 + exp(par[1]))^2 )
153     ld = ld + prior_phi2(phi2 = phi2, gamma = gamma,
154     phi1 = phi1, lambda = lambda_phi2,

```

```

153   eig_1=eig_1,inv=inv,Ident=Ident,n=n) +
154   log( exp(par[2]) / (1 + exp(par[2]))^2 )
155   ld = ld + prior_gamma(gamma = gamma ,phi2 = phi2,
156   phi1 = phi1 ,lambda = lambda_gamma,eig_1=eig_1,
157   inv=inv,Ident=Ident,n=n) +
158   log( exp(par[5]) / (1 + exp(par[5]))^2 )
159
160 ##### Matrix
161
162
163   R1 = cbind(Q+prec.eps.1/t1*(1-gamma)*phil* Ident+prec.eps.2/t2*
164   phil*Ident,
165   prec.eps.1*sqrt((1-gamma)*phil*gamma*phi2)/t1*Ident,
166   prec.eps.1*(1-gamma)*sqrt(phi1*(1-phi1))/t1*Ident
167   + prec.eps.2*sqrt((1-phi1)*phi1)/t2*Ident,
168   prec.eps.1*sqrt((1-gamma)*gamma*phil*(1-phi2))/t1*Ident,
169   prec.eps.1*sqrt((1-gamma)*phil/t1)*one,
170   prec.eps.2*sqrt(phi1/t2)*one)
171
172   R2 = cbind(prec.eps.1*sqrt((1-gamma)*phil*gamma*phi2)/t1*Ident,
173   prec.eps.1*gamma*phi2/t1*Ident+Q,
174   prec.eps.1*sqrt((1-gamma)*(1-phi1)*gamma*phi2)/t1*Ident,
175   prec.eps.1*gamma/t1*sqrt(phi2*(1-phi2))*Ident,
176   prec.eps.1*sqrt(gamma*phi2/t1)*one,
177   zero)
178   R3 = cbind(prec.eps.1*(1-gamma)*sqrt(phi1*(1-phi1))/t1*Ident
179   + prec.eps.2*sqrt((1-phi1)*phi1)/t2*Ident,
180   prec.eps.1*sqrt((1-gamma)*(1-phi1)*gamma*phi2)/t1*Ident,
181   prec.eps.1*(1-gamma)*(1-phi1)/t1*Ident
182   + prec.eps.2*(1-phi1)/t2*Ident+Ident,
183   prec.eps.1*sqrt((1-gamma)*
184   gamma*(1-phi1)*(1-phi2))/t1*Ident,
185   prec.eps.1*sqrt((1-gamma)*(1-phi1)/t1)*one,
186   prec.eps.2*sqrt((1-phi1)/t2)*one
187   )
188   R4 = cbind( prec.eps.1*sqrt((1-gamma)*gamma*phil*
189   (1-phi2))/t1*Ident,
190   prec.eps.1*gamma/t1*sqrt(phi2*(1-phi2))*Ident,
191   prec.eps.1*sqrt((1-gamma)*gamma*
192   (1-phi1)*(1-phi2))/t1*Ident,
193   prec.eps.1*gamma*(1-phi2)/t1*Ident+ Ident,
194   prec.eps.1*sqrt(gamma*(1-phi2)/t1)*one,
195   zero
196   )

```

```

196   R5 = cbind( prec. eps.1*sqrt((1-gamma)*phi1/t1)*t( one) ,
197             prec. eps.1*sqrt(gamma*phi2/t1)*t( one) ,
198             prec. eps.1*sqrt((1-gamma)*(1-phi1)/t1)*t( one) ,
199             prec. eps.1*sqrt(gamma*(1-phi2)/t1)*t( one) ,
200             prec. eps.1*n+prec. mean.1*1 ,
201             0)
202   R6 = cbind( prec. eps.2*sqrt(phi1/t2)*t( one) ,
203             t( zero) ,
204             prec. eps.2*sqrt((1-phi1)/t2)*t( one) ,
205             t( zero) ,
206             0 ,
207             prec. eps.2*n+prec. mean.2*1)
208   QQ = rbind( R1, R2, R3, R4, R5, R6)
209   b = c( prec. eps.1*sqrt(phi1*(1-gamma)/t1)*y1+
210         prec. eps.2*sqrt(phi1/t2)*y2 ,
211         prec. eps.1*sqrt(gamma*phi2/t1)*y1 ,
212         prec. eps.1*sqrt((1-gamma)*(1-phi1)/t1)*y1+
213         prec. eps.2*sqrt((1-phi1)/t2)*y2 ,
214         prec. eps.2*sqrt(gamma*(1-phi2)/t1)*y2 ,
215         prec. eps.1*sum(y1) ,
216         prec. eps.2*sum(y2))
217
218   QQ = INLA:::inla.as.sparse(QQ) # GMRF precision
219   mu = c(inla.qsolve(QQ, matrix(b, 4*n+2, 1))) # GMRF mean
220   # constraint matrix
221   e=c(0,0) # constraint RHS
222   A=matrix(0,2,4*n+2)
223   A[1,1:n]=1
224   A[2,(n+1):(2*n)]=1
225   # compute density
226   ld = ld - c(inla.qsample(Q=QQ, mu=mu, sample = rep(0, 4*n+2),
227     logdens=TRUE, constr = list(A = A, e = e))$logdens)
228   print(round(c(par,ld), digits=5))
229   return (ld)
230 }
231 # Function to obtain the mode of the posterior marginal for u1 u2 v1
232     v2 mu1 mu2
232   get_mode_last = function(par, y1, y2, Q) {
233     require(mvtnorm)
234     # Change of variables
235     phi1 = exp(par[1])/(1+exp(par[1]))
236     phi2 = exp(par[2])/(1+exp(par[2]))
237     t1 = exp(par[3])
238     t2 = exp(par[4])

```

```

239  gamma = exp(par[5])/(1+exp(par[5]))
240  n = dim(Q)[1]
241  Ident = diag(n)
242  one = matrix(1,n,1)
243  zerom = matrix(0,n,n)
244  zero = matrix(0,n,1)
245
246  R1 = cbind(Q+prec.eps.1/t1*(1-gamma)*phi1* Ident+
247  prec.eps.2/t2*phi1*Ident ,
248  prec.eps.1*sqrt((1-gamma)*phi1*gamma*phi2)/t1*Ident ,
249  prec.eps.1*(1-gamma)*sqrt(phi1*(1-phi1))/t1*Ident
250  + prec.eps.2*sqrt((1-phi1)*phi1)/t2*Ident ,
251  prec.eps.1*sqrt((1-gamma)*gamma*phi1*(1-phi2))/t1*Ident ,
252  prec.eps.1*sqrt((1-gamma)*phi1/t1)*one ,
253  prec.eps.2*sqrt(phi1/t2)*one)
254
255  R2 = cbind(prec.eps.1*sqrt((1-gamma)*phi1*gamma*phi2)/t1*Ident ,
256  prec.eps.1*gamma*phi2/t1*Ident+Q,
257  prec.eps.1*sqrt((1-gamma)*(1-phi1)*gamma*phi2)/t1*Ident ,
258  prec.eps.1*gamma/t1*sqrt(phi2*(1-phi2))*Ident ,
259  prec.eps.1*sqrt(gamma*phi2/t1)*one ,
260  zero)
261  R3 = cbind(prec.eps.1*(1-gamma)*sqrt(phi1*(1-phi1))/t1*Ident
262  + prec.eps.2*sqrt((1-phi1)*phi1)/t2*Ident ,
263  prec.eps.1*sqrt((1-gamma)*(1-phi1)*gamma*phi2)/t1*Ident ,
264  prec.eps.1*(1-gamma)*(1-phi1)/t1*Ident
265  + prec.eps.2*(1-phi1)/t2*Ident+Ident ,
266  prec.eps.1*sqrt((1-gamma)*gamma*(1-phi1)*
267  (1-phi2))/t1*Ident ,
268  prec.eps.1*sqrt((1-gamma)*(1-phi1)/t1)*one ,
269  prec.eps.2*sqrt((1-phi1)/t2)*one
270  )
271  R4 = cbind( prec.eps.1*sqrt((1-gamma)*gamma*
272  phi1*(1-phi2))/t1*Ident ,
273  prec.eps.1*gamma/t1*sqrt(phi2*(1-phi2))*Ident ,
274  prec.eps.1*sqrt((1-gamma)*gamma*
275  (1-phi1)*(1-phi2))/t1*Ident ,
276  prec.eps.1*gamma*(1-phi2)/t1*Ident+ Ident ,
277  prec.eps.1*sqrt(gamma*(1-phi2)/t1)*one ,
278  zero
279  )
280  R5 = cbind(prec.eps.1*sqrt((1-gamma)*phi1/t1)*t(one) ,
281  prec.eps.1*sqrt(gamma*phi2/t1)*t(one) ,
282  prec.eps.1*sqrt((1-gamma)*(1-phi1)/t1)*t(one) ,

```

```

283         prec.eps.1*sqrt(gamma*(1-phi2)/t1)*t(one),
284         prec.eps.1*n+prec.mean.1*1,
285         0)
286 R6 = cbind(prec.eps.2*sqrt(phi1/t2)*t(one),
287           t(zero),
288           prec.eps.2*sqrt((1-phi1)/t2)*t(one),
289           t(zero),
290           0,
291           prec.eps.2*n+prec.mean.2*1)
292 QQ = rbind(R1,R2,R3,R4,R5,R6)
293 b = c(prec.eps.1*sqrt(phi1*(1-gamma)/t1)*y1+
294       prec.eps.2*sqrt(phi1/t2)*y2,
295       prec.eps.1*sqrt(gamma*phi2/t1)*y1,
296       prec.eps.1*sqrt((1-gamma)*(1-phi1)/t1)*y1+
297       prec.eps.2*sqrt((1-phi1)/t2)*y2,
298       prec.eps.2*sqrt(gamma*(1-phi2)/t1)*y2,
299       prec.eps.1*sum(y1),
300       prec.eps.2*sum(y2))
301 QQ = INLA::inla.as.sparse(QQ)
302 mu= c(inla.qsolve(QQ, matrix(b, 4*n+2, 1)))
303 e=c(0,0)
304 A=matrix(0,2,4*n+2) # constraint matrix
305 A[1,1:n]=1
306 A[2,(n+1):(2*n)]=1
307 #####
308 res=inla.qsample(Q=QQ, mu=mu,logdens=TRUE,
309                 constr = list(A = A, e = e),compute.mean = TRUE)
310 u1 = res$mean[1:544]
311 u2 = res$mean[545:1088]
312 v1 = res$mean[1089:1632]
313 v2 = res$mean[1633:2176]
314 mu1 = res$mean[2177]
315 mu2 = res$mean[2178]
316 eta1= mu1 + 1/sqrt(t1)*(sqrt(1-gamma)*(sqrt(phi1)*u1+
317 sqrt(1-phi1)*v1) + sqrt(gamma)*(sqrt(phi2)*u2+
318 sqrt(1-phi2)*v2) )
319 eta2 = mu2 + 1/sqrt(t2)*(sqrt(1-phi1)*v1 +
320 sqrt(phi1)* u1)
321 result1 = data.frame(u1 =u1 ,u2=u2 ,v1=v1 ,mu1=mu1,
322 mu2=mu2,eta2 = eta2, eta1 = eta1)
323 return(result1)
324 }

```

A.4.5 Code for the results of the second model

This code shows how the results are computed using the code already written.

```

1 solution= optim(par= solution$par,
2               fn=logdens_joint_last,
3               y1 = y,
4               y2=sqrt(lung_y),
5               Q = Q,
6               prec.par = prec.par,
7               phi.par = phi.par,
8               eig_1 = eig_1,
9               inv = inv,
10              control = list(fnscale=-1, factr=1e10),
11              method = "L-BFGS-B", lower = c(-4,-4,-4,-4))
12 res_sol = get_mode_last(solution$par, y, sqrt(lung_y), Q)
13 solution2= optim(par= solution2$par,
14                fn=logdens_joint_last,
15                y1 = sqrt(smr_unkcancer),
16                y2=sqrt(lung_y),
17                Q = Q,
18                prec.par = prec.par,
19                phi.par = phi.par,
20                eig_1 = eig_1,
21                inv = inv,
22                control = list(fnscale=-1, factr=1e10),
23                method = "L-BFGS-B", lower = c(-10,-10, -10,-10, -10),
24                upper = c(30,30,30,30,30))
25 res_sol2 = get_mode_last(solution2$par, sqrt(smr_unkcancer), sqrt(lung_
26                y), Q)
27 solution3= optim(par= solution2$par,
28                fn=logdens_joint_last,
29                y1 = sqrt(larynx),
30                y2=sqrt(lung_y),
31                Q = Q,
32                prec.par = prec.par,
33                phi.par = phi.par,
34                eig_1 = eig_1,
35                inv = inv,
36                control = list(fnscale=-1, factr=1e10),
37                method = "L-BFGS-B", lower = c(-10,-10, -10,-10, -10),
38                upper = c(30,30,30,30,30))
39 res_sol3 = get_mode_last(solution3$par, sqrt(larynx), sqrt(lung_y), Q)

```

```

39 # Function that computes the weights for the shared and disease
    specific component
40 bouiboui = function(par)
41   {
42     w1 = sqrt(par[1]*(1-par[5]))
43     w2 = sqrt(par[2]*par[5])
44     return(c(w1,w2))
45   }

```

A.4.6 code for the test using the increased mesh size

In this part is shown the code used to obtain the results with an increased mesh size

Constant risk

```

1 # Generate a sample with a constant risk
2 generate_sample = function(p){
3   a=cell2nb(p,p,torus = TRUE)
4   Q_test= as.matrix(-nb2mat(a, style="B"))
5   diag(Q_test) = -rowSums(Q_test)
6   Q_test = INLA:::inla.scale.model.bym(Q_test)
7   diag(Q_test) = diag(Q_test) + 1e-8
8   inv_test = ginv(as.matrix(Q_test))
9   eig_test = eigen(Q_test)$value[1:(length(diag(Q_test))-1)]
10  eig_test = 1/eig_test
11  n_test = dim(Q_test)[1]
12  e=c(0)
13  A=matrix(0,1,n_test)
14  A[1,1:n_test]=1
15  Ident_test = diag(n_test)
16  grid1 = rep(1,n_test)
17  grid2 = rep(1.05,n_test)
18  result = data.frame(grid1, grid2)
19  return(result)
20 }
21 # Generate the precision matrix
22 generate_mat = function(p){
23   a=cell2nb(p,p,torus = TRUE)
24   Q_test= as.matrix(-nb2mat(a, style="B"))
25   diag(Q_test) = -rowSums(Q_test)
26   Q_test = INLA:::inla.scale.model.bym(Q_test)
27   diag(Q_test) = diag(Q_test) + 1e-8

```

```

28   return(Q_test)
29 }
30 # Generate the eigen value of the precision matrix
31 generate_eig = function(p){
32   a=cell2nb(p,p,torus = TRUE)
33   Q_test= as.matrix(-nb2mat(a, style="B"))
34   diag(Q_test) = -rowSums(Q_test)
35   Q_test = INLA:::inla.scale.model.bym(Q_test)
36   diag(Q_test) = diag(Q_test) + 1e-8
37   inv_test = ginv(as.matrix(Q_test))
38   eig_test = eigen(Q_test)$value[1:(length(diag(Q_test))-1)]
39   eig_test = 1/eig_test
40   return(eig_test)}
41 # Generate the generalized inverse of the precision matrix
42 generate_inv = function(p){
43   a=cell2nb(p,p,torus = TRUE)
44   Q_test= as.matrix(-nb2mat(a, style="B"))
45   diag(Q_test) = -rowSums(Q_test)
46   Q_test = INLA:::inla.scale.model.bym(Q_test)
47   diag(Q_test) = diag(Q_test) + 1e-8
48   inv_test = ginv(as.matrix(Q_test))
49   return(inv_test)
50 }
51 # Constant risk surface test
52 # test 1 mesh size 10x10
53 p =10
54 sam1 = generate_sample(p)
55 mat1 = generate_mat(p)
56 inv1 = generate_inv(p)
57 eig1 = generate_eig(p)
58 grid_test1= optim(par= c(0.45790, -1.07327,
59   7.07158, 7.20251, -0.08590),
60   fn=logdens_joint_last ,
61   y1 = sam1$grid1 ,
62   y2=sam1$grid2 ,
63   Q = mat1 ,
64   prec.par = prec.par ,
65   phi.par = phi.par ,
66   eig_1 = eig1 ,
67   inv = inv1 ,
68   control = list(fnscale=-1, factr=1e10) ,
69   method = "L-BFGS-B", lower = c(-3,-3,-10,-10,-2) ,
70   upper = c(3,3,10,10,3))
71 #test 2 mesh size 21*21

```

```
72 p =21
73 sam2 = generate_sample(p)
74 mat2 = generate_mat(p)
75 inv2 = generate_inv(p)
76 eig2 = generate_eig(p)
77 grid_test2= optim(par= c(0.45790, -1.07327,
78     7.07158,     7.20251,     -0.08590),
79     fn=logdens_joint_last ,
80     y1 = sam2$sample1 ,
81     y2=sam2$sample1.1 ,
82     Q = mat2 ,
83     prec.par = prec.par ,
84     phi.par = phi.par ,
85     eig_1 = eig2 ,
86     inv = inv2 ,
87     control = list (fnscale=-1,factr=1e10) ,
88     method = "L-BFGS-B" ,lower = c(-3,-3,-10,-10,-3) ,
89     upper = c(3,3,10,10,3))
90 #test3 mesh size 29*29
91 p =29
92 sam3 = generate_sample(p)
93 mat3 = generate_mat(p)
94 inv3 = generate_inv(p)
95 eig3 = generate_eig(p)
96 grid_test3= optim(par= c(0.45790, -1.07327,
97     7.07158,     7.20251,     -0.08590),
98     fn=logdens_joint_last ,
99     y1 = sam3$sample1 ,
100    y2=sam3$sample1.1 ,
101    Q = mat3 ,
102    prec.par = prec.par ,
103    phi.par = phi.par ,
104    eig_1 = eig3 ,
105    inv = inv3 ,
106    control = list (fnscale=-1,factr=1e10) ,
107    method = "L-BFGS-B" ,lower = c(-3,-3,-10,-10,-3) ,
108    upper = c(3,3,10,10,3))
109 #test4
110 p = 33
111 sam4 = generate_sample(p)
112 mat4 = generate_mat(p)
113 inv4 = generate_inv(p)
114 eig4 = generate_eig(p)
115 grid_test4= optim(par= grid_test3$par ,
```

```

116         fn=logdens_joint_last ,
117         y1 = sam4$sample1 ,
118         y2=sam4$sample1.1 ,
119         Q = mat4 ,
120         prec.par = prec.par ,
121         phi.par = phi.par ,
122         eig_1 = eig4 ,
123         inv = inv4 ,
124         control = list (fn scale=-1,factr=1e10) ,
125         method = "L-BFGS-B",lower = c(-3,-3,-10,-10,-3) ,
126         upper = c(3,3,10,10,3))

```

No disease specific effect

```

1  # Test with no disease specific effect
2  generate_sample_spec = function(p){
3      a=cell2nb(p,p,torus = TRUE)
4      Q_test= as.matrix(-nb2mat(a, style="B"))
5      diag(Q_test) = -rowSums(Q_test)
6      Q_test = INLA:::inla.scale.model.bym(Q_test)
7      diag(Q_test) = diag(Q_test)+1e-8
8      inv_test = ginv(as.matrix(Q_test))
9      eig_test = eigen(Q_test)$value[1:(length(diag(Q_test))-1)]
10     eig_test = 1/eig_test
11     n_test = dim(Q_test)[1]
12     e=c(0)
13     A=matrix(0,1,n_test)
14     A[1,1:n_test]=1
15     Ident_test = diag(n_test)
16     u1 = inla.qsample(Q=Q_test,mu=rep(0,n_test),compute.mean = TRUE,
17     constr = list(A = A, e = e))$sample
18     b1 = mvrnorm(n = 1, mu=rep(0,n_test), Sigma=Ident_test, tol = 1e
19     -6,
20     empirical = FALSE, EISPACK = FALSE)
21     grid1 = 1/3*(1/4*u1+1/4*b1)
22     grid2 = 1/3*(1/4*u1+1/4*b1)
23     result = data.frame(v1= grid1, v2 =grid2)
24     return(result)
25 }
26 #test 1 mesh size 10x10
27 p =10
28 sam21 = generate_sample_spec(p)
29 mat21 = generate_mat(p)

```

```

29 inv21 = generate_inv(p)
30 eig21 = generate_eig(p)
31
32 grid_test21= optim(par= c(0.45790,  -1.07327,
33   7.07158,    7.20251,   -0.08590),
34               fn=logdens_joint_last ,
35               y1 = sam21$sample1 ,
36               y2=sam21$sample1.1 ,
37               Q = mat21 ,
38               prec.par = prec.par ,
39               phi.par = phi.par ,
40               eig_1 = eig21 ,
41               inv = inv21 ,
42               control = list (fn*scale=-1,factr=1e10) ,
43               method = "L-BFGS-B",lower = c(-2,-2,-2,-2,-2) ,
44               upper = c(2,2,10,10,2))
45 #test 2 mesh size 21x 21
46 p =21
47 sam22 = generate_sample_spec(p)
48 mat22 = generate_mat(p)
49 inv22 = generate_inv(p)
50 eig22 = generate_eig(p)
51 grid_test22= optim(par= c(0.45790,  -1.07327,    7.07158,    7.20251,
52   -0.08590) ,
53               fn=logdens_joint_last ,
54               y1 = sam22$sample1 ,
55               y2=sam22$sample1.1 ,
56               Q = mat22 ,
57               prec.par = prec.par ,
58               phi.par = phi.par ,
59               eig_1 = eig22 ,
60               inv = inv22 ,
61               control = list (fn*scale=-1,factr=1e10) ,
62               method = "L-BFGS-B",lower = c(-3,-3,-10,-10,-3) ,
63               upper = c(3,3,10,10,3))
64 # test 3 mesh size 29 x29
65 p =29
66 sam23 = generate_sample_spec(p)
67 mat23 = generate_mat(p)
68 inv23 = generate_inv(p)
69 eig23 = generate_eig(p)
70 grid_test23= optim(par= grid_test22$par ,
71               fn=logdens_joint_last ,
72               y1 = sam23$sample1 ,

```

```
72         y2=sam23$sample1.1 ,
73         Q = mat23 ,
74         prec.par = prec.par ,
75         phi.par = phi.par ,
76         eig_1 = eig23 ,
77         inv = inv23 ,
78         control = list( fnscale=-1,factr=1e10) ,
79         method = "L-BFGS-B",lower = c(-3,-3,-10,-10,-3) ,
80         upper = c(3,3,10,10,3))
81
82 #test4 mesh size 33*33
83 p =33
84 sam24 = generate_sample_spec(p)
85 mat24 = generate_mat(p)
86 inv24 = generate_inv(p)
87 eig24 = generate_eig(p)
88 grid_test24= optim(par= grid_test23$par ,
89                   fn=logdens_joint_last ,
90                   y1 = sam24$sample1 ,
91                   y2=sam24$sample1.1 ,
92                   Q = mat24 ,
93                   prec.par = prec.par ,
94                   phi.par = phi.par ,
95                   eig_1 = eig24 ,
96                   inv = inv24 ,
97                   control = list( fnscale=-1,factr=1e10) ,
98                   method = "L-BFGS-B",lower = c(-3,-3,-10,-10,-3) ,
99                   upper = c(3,3,10,10,3))
```

Appendix B

Additional explanations

B.1 Precision matrix

$$\begin{aligned}Q_{uu} &= R + \frac{\phi}{\tau} \frac{I}{\sigma^2} \\Q_{vv} &= I + \frac{1-\phi}{\tau} \frac{I}{\sigma^2} \\Q_{\mu\mu} &= \frac{1}{\sigma_{max}^2} + \frac{n}{\sigma^2} \\Q_{uv} &= \frac{\sqrt{(1-\phi)\phi}}{\tau} \frac{I}{\sigma^2} \\Q_{vu} &= \frac{\sqrt{(1-\phi)\phi}}{\tau} \frac{I}{\sigma^2} \\Q_{\mu v} &= \sqrt{\frac{\phi}{\tau}} \frac{(1, \dots, 1)}{\sigma^2} \\Q_{\mu u} &= \sqrt{\frac{1-\phi}{\tau}} \frac{(1, \dots, 1)}{\sigma^2} \\Q_{uz} &= -\sqrt{\frac{1-\phi}{\tau}} \frac{I}{\sigma^2} \\Q_{vz} &= -\sqrt{\frac{\phi}{\tau}} \frac{I}{\sigma^2} \\Q_{\mu z} &= -\frac{(1, \dots, 1)}{\sigma^2}\end{aligned}$$

B.2 Precision matrix First model with PC priors

$$\begin{aligned}
Q_{u_1 u_1} &= Q^* + \left(\tau_{\epsilon_1} \frac{(1 - \phi_1)\gamma}{\tau_1} + \tau_{\epsilon_2} \frac{\phi_2}{\tau_2} \right) I \\
Q_{u_1 u_2} &= \tau_{\epsilon_1} \frac{\sqrt{\gamma(1 - \phi_1)\phi_1}}{\tau_1} I \\
Q_{u_1 v_1} &= \tau_{\epsilon_1} \frac{\sqrt{\gamma(1 - \gamma)(1 - \phi_1)}}{\tau_1} I + \tau_{\epsilon_2} \frac{\sqrt{\phi_2(1 - \phi_2)}}{\tau_2} I \\
Q_{u_1 \mu_1} &= \tau_{\epsilon_1} \frac{\sqrt{(1 - \phi_1)\gamma}}{\sqrt{\tau_1}} (1, \dots, 1)^T \\
Q_{u_1 \mu_2} &= \tau_{\epsilon_2} \frac{\sqrt{\phi_2}}{\sqrt{\tau_2}} (1, \dots, 1)^T \\
Q_{u_2 u_2} &= Q^* + \tau_{\epsilon_1} \frac{\phi_1}{\tau_1} I \\
Q_{u_2 v_1} &= \tau_{\epsilon_1} \frac{\sqrt{(1 - \gamma)\phi_1(1 - \phi_1)}}{\tau_1} I \\
Q_{u_2 \mu_1} &= \tau_{\epsilon_1} \frac{\sqrt{\phi_1}}{\sqrt{\tau_1}} (1, \dots, 1)^T \\
Q_{u_2 \mu_2} &= 0 \\
Q_{v_1 v_1} &= \left(1 + \frac{\tau_{\epsilon_1}(1 - \phi_1)(1 - \gamma)}{\tau_1} \right) I + \left(1 + \frac{\tau_{\epsilon_2}(1 - \phi_2)}{\tau_2} \right) I \\
Q_{v_1 \mu_1} &= \frac{\tau_{\epsilon_1} \sqrt{(1 - \phi_1)(1 - \gamma)}}{\sqrt{\tau_1}} (1, \dots, 1)^T \\
Q_{v_1 \mu_2} &= \frac{\tau_{\epsilon_2} \sqrt{1 - \phi_2}}{\sqrt{\tau_2}} (1, \dots, 1)^T \\
Q_{\mu_1 \mu_1} &= \tau_{\epsilon_1} n + \tau_{\max_1} \\
Q_{\mu_1 \mu_2} &= 0 \\
Q_{\mu_2 \mu_2} &= \tau_{\epsilon_2} n + \tau_{\max_2}
\end{aligned}$$

B.3 Precision matrix Second model with PC priors

$$\begin{aligned}
Q_{u_1 u_1} &= Q + \tau_{\epsilon_1} \frac{1}{\tau_1} (1 - \gamma) \phi_1 I + \tau_{\epsilon_2} \frac{1}{\tau_2} \phi_1 I \\
Q_{u_1 u_2} &= \tau_{\epsilon_1} \frac{\sqrt{(1 - \gamma) \phi_1 \gamma \phi_2}}{\tau_1} \\
Q_{u_1 v_1} &= \left(\tau_{\epsilon_1} \frac{(1 - \gamma) \sqrt{\phi_1 (1 - \phi_1)}}{\tau_1} + \tau_{\epsilon_2} \frac{\sqrt{\phi_1 (1 - \phi_1)}}{\tau_2} \right) I \\
Q_{u_1 v_2} &= \tau_{\epsilon_1} \frac{\sqrt{(1 - \gamma) (\phi_1) (1 - \phi_2) \gamma}}{\tau_1} I \\
Q_{u_1 \mu_1} &= \tau_{\epsilon_1} \frac{\sqrt{(1 - \gamma) \phi_1}}{\sqrt{\tau_1}} (1, \dots, 1)^T \\
Q_{u_1 \mu_2} &= \tau_{\epsilon_2} \frac{\sqrt{\phi_1}}{\sqrt{\tau_2}} (1, \dots, 1)^T \\
Q_{u_2 u_2} &= \tau_{\epsilon_1} \frac{\gamma \phi_2}{\tau_1} I + Q \\
Q_{u_2 v_1} &= \tau_{\epsilon_1} \frac{\sqrt{(1 - \gamma) (1 - \phi_1) \gamma \phi_2}}{\tau_1} I \\
Q_{u_2 v_2} &= \tau_{\epsilon_1} \frac{\gamma \sqrt{\phi_2 (1 - \phi_2)}}{\tau_1} \\
Q_{u_2 \mu_1} &= \tau_{\epsilon_1} \frac{\sqrt{\gamma \phi_2}}{\tau_1} (1, \dots, 1)^T \\
Q_{u_2 \mu_2} &= 0 \\
Q_{v_1 v_1} &= \left(\tau_{\epsilon_1} \frac{(1 - \gamma) (1 - \phi_1)}{\tau_1} + \tau_{\epsilon_2} \frac{1 - \phi_1}{\tau_2} + 1 \right) I \\
Q_{v_1 v_2} &= \tau_{\epsilon_1} \frac{\sqrt{(1 - \gamma) \gamma (1 - \phi_1) (1 - \phi_2)}}{\tau_1} \\
Q_{v_1 \mu_1} &= \tau_{\epsilon_1} \frac{\sqrt{(1 - \gamma) (1 - \phi_1)}}{\tau_1} (1, \dots, 1)^T \\
Q_{v_1 \mu_2} &= \tau_{\epsilon_2} \frac{\sqrt{1 - \phi_2}}{\tau_2} (1, \dots, 1)^T \\
Q_{v_2 v_2} &= \left(\tau_{\epsilon_1} \frac{\gamma (1 - \phi_2)}{\tau_1} + 1 \right) I \\
Q_{v_2 \mu_1} &= \tau_{\epsilon_1} \sqrt{\frac{\gamma (1 - \phi_2)}{\tau_1}} \\
Q_{v_2 \mu_2} &= 0 \\
Q_{\mu_1 \mu_1} &= \tau_{\epsilon_1} n + \tau_{\max_1} \\
Q_{\mu_1 \mu_2} &= 0 \\
Q_{\mu_2 \mu_2} &= \tau_{\epsilon_2} n + \tau_{\max_2}
\end{aligned}$$

Bibliography

- Besag, York, and Mollié. Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1):1–20, 1991.
- Held, Natário, Fenton, Rue, and Becker. Towards joint disease mapping. *Statistical Methods in Medical Research*, 14(1):61–82, 2005.
- Knorr-Held and Becker. Bayesian modelling of spatial heterogeneity in disease maps with application to German cancer mortality data. *Journal of the German Statistical Society (Allgemeines Statistisches Archiv)*, 84, 1999.
- Knorr-Held and Best. A shared component model for detecting joint and selective clustering of two diseases. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 164(1):73–85, 2001.
- Kullback and Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.
- Lawson. *Bayesian Disease Mapping: Hierarchical Modeling in Spatial Epidemiology, Second Edition*. Chapman & Hall/CRC Interdisciplinary Statistics. Taylor & Francis, 2013.
- Nocedal and Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Berlin, 2006.
- Riebler, Sorbye, Simpson, and Rue. An intuitive Bayesian spatial model for disease mapping that accounts for scaling. *arXiv:1601.01180*, 2016.
- Rue and Held. *Gaussian Markov Random Fields : Theory and Applications*. Chapman and Hall/ CRC, 2nd edition, 2005.
- Rue, Martino, and Chopin. Approximate Bayesian inference for latent Gaussian models using integrated nested Laplace approximations (with discussion). *Journal of the Royal Statistical Society, Series B*, 71(2):319–392, 2009.
- Simpson, Rue, Martins, Riebler, and Sorbye. Penalizing model component complexity: A principled, practical approach to constructing priors. *arXiv:1403.4630*, 2015.

Sorbye and Rue. Scaling intrinsic Gaussian Markov Random Field priors in spatial modeling. *Spatial Statistics*, 8:39 – 51, 2014.

Wakefield. Disease mapping and spatial regression with count data. *Biostatistics*, 8(2): 158–183, 2007.

www.cancer.org. American cancer society. URL www.cancer.org.