

# Supplementary Materials for “Bayesian Generalized Two-way ANOVA Modeling for Functional Data Using INLA”

Yu Ryan Yue<sup>a</sup>, David Bolin<sup>b</sup>, Håvard Rue<sup>c</sup>, and Xiao-Feng Wang<sup>d\*</sup>

<sup>a</sup>*Baruch College, The City University of New York, USA.*

<sup>b</sup>*Chalmers University of Technology, Sweden.*

<sup>c</sup>*Norwegian University of Science and Technology, Norway.*

<sup>d</sup>*Cleveland Clinic Lerner Research Institute, USA.*

## 1 INLA Demonstration

In this section we demonstrate how to implement INLA for functional ANOVA analysis in the real data examples given in the article. Note that besides INLA you also need `excursions` and `brinla` R packages. The former is available on CRAN, and the latter is under development by the authors and can be downloaded from <https://github.com/julianfaraway/brinla>.

### 1.1 Diffusion tensor imaging

Let's first load required packages and the data set

```
library(INLA)
library(excursions)
library(brinla)
data(DTI, package = 'refund')
```

The data description can be found in the R document. We study a subset of 18 patients, who had completed 4 visits within approximately one year

```
DTI.sub <- DTI[DTI$Nscans==4 & DTI$case == 1, ]
```

Let  $y_{ij}(x)$  denote the FA measure at location  $x$  for  $i$ th visit from  $j$ th subject. We assume  $y_{ij}(x)$  follows a beta distribution, denoted by  $Beta(p_{ij}(x), \tau)$ , with mean  $p_{ij}(x)$  and variance  $p_{ij}(x)(1 - p_{ij}(x))/(1 + \tau)$ . To study the visit effect on FA measure, the following model is considered

$$\text{logit}(p_{ij}) = \mu(x) + \alpha_i(x), \quad i = 1, 2, 3, 4, \quad (1)$$

---

\*Corresponding author. Email: wangx6@ccf.org

where  $\mu$  is the grand mean function and  $\alpha_i$  is the main effect function of  $i$ th visit. For identifiability we let  $\alpha_i(x) = 0$  for all  $x$ . Letting  $\boldsymbol{\mu}$  and  $\boldsymbol{\alpha}$  be the vectors of the realizations of  $\mu$  and  $\alpha_i$ , we then can write down model (1) in matrix form

$$\text{logit}(p) = \mathbf{A}_\mu \boldsymbol{\mu} + \mathbf{A}_\alpha \boldsymbol{\alpha} = \mathbf{A} \mathbf{f},$$

where  $\mathbf{A} = [\mathbf{A}_\mu, \mathbf{A}_\alpha]$  and  $\mathbf{f} = [\boldsymbol{\mu}', \boldsymbol{\alpha}']'$ . Here  $\mathbf{A}_\mu$  and  $\mathbf{A}_\alpha$  are incidence matrices used to map  $\boldsymbol{\mu}$  and  $\boldsymbol{\alpha}$  to  $p$ . We assign independent RW2 priors for  $\boldsymbol{\mu}$  and  $\alpha_i$ , i.e.,  $\mu \sim \text{RW2}(\theta_\mu)$ ,  $\alpha_i \sim \text{RW2}(\theta_\alpha)$  for all  $i$ . Regarding the precision  $\tau$ , we use the default gamma prior as specified in INLA.

To fit the model let's first make the vector  $\mathbf{y}$  of all FA measures and its corresponding  $\mathbf{A}$  matrix

```
y <- as.vector(t(DTI.sub$rcst))
ns <- dim(DTI.sub$rcst)[2]      ## number of locations
ng <- length(unique(DTI.sub$visit)) ## number of visits (groups)
n <- length(unique(DTI.sub$ID))  ## number of subjects

tmp1 <- Diagonal(n = ns, x = 1)
tmp2 <- Matrix(rep(1, ng*n), ng*n, 1)
A.mu <- kronecker(tmp2, tmp1)

tmp1 <- Diagonal(n = ns, x = 1)
tmp2 <- Diagonal(n = (ng-1), x = 1)
tmp3 <- Matrix(rep(0, ng-1), 1, ng-1)
tmp4 <- kronecker(rBind(tmp2, tmp3), tmp1)
A.a <- kronecker(Matrix(rep(1, n), n, 1), tmp4)
A <- cBind(A.mu, A.a)
```

Then, we make the index vectors for  $\boldsymbol{\mu}$ ,  $\boldsymbol{\alpha}$  and their replicates

```
mu <- 1:ns
alpha <- rep(1:ns, ng-1)
alpha.rep <- rep(1:(ng-1), each = ns)
```

and add a few NA's to those vectors in order to make them conformable with  $\mathbf{A}$

```
mu2 <- c(mu, rep(NA, length(alpha)))
alpha2 <- c(rep(NA, length(mu)), alpha)
alpha2.rep <- c(rep(NA, length(mu)), alpha.rep)
```

Finally, we fit model (1) in INLA as follows

```
data.inla <- list(y = y, mu = mu2, alpha = alpha2, alpha.rep = alpha2.rep)
formula <- y ~ -1 + f(mu, model = 'rw2', constr = FALSE, scale.model = TRUE) +
  ↪ f(alpha, model = 'rw2', constr = FALSE, scale.model = TRUE, replicate =
  ↪ alpha.rep)
result <- inla(formula, data = data.inla, family = 'beta', control.predictor =
  ↪ list(A = A, compute = TRUE), control.compute = list(config = TRUE))
```

The posterior summary of each main effect function can be retrieved via

```
result$summary.random$alpha
```

We obtain the 95% simultaneous credible band for each main effect function via

```
res.ci.a1 <- simconf.inla(result, name = 'alpha', ind=1:ns, alpha = 0.05)
res.ci.a2 <- simconf.inla(result, name = 'alpha', ind=1:ns+ns, alpha = 0.05)
res.ci.a3 <- simconf.inla(result, name = 'alpha', ind=1:ns+2*ns, alpha = 0.05)
```

Let's plot the result for the main effect of Visit #1

```

plot(result$summary.random$alpha$mean[1:ns], type = 'l', ylim = c(-.6, 0.15))
lines(res.ci.a1$a)
lines(res.ci.a1$b)
lines(res.ci.a1$a.marginal, lty = 2)
lines(res.ci.a1$b.marginal, lty = 2)
abline(0,0)

```

The joint and marginal probabilities that the main effect functions are non-zeroes for each visit can be computed as follows

```

res.exc1 <- excursions.brinla(result, name = 'alpha', ind = 1:ns, u = 0, type
  ↪ = '!=', alpha = 0.05, method = 'NIQC')
res.exc2 <- excursions.brinla(result, name = 'alpha', ind = 1:ns+ns, u = 0,
  ↪ type = '!=', alpha = 0.05, method = 'NIQC')
res.exc3 <- excursions.brinla(result, name = 'alpha', ind = 1:ns+2*ns, u = 0,
  ↪ type = '!=', alpha = 0.05, method = 'NIQC')

```

We then may plot the result of, for example, Visit #1

```
bri.excursions.ggplot(res.exc1)
```

The resulting shaded region, denoted by  $D$ , contains the locations where the main effect function of Visit #1 is not zero with at least 0.95 joint probability, i.e.,  $P(\alpha_1(\mathbf{x}) \neq 0) \geq 0.95$  for all  $\mathbf{x} \in D$ .

## 1.2 Apgar scores

Let's load the data, together with map file for plotting the results and required packages:

```

library(INLA)
library(excursions)
library(spdep)
dat <- read.csv('born2012PR.csv')
load('prmap.RData') ## map file

```

Recall the model used in this example is

$$\begin{aligned}
 y_{ijk}(\mathbf{x}) \mid p_{ijk} &\sim \text{Bin}(p_{ijk}(\mathbf{x})), \quad \mathbf{x} \in \mathcal{X} \subset \mathbf{R}^2 \\
 \text{logit}(p_{ijk}(\mathbf{x})) &= \phi_0 + z_{1ijk}\phi_1 + z'_{2ijk}\phi_2 + \mu_{ij}(\mathbf{x}) \\
 \mu_{ij}(\mathbf{x}) &= \mu(\mathbf{x}) + \alpha_i(\mathbf{x}) + \beta_j(\mathbf{x}) + \gamma_{ij}(\mathbf{x}), \tag{2}
 \end{aligned}$$

where we take diffuse normal priors on the linear effects  $(\phi_0, \phi_1, \phi_2)$ , and the Besag priors on the effect functions  $(\mu, \alpha_i, \beta_j, \gamma_{ij})$ . To implement the Besag model, we need to build a graph that contains the neighbor information

```

dat$area <- match(dat$CODMUNRES, substr(as.character(prmap$GEOCODIGO), 1, 6))
graph <- Matrix(nb2mat(poly2nb(prmap), style = 'B'))

```

The resulting graph is a sparse incidence matrix, where the neighbors for each city are labelled by value 1. Then, we need to select the variables of interest from the data

```

dat$y <- (dat$APGAR1 < 7) + 0 ## binary response
dat$weight <- dat$PESO ## weight
dat$dur <- dat$GESTACAO ## duration
dat$care <- ifelse(dat$CONSULTAS > 3, 1, 0) ## consultation
dat$edu <- ifelse(dat$ESMAE > 4, 1, 0) ## education

```

```
dat$care[dat$CONSULTAS==9] <- NA
dat$edu[dat$ESMAE==9] <- NA
dat$dur[dat$GESTACAO==9] <- NA
```

Note that there are a few missing values (labeled by ‘9’) in the covariates, so we relabel them as NA and remove them from the data

```
bdat <- dat[which(rowSums(is.na(dat[c('care', 'edu', 'weight', 'dur')]))==0),]
```

We also relabel the missing values in the binary response as NA, and make *duration* a categorical variable in R

```
bdat$y[bdat$APGAR1==99] <- NA
bdat$dur <- as.factor(c('21orLess', '22-27w', '28-31w', '32-36w', '37-41w', '
  ↪ 42+w')[bdat$dur])
```

As mentioned in the paper, we may simplify model (2) by using the same function for each effect level but with different weights. The weights can be made as follows

```
wt.a <- wt.b <- wt.ab <- rep(NA, dim(bdat)[1])
wt.a[bdat$care==1] <- 1
wt.a[bdat$care==0] <- -1
wt.b[bdat$edu==1] <- 1
wt.b[bdat$edu==0] <- -1
wt.ab[bdat$care==1 & bdat$edu==1] <- 1
wt.ab[bdat$care==1 & bdat$edu==0] <- -1
wt.ab[bdat$care==0 & bdat$edu==1] <- -1
wt.ab[bdat$care==0 & bdat$edu==0] <- 1
```

where *wt.a*, *wt.b* and *wt.ab* are the vectors of weights for the effect functions  $\alpha$ ,  $\beta$  and  $\gamma$ , respectively. Then, model (2) can be represented in INLA as

```
dat.inla <- list(y = bdat$y, weight = bdat$weight, duration = bdat$dur, sm =
  ↪ bdat$area, sa = bdat$area, sb = bdat$area, sab = bdat$area, wt.a = wt.a,
  ↪ wt.b = wt.b, wt.ab = wt.ab)
formula <- y ~ 1 + weight + duration + f(sm, model = 'besag', graph = graph,
  ↪ scale.model = TRUE) + f(sa, wt.a, model = 'besag', graph = graph, scale.
  ↪ model = TRUE) + f(sb, wt.b, model = 'besag', graph = graph, scale.model
  ↪ = TRUE) + f(sab, wt.ab, model = 'besag', graph = graph, scale.model =
  ↪ TRUE)
```

and we fit the model as follows

```
result <- inla(formula, data = dat.inla, family = 'binomial', control.compute
  ↪ = list(config = TRUE))
```

Now let’s take a look at the result. The linear effect estimates are given by

```
summary(result)$fixed
```

	mean	sd	0.025quant	0.5quant	0.975quant	mode	kld
(Intercept)	1.0351	0.2288	0.5860	1.0351	1.4841	1.0350	0
weight	-0.0006	0.0000	-0.0006	-0.0006	-0.0005	-0.0006	0
duration22-27w	0.3053	0.2401	-0.1657	0.3052	0.7765	0.3049	0
duration28-31w	-0.8963	0.2315	-1.3505	-0.8964	-0.4419	-0.8967	0
duration32-36w	-1.6447	0.2259	-2.0875	-1.6449	-1.2011	-1.6454	0
duration37-41w	-2.3641	0.2260	-2.8071	-2.3644	-1.9199	-2.3649	0
duration42+w	-2.1507	0.2347	-2.6108	-2.1509	-1.6897	-2.1514	0

The posterior means of each function effect can be extracted using

```
mu.est <- result$summary.random$sm$mean
alpha.est <- result$summary.random$sa$mean
beta.est <- result$summary.random$sb$mean
gamma.est <- result$summary.random$sab$mean
```

and they can be plotted on maps as follows

```
post.mean <- data.frame(cbind(mu.est, alpha.est, beta.est, gamma.est))
names(post.mean) <- c("mu", "alpha", "beta", "gamma")
pr.m.tmp <- prmap
pr.m.tmp@data <- data.frame(pr.m.tmp@data, post.mean)
spplot(pr.m.tmp, c("mu"))
spplot(pr.m.tmp, c("alpha"))
spplot(pr.m.tmp, c("beta"))
spplot(pr.m.tmp, c("gamma"))
```

To identify the cities where the main effects and/or interaction are statistically significant, we need to compute the level-zero contour avoiding function for each effect

```
res.a <- excursions.inla(result, name='sa', u=0, type='!=', method='NIQC')
res.b <- excursions.inla(result, name='sb', u=0, type='!=', method='NIQC')
res.ab <- excursions.inla(result, name='sab', u=0, type='!=', method='NIQC')
```

We then extract each contour avoiding function and the corresponding 95%-level avoiding set

```
F.a <- res.a$F
F.a95 <- rep(NA, length(F.a))
F.a95[F.a>=.95] <- F.a[F.a>=.95]
F.b <- res.b$F
F.b95 <- rep(NA, length(F.b))
F.b95[F.b>=.95] <- F.b[F.b>=.95]
F.ab <- res.ab$F
F.ab <- rep(NA, length(F.ab))
F.ab[F.ab>=.95] <- F.ab[F.ab>=.95]
```

and plot the results as follows

```
F.exc <- data.frame(cbind(F.a, F.b, F.ab))
F.95 <- data.frame(cbind(F.a95, F.b95, F.ab95))
names(F.exc) <- c("a.exc", "b.exc", "ab.exc")
names(F.95) <- c("a.95", "b.95", "ab.95")
pr.m.tmp <- prmap
pr.m.tmp@data <- data.frame(pr.m.tmp@data, F.exc, F.95)
spplot(pr.m.tmp, "a.exc")
spplot(pr.m.tmp, "b.exc")
spplot(pr.m.tmp, "ab.exc")
spplot(pr.m.tmp, "b.95")
```

### 1.3 Global solar radiation

Recall the model used in this example is

$$y_{ij}(\mathbf{x}) \mid \mu_i, \phi_i, \tau \sim N(\mu_i(\mathbf{x}) + \phi_i(z), \tau^{-1})$$

$$\mu_i(\mathbf{x}) = \mu(\mathbf{x}) + \alpha_i(\mathbf{x}), \quad \mathbf{x} \subset \mathbf{R}^3, \quad z \subset \mathbf{R}. \quad (3)$$

Let  $\mathbf{y}$ ,  $\boldsymbol{\mu}$ ,  $\boldsymbol{\alpha}$  and  $\boldsymbol{\phi}$  be the vectors of the realizations of  $y_{ij}$ ,  $\mu$ ,  $\alpha_i$  and  $\phi_i$ , respectively. Then we may write down model (3) in matrix form

$$\begin{aligned}\mathbf{y} &= \mathbf{A}_\mu \boldsymbol{\mu} + \mathbf{A}_\alpha \boldsymbol{\alpha} + \mathbf{A}_\phi \boldsymbol{\phi} + \boldsymbol{\varepsilon}, \\ &= \mathbf{A} \mathbf{f} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \tau^{-1} \mathbf{I})\end{aligned}$$

where  $\mathbf{A} = [\mathbf{A}_\mu, \mathbf{A}_\alpha, \mathbf{A}_\phi]$  and  $\mathbf{f} = (\boldsymbol{\mu}', \boldsymbol{\alpha}', \boldsymbol{\phi}')'$ . Here  $\mathbf{A}_\mu$ ,  $\mathbf{A}_\alpha$  and  $\mathbf{A}_\phi$  are the incidence matrices used to map the corresponding effects to the response observations.

For ease of computation, we lower the resolution of the data by randomly selecting about 4000 locations. We also spatially smooth the data using Gaussian kernel to fulfill the normality assumption in model (3). Let's load the data file and required packages

```
library(INLA)
library(excursions)
library(fields)
load(file = 'direct_insolation.RData')
```

which has two components: `loc.xyz` is a matrix of 3 columns giving the locations on the globe, and `y.mat` is a matrix of 12 columns giving direct insolation measurements. To fit model (3) with INLA, we first build a mesh on the globe and its corresponding SPDE object

```
mesh <- inla.mesh.create(globe = 15)
spde <- inla.spde2.matern(mesh)
n.mesh <- mesh$n
```

Then we build the design matrix  $\mathbf{A}$

```
ns <- dim(y.mat)[1] # number of locations
ng <- 4 # number of seasons
n <- 3 # number of months in each season

tmp1 <- matrix(c(rep(1,n), rep(0,n*(ng-1))), n*ng, 1)
A.lat1 <- kronecker(tmp1, Diagonal(n=ns, x=1))
tmp2 <- matrix(c(rep(0,n), rep(1,n), rep(0,n*(ng-2))), n*ng, 1)
A.lat2 <- kronecker(tmp2, Diagonal(n=ns, x=1))
tmp3 <- matrix(c(rep(0,2*n), rep(1,n), rep(0,n*(ng-3))), n*ng, 1)
A.lat3 <- kronecker(tmp3, Diagonal(n=ns, x=1))
tmp4 <- matrix(c(rep(0,3*n), rep(1,n)), n*ng, 1)
A.lat4 <- kronecker(tmp4, Diagonal(n=ns, x=1))

tmp1 <- Diagonal(n=n.mesh, x=1)
tmp2 <- Diagonal(n=(ng-1), x=1)
tmp3 <- Matrix(rep(0, ng-1), 1, ng-1)
A.a1 <- kronecker(rBind(tmp2,tmp3), tmp1)
A.a2 <- inla.spde.make.A(mesh, loc = loc.xyz, index = rep(1:ns, times = ng*n),
  ↪ repl = rep(1:ng, each = ns*n))
A.alpha = A.a2**%A.a1
A.mu <- inla.spde.make.A(mesh, loc = loc.xyz, index = rep(1:ns, times = ng*n))
A <- cBind(A.lat1, A.lat2, A.lat3, A.lat4, A.mu, A.alpha)
```

and make index vectors for the model effects

```
loc.xy <- inla.mesh.map(loc.xyz, inverse = F, projection = 'longlat')
lat <- loc.xy[,2]
mu.idx <- 1:n.mesh
```

```

alpha.idx <- rep(1:n.mesh, ng-1)
alpha.rep.idx <- rep(1:(ng-1), each = n.mesh)

lat1 <- c(lat, rep(NA, 3*ns), rep(NA, 4*n.mesh))
lat2 <- c(rep(NA, 1*ns), lat, rep(NA, 2*ns), rep(NA, 4*n.mesh))
lat3 <- c(rep(NA, 2*ns), lat, rep(NA, 1*ns), rep(NA, 4*n.mesh))
lat4 <- c(rep(NA, 3*ns), lat, rep(NA, 4*n.mesh))

mu <- c(rep(NA, 4*ns), mu.idx, rep(NA, 3*n.mesh))
alpha <- c(rep(NA, 4*ns), rep(NA, n.mesh), alpha.idx)
alpha.rep <- c(rep(NA, 4*ns), rep(NA, n.mesh), alpha.rep.idx)

```

Finally, we fit the model as follows:

```

dat.inla <- list(y = as.vector(yamat), lat1 = lat1, lat2 = lat2, lat3 = lat3,
  ↪ lat4 = lat4, mu = mu, alpha = alpha, alpha.rep = alpha.rep)
formula <- y ~ -1 + f(lat1, model = 'rw2') + f(lat2, model = 'rw2') + f(lat3,
  ↪ model = 'rw2') + f(lat4, model = 'rw2') + f(mu, model = spde) + f(alpha,
  ↪ model = spde, replicate = alpha.rep)
result <- inla(formula, data = dat.inla, control.predictor = list(A = A),
  ↪ control.compute = list(config = TRUE))

```

We now extract results for  $\mu$  and  $\alpha_i$

```

res.mu <- inla.spde2.result(result, 'mu', spde)
res.alpha <- inla.spde2.result(result, 'alpha', spde)

```

and plot the estimated  $\mu$  function

```

mu.est <- res.mu$summary.values$mean
proj <- inla.mesh.projector(mesh, ylim = c(-66, 65), dim = c(360, 132))
image.plot(proj$x, proj$y, inla.mesh.project(proj, field = mu.est))
map("world", add = TRUE, ylim = c(-66, 65))

```

The estimated  $\alpha_i$  function can be plotted similarly.

To find level-zero contour avoiding function of  $\alpha_i$ , we do

```

res.exc <- excursions.inla(result, name = 'alpha', method = 'QC', u = 0, type
  ↪ = '!=')

```

The result can be plotted by

```

alpha.exc <- res.exc$F
for(i in 1:(ng-1)){
  obj.a <- inla.mesh.project(proj, field = alpha.exc[1:n.mesh + (i-1)*n.mesh])
  image.plot(proj$x, proj$y, obj.a, zlim = c(0,1))
  map("world", add = TRUE, ylim = c(-66,65), col = 'white')}

```

To find the regions that have significant seasonal pattern, we need to threshold the avoiding function at 95% level:

```

tmp <- rep(0, length(alpha.exc))
tmp[alpha.exc >= .95] <- alpha.exc[alpha.exc >= .95]
for(i in 1:(ng-1)){
  obj.a <- inla.mesh.project(proj, field=tmp[1:n.mesh+(i-1)*n.mesh])
  image.plot(proj$x, proj$y, obj.a, zlim = c(0.95,1))
  map("world", add = TRUE, ylim = c(-66,65))}

```

The resulting maps show the regions where the main effect  $\alpha_i \neq 0$  with at least 0.95 joint proba-

bility.

## 2 Regional Climate Model Experiment

In climate research field, the regional climate models (RCMs) are used to model the evolution of the climate system over a limited area, addressing smaller spatial regions than global climate models (GCMs) do (Kaufman and Sain, 2010). Due to their limited area, the RCMs require boundary conditions, which are often provided by the output of GCMs. Climate scientists are interested in how much variability in the RCM output is attributable to the RCM itself, and how much is due simply to large-scale boundary conditions provided by the GCM. The data come from the PRUDENCE project (Christensen et al., 2002), where the factors of RCM model choice and GCM boundary conditions are crossed in an experiment involving regional models over Europe from various climate research centers. We examine the control runs (1961-1990) for two RCMs crossed with two GCMs, over the United Kingdom and Ireland. The two RCMs are HIRHAM and RCAO while the two GCMs are ECHAM4 and HadAm3H. Details regarding all the models and references concerning their development can be found at <http://prudence.dmi.dk/>.

Following Kaufman and Sain (2010), we employ a decomposition of the temperature mean response into the effect of RCM, effect of GCM, and their interaction. Also, a linear effect is used to account for an increasing trend in the data for both models. However, what is different from Kaufman and Sain’s approach is that we add to the model an independent Gaussian random field to account for the local spatial variations that cannot be captured by the mean functions. Let  $y_{ijk}(\mathbf{x})$  denote the output at location  $\mathbf{x}$  of RCM  $i$  with boundary conditions provided by GCM  $j$  at time  $k$ . Then the following model is considered

$$y_{ijk}(\mathbf{x}) \mid \mu_{ij}, \xi_{ij}, \phi, \tau \sim N(\mu_{ij}(\mathbf{x}) + \xi_{ij} + \phi k, \tau^{-1})$$

$$\mu_{ij}(\mathbf{x}) = \mu(\mathbf{x}) + \alpha_i(\mathbf{x}) + \beta_j(\mathbf{x}) + \gamma_{ij}(\mathbf{x}) \quad (4)$$

for  $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^2$ ,  $i = 1, 2$ ,  $j = 1, 2$  and  $k = 1, \dots, 30$ , where  $\alpha_i(\mathbf{x})$  is the RCM effect,  $\beta_j(\mathbf{x})$  is the GCM effect, and  $\gamma_{ij}(\mathbf{x})$  is the interaction. Since there are only two levels per factor, we let  $i = 1$  represent the RCM HIRHAM,  $i = -1$  the RCM RCAO,  $j = 1$  the GCM ECHAM4, and  $j = -1$  the GCM HadAm3H, and then we have  $\alpha_i = i\alpha$ ,  $\beta_j = j\beta$  and  $\gamma_{ij} = ij\gamma$ . This reparameterization simplifies the model as well as satisfies the constraints for identifiability. We take RW2D priors on the effect functions  $\mu$ ,  $\alpha$ ,  $\beta$  and  $\gamma$ , a diffuse normal prior on the linear effect  $\phi$ , and independent normal priors on  $\xi_{ij}$ , i.e.,  $\xi_{ij} \sim N(0, \tau_\xi^{-1})$  and  $\tau_\xi$  follows a gamma prior.

We now show how to fit model (4) using INLA. Let’s load data and required packages first

```
library(INLA)
library(excursions)
library(fields)
load("summertemp.RData")
```

and then extract the variables we need and put them in an array

```
n <- prod(dim(temp.jja[[1]])[1:2]) # Spatial locations
N <- dim(temp.jja[[1]])[3] # Number of years
nA <- nB <- 2 # Number of levels in each factor
z <- array(NA, dim=c(2, 2, N, n)) # Dimensions: model, driver, year, location
```



```

for (i in 1:2){
  for (j in 1:2){
    index <- (1:length(temp.jja))[experiments == "Control" & rcms == levels(
      ↪ rcms)[i] & gcms == levels(gcms)[j]]
    z[i, j, , ] <- matrix(temp.jja[[index]], nrow = N, ncol = n, byrow = TRUE)
      ↪ - 273.15 # Convert to deg C
  }
}

```

We then create a data frame that is tailored to the functional ANOVA analysis using model (4) based on the array given above

```

ncol <- length(unique(lonlat$lon))
nrow <- length(unique(lonlat$lat))
ntot <- nA*nB*n*N
dat <- matrix(NA, ntot, 4)
y.tmp <- NULL
count <- 0
for(i in 1:2){
  for(j in 1:2){
    for(t in 1:30){
      for(s in 1:n){
        count <- count+1
        dat[count,2] <- i
        dat[count,3] <- j
        dat[count,4] <- t
      }
      tmp <- z[i,j,t,]
      tmp2 <- matrix(tmp, ncol=ncol, byrow=TRUE)
      tmp3 <- as.vector(tmp2[nrow:1,])
      y.tmp <- c(y.tmp, tmp3)
    }
  }
}
dat[,1] <- y.tmp
dat <- data.frame(dat)
names(dat) <- c('Y', 'I', 'J', "T")

```

We also need to make a index vector for the interaction effect

```

IJ <- rep(NA, ntot)
IJ[dat$I==1&dat$J==1] <- 1
IJ[dat$I==1&dat$J==2] <- 2
IJ[dat$I==2&dat$J==1] <- 3
IJ[dat$I==2&dat$J==2] <- 4
dat$IJ <- IJ

```

and weights for each effect function

```

wt.a <- wt.b <- wt.ab <- rep(NA, dim(dat)[1])
wt.a[dat$I==1] <- 1
wt.a[dat$I==2] <- -1
wt.b[dat$J==1] <- 1
wt.b[dat$J==2] <- -1
wt.ab[dat$IJ==1] <- 1
wt.ab[dat$IJ==2] <- -1

```

```
wt.ab[dat$IJ==3] <- -1
wt.ab[dat$IJ==4] <- 1
```

Then, a data list is created for INLA

```
sm <- sa <- sb <- sab <- serr <- rep(1:n, nA*nB*N)
dat.inla <- list(y = dat$y, t = dat$T, sm = sm, sa = sa, sb = sb, sab = sab,
  ↪ serr = serr, wt.a = wt.a, wt.b = wt.b, wt.ab = wt.ab, ab.rep = dat$IJ)
```

and we fit the model as follows

```
formula <- y ~ 1 + t + f(sm, model="rw2d", nrow=nrow, ncol=ncol, scale.model=
  ↪ TRUE) + f(sa, wt.a, model="rw2d", nrow=nrow, ncol=ncol, scale.model=TRUE
  ↪ ) + f(sb, wt.b, model="rw2d", nrow=nrow, ncol=ncol, scale.model=TRUE) +
  ↪ f(sab, wt.ab, model="rw2d", nrow=nrow, ncol=ncol, scale.model=TRUE) + f(
  ↪ serr, model="iid", replicate=ab.rep)
result <- inla(formula, data=dat.inla, control.compute = list(config=TRUE))
```

The estimates of the three effect functions are given by

```
alpha.est <- result$summary.random$sa$mean
beta.est <- result$summary.random$sb$mean
gamma.est <- result$summary.random$sab$mean
```

and we may plot one estimated effect function as follows

```
xx <- sort(unique(lonlat$lon))
yy <- sort(unique(lonlat$lat))
tmp <- inla.vector2matrix(alpha.est, nrow=nrow)
obj.a <- list(x=xx, y=yy, z=t(tmp[nrow:1,]))
image.plot(obj.a, xlab='', ylab='')
map(add = TRUE)
```

The level-zero avoiding functions for the functional effects can be computed

```
res.a <- excursions.inla(result, name='sa', u=0, type='!=', method='QC')
res.b <- excursions.inla(result, name='sb', u=0, type='!=', method='QC')
res.ab <- excursions.inla(result, name='sab', u=0, type='!=', method='QC')
```

and the resulting contour set can be plotted

```
F.a <- res.a$F
tmp <- rep(NA, length(F.a))
tmp[F.a>=.95] <- F.a[F.a>=.95]
tmp2 <- inla.vector2matrix(tmp, nrow=nrow)
obj.a <- list(x=xx, y=yy, z=t(tmp2[nrow:1,]))
image.plot(obj.a, xlab='', ylab='')
map(add=TRUE)
```

The top panel of Figure 1 shows the posterior means of  $\alpha$  (RCM effect),  $\beta$  (GCM effect) and  $\gamma$  (interaction). After accounting for the local spatial noises, both models show large-scale effects with respect to spatial extent, whereas the RCM effect is estimated to be more localized in Kaufman and Sain (2010). The effect of RCM has relatively small magnitude for most locations, and the RCM is warmer in the Irish Sea and along the east coast, but cooler near the west coast. The GCM effect, however, imposes a little larger magnitude, especially yielding a significant variability over the North Sea. There seems to be very little interaction between the choice of RCM and the choice of GCM with its boundary conditions.

The bottom panels of Figure 1 show the level-zero contour avoiding functions thresholded at

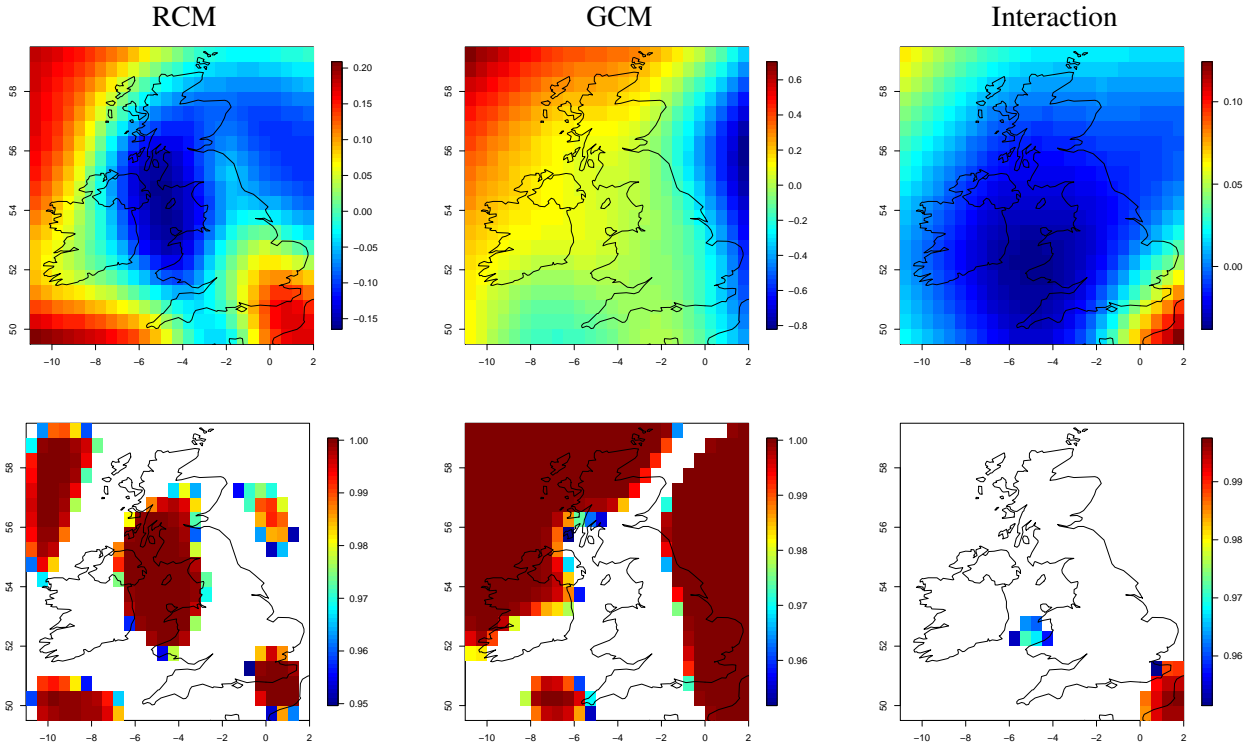


Figure 1: Posterior means (top row) and 95% level-zero contour avoiding sets (bottom row) for the effect of RCM (left column), the effect of GCM (middle column), and their interaction (right column). Due to the  $\pm 1$  coding, the difference between levels are twice this values.

0.95 for the RCM effect, GCM effect and their interaction. The region where these functions are above 0.95 is the largest region where we can say that with probability 0.95 the effect function is jointly different from zero. As we can see, the effect of regional model is statistically significant at the locations over the Irish Sea, along with some locations on the seas outside UK. The significant effect of global model is mainly at the locations over the North Sea, Atlantic Ocean, as well as a few locations on the Celtic Sea. The interaction effect is not significant except for a few locations in the English Channel and along the coast of Wales.

### 3 Additional Figures

In this section we present a few figures, which are referred to in the main manuscript.

## References

- Christensen, J. H., T. R. Carter, and F. Giorgi (2002). Prudence employs new methods to assess european climate change. *EOS, Transactions American Geophysical Union* 83(13), 147–147.
- Kaufman, C. G. and S. R. Sain (2010). Bayesian functional ANOVA modeling using Gaussian process prior distributions. *Bayesian Analysis* 5(1), 123–150.

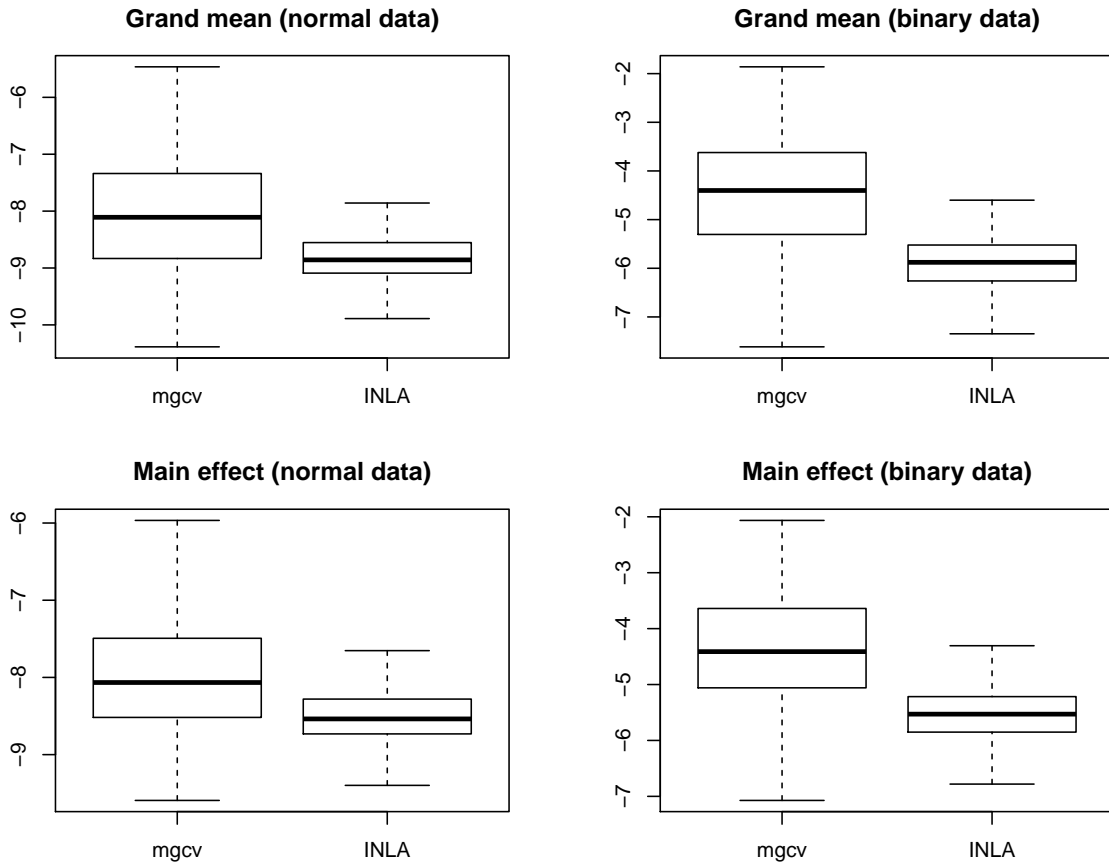


Figure 2: Simulation results for Scenario I (smooth curves):  $\log(\text{MSE})$  of  $\mu(x)$  (grand mean) and  $\alpha_1(x)$  (main effect) given by mgcv and INLA methods for normal and binary data.

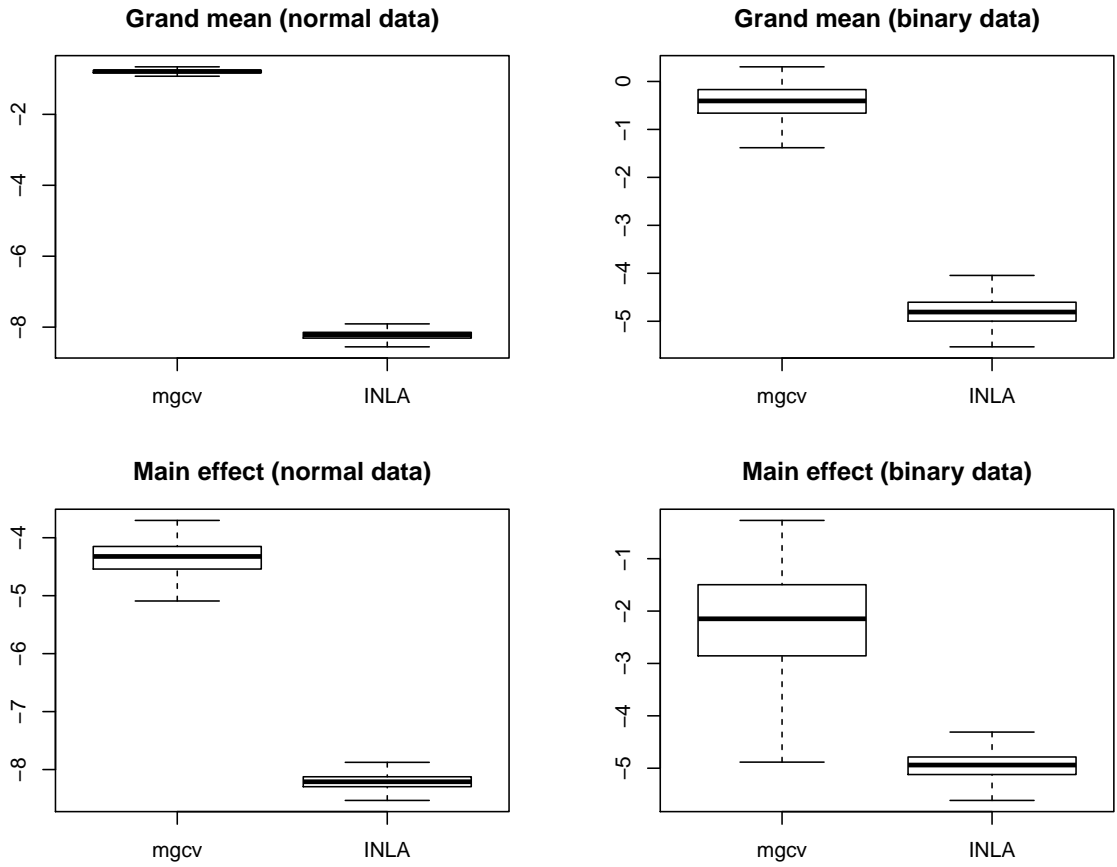
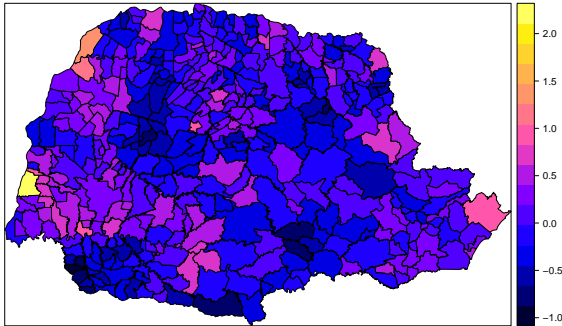
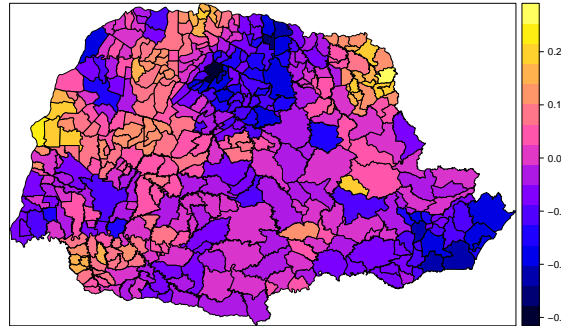


Figure 3: Simulation results for Scenario II (2D functions):  $\log(\text{MSE})$  of  $\mu(x)$  (grand mean) and  $\alpha_1(x)$  (main effect) given by mgcv and INLA methods for normal and binary data.

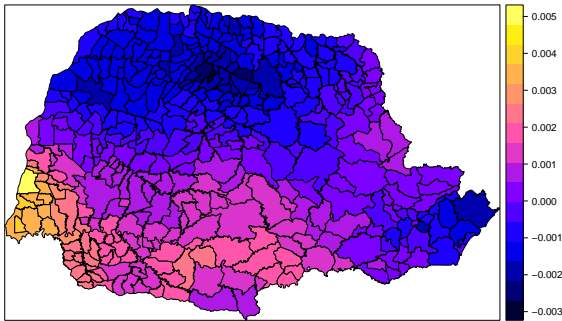
(a)



(c)



(b)



(d)

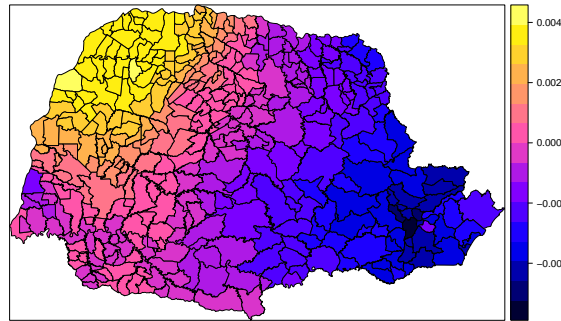


Figure 4: Apgar scores of newborns: Posterior means of (a) grand mean function; (b) main effect function of consultation; (c) main effect function of mother education level; (d) interaction function.

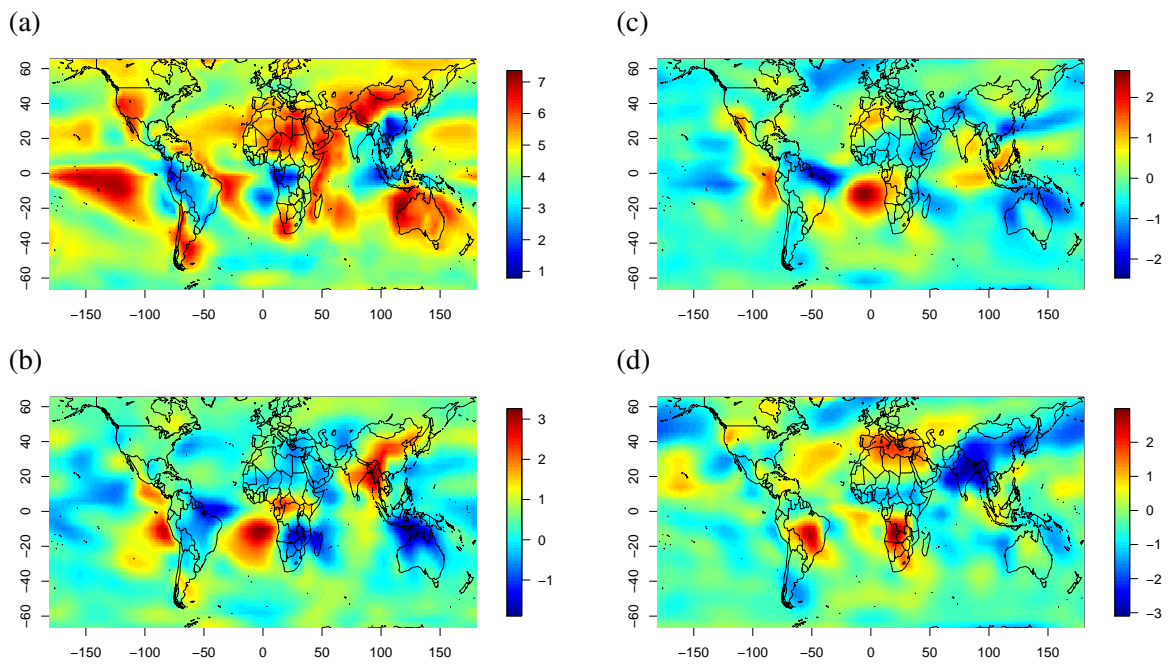


Figure 5: Global solar radiation: Posterior means of (a) grand mean function (fall season), and seasonal effect functions for (b) winter, (c) spring and (d) summer.