Hierarchical Modeling for Large Multivariate Datasets in R

Andrew O. Finley and Sudipto Banerjee

October 11, 2012

1 Data preparation and initial exploration

We make use of several libraries in the following example session, including:

- `library(spBayes)`
- `library(fields)`
- `library(geoR)`
- `library(lattice)`
- `library(MBA)`
- `library(maptools)`
- `library(rgdal)`
- `library(cluster)`
- `library(sp)`

We again make use of the Bartlett Experimental Forest (BEF) dataset. This dataset holds 1991 and 2002 forest inventory data for 437 plots. Variables include species specific basal area and total tree biomass; inventory plot coordinates; slope; elevation; and tasseled cap brightness (TC1), greenness (TC2), and wetness (TC3) components from spring, summer, and fall 2002 Landsat images. Total tree biomass is the sum of bole, branches, and foliage biomass. These quantities are strongly associated, suggesting that we could build a richer model by explicitly accounting for spatial association among the $q = 3$ response variables. Our objective is to obtain an estimate, with an associated measure of uncertainty, of bole, branches, and foliage biomass as a continuous surface over the domain.

We begin by removing non-forest inventory plots, converting biomass measurements from kilograms per hectare to the log of metric tons per hectare, and taking a look at plot locations across the forest.

2 Spatial data visualization

```
> data(BEF.dat)
> BEF.dat <- BEF.dat[BEF.dat$ALLBIO02_KGH > 0, ]
> bio <- BEF.dat$ALLBIO02_KGH * 0.001
> log.bio <- as.matrix(log(0.001 * BEF.dat[, c("BOLE02_KGH",
```

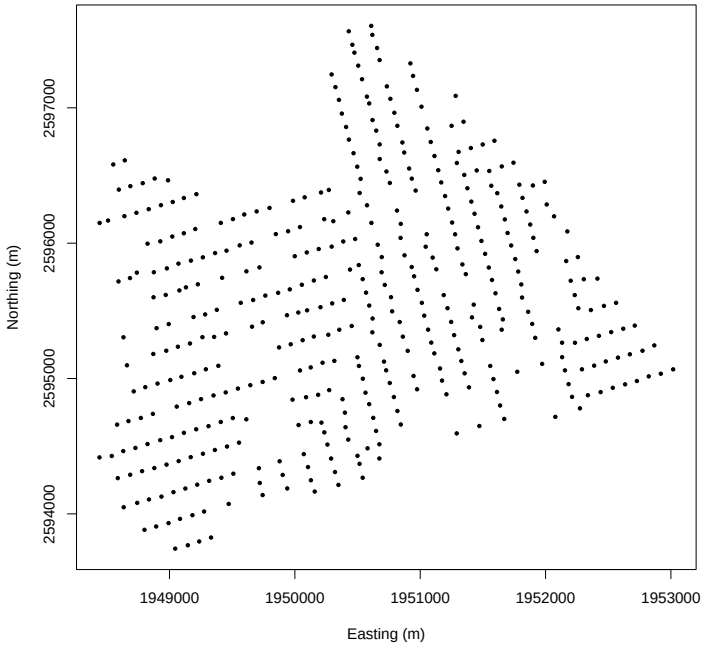

Figure 1: Forest inventory plot locations across the BEF.

```
+     "BRANCH02_KGH", "FOLIAGE02_KGH")]))
> colnames(log.bio) <- c("log.bole.mt", "log.branch.mt",
+     "log.foliage.mt")
> coords <- as.matrix(BEF.dat[, c("XUTM", "YUTM")])
> plot(coords, pch = 19, cex = 0.5, xlab = "Easting (m)",
+     ylab = "Northing (m)")
> cov(log.bio)
```

```
               log.bole.mt log.branch.mt log.foliage.mt
log.bole.mt     0.12661062     0.1214054     0.09020484
log.branch.mt   0.12140543     0.1501818     0.08910730
log.foliage.mt  0.09020484     0.0891073     0.12114077
```

We hope to improve prediction by using elevation, slope, and variables derived from a summer date of 30×30 m resolution Landsat ETM+ satellite imagery. These predictor variables are included in the `BEF.dat` dataset. In the code block below we regress biomass onto the set of predictor variables.

```
> lm.bole <- lm(log.bio[, "log.bole.mt"] ~ ELEV + SLOPE +
+     SUM_02_TC1 + SUM_02_TC2 + SUM_02_TC3, data = BEF.dat)
```

```
> lm.branch <- lm(log.bio[, "log.branch.mt"] ~ ELEV +
+     SLOPE + SUM_02_TC1 + SUM_02_TC2 + SUM_02_TC3, data = BEF.dat)
> lm.foliage <- lm(log.bio[, "log.foliage.mt"] ~ ELEV +
+     SLOPE + SUM_02_TC1 + SUM_02_TC2 + SUM_02_TC3, data = BEF.dat)
> summary(lm.bole)

Call:
lm(formula = log.bio[, "log.bole.mt"] ~ ELEV + SLOPE + SUM_02_TC1 +
    SUM_02_TC2 + SUM_02_TC3, data = BEF.dat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.83144 -0.14015  0.03244  0.21056  0.68026

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.5803241  0.6921018   0.838 0.402242
ELEV         0.0006252  0.0001953   3.201 0.001477 **
SLOPE       -0.0115190  0.0030341  -3.796 0.000169 ***
SUM_02_TC1   0.0080348  0.0055347   1.452 0.147353
SUM_02_TC2   0.0074467  0.0036315   2.051 0.040944 *
SUM_02_TC3   0.0236966  0.0050522   4.690 3.72e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3217 on 409 degrees of freedom
Multiple R-squared: 0.1926,        Adjusted R-squared: 0.1828
F-statistic: 19.52 on 5 and 409 DF,  p-value: < 2.2e-16

> summary(lm.branch)

Call:
lm(formula = log.bio[, "log.branch.mt"] ~ ELEV + SLOPE + SUM_02_TC1 +
    SUM_02_TC2 + SUM_02_TC3, data = BEF.dat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.71109 -0.15654  0.05157  0.20508  0.72509

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.5211260  0.7432739  -2.047 0.041344 *
ELEV         0.0007123  0.0002098   3.396 0.000752 ***
SLOPE       -0.0065855  0.0032585  -2.021 0.043925 *
SUM_02_TC1   0.0177030  0.0059440   2.978 0.003071 **
SUM_02_TC2   0.0044277  0.0039000   1.135 0.256912
SUM_02_TC3   0.0254566  0.0054257   4.692  3.7e-06 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3455 on 409 degrees of freedom
Multiple R-squared: 0.215,        Adjusted R-squared: 0.2054
F-statistic:  22.4 on 5 and 409 DF,  p-value: < 2.2e-16
```

```
> summary(lm.foliage)

Call:
lm(formula = log.bio[, "log.foliage.mt"] ~ ELEV + SLOPE + SUM_02_TC1 +
    SUM_02_TC2 + SUM_02_TC3, data = BEF.dat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.44703 -0.17101  0.01492  0.20811  0.69482

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.1509140  0.6820598  -3.154  0.00173 **
ELEV         0.0005005  0.0001925   2.600  0.00966 **
SLOPE       -0.0070925  0.0029901  -2.372  0.01815 *
SUM_02_TC1   0.0038996  0.0054544   0.715  0.47506
SUM_02_TC2   0.0028403  0.0035788   0.794  0.42786
SUM_02_TC3   0.0310666  0.0049788   6.240  1.1e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.317 on 409 degrees of freedom
Multiple R-squared: 0.1805,        Adjusted R-squared: 0.1705
F-statistic: 18.01 on 5 and 409 DF,  p-value: 3.701e-16
```

3 Fitting the spatial regression model

Given a well distributed sample array, we can either place knots on a grid or use one of several clustering algorithms illustrated in the code block below and plotted in Figure 2.

```
> m <- 50
> km.knots <- kmeans(coords, m)$centers
> cl.knots <- clara(coords, m)$medoids
> cd.knots <- cover.design(coords, nd = m)$design
> plot(coords, pch = 19, cex = 0.5, xlab = "Easting (m)",
+     ylab = "Northing (m)")
> points(km.knots, pch = 5, cex = 1, col = "blue")
> points(cl.knots, pch = 6, cex = 1, col = "green")
```

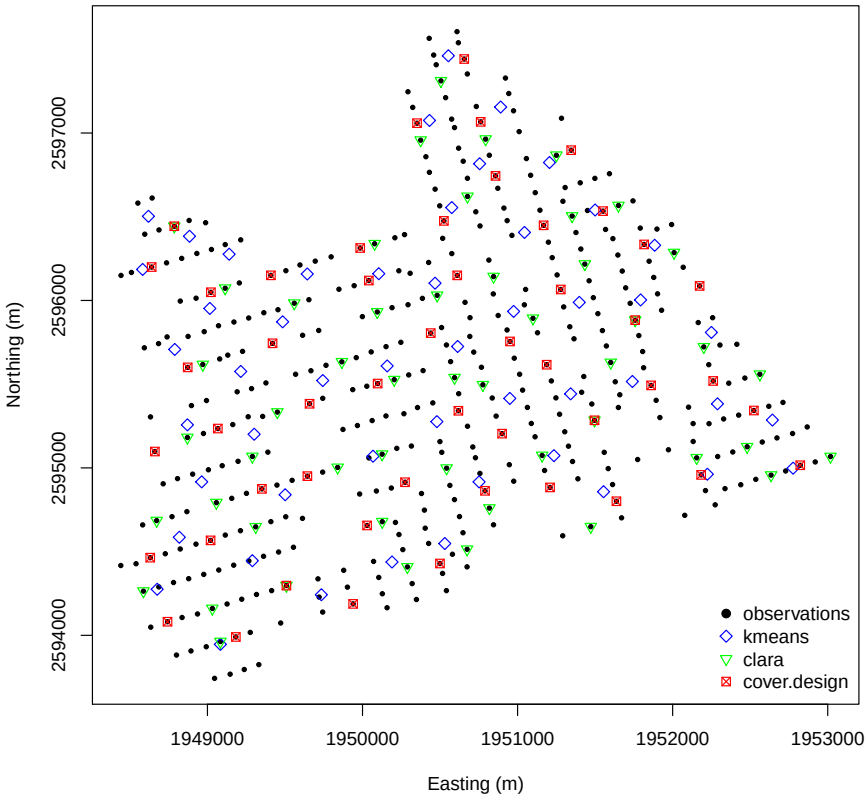

Figure 2: Options for choosing knot locations.

```
> points(cd.knots, pch = 7, cex = 1, col = "red")
> legend("bottomright", cex = 1, pch = c(19, 5, 6, 7),
+     bty = "n", col = c("black", "blue", "green", "red"),
+     legend = c("observations", "kmeans", "clara", "cover.design"))
```

Given the knots we can now fit the predictive process model. In the code block below we call `spMvLM` and choose to use the k-means based knots and the bias-adjusted predictive process (i.e., `modified.pp=TRUE`). Again, for brevity, we have stored the samples from a previous run which took $\sim$45 minutes to collect 25,000 samples.

```
> q <- 3
> A.starting <- diag(0.2, q)[lower.tri(diag(1, q), TRUE)]
> L.starting <- diag(0.2, q)[lower.tri(diag(1, q), TRUE)]
```

```
> A.tuning <- rep(0.01, length(L.starting))
> L.tuning <- rep(0.01, length(L.starting))
> n.samples <- 10
> model <- list(log.bio[, "log.bole.mt"] ~ ELEV + SLOPE +
+     SUM_02_TC1 + SUM_02_TC2 + SUM_02_TC3, log.bio[,
+     "log.branch.mt"] ~ ELEV + SLOPE + SUM_02_TC1 +
+     SUM_02_TC2 + SUM_02_TC3, log.bio[, "log.foliage.mt"] ~
+     ELEV + SLOPE + SUM_02_TC1 + SUM_02_TC2 + SUM_02_TC3)
> bef.spMvLM <- spMvLM(model, coords = coords, knots = km.knots,
+     data = BEF.dat, starting = list(phi = rep(3/500,
+         q), A = A.starting, L = L.starting), sp.tuning = list(phi = rep(0.3,
+         q), A = A.tuning, L = L.tuning), priors = list(phi.Unif = rep(c(3/2500,
+         3/100), q), modified.pp = TRUE, K.IW = list(q +
+         1, diag(0.1, q)), Psi.IW = list(q + 1, diag(0.1,
+         q))), cov.model = "exponential", n.samples = n.samples,
+     sub.samples = c(1, n.samples, 1), verbose = TRUE,
+     n.report = 100)

----------------------------------------
        General model description
----------------------------------------
Model fit with 415 observations.

Number of covariates 18 (including intercept if specified).

Using the exponential spatial correlation model.

Using modified predictive process with 50 knots.

Number of MCMC samples 10.

Priors and hyperpriors:
        beta flat.

        K IW hyperpriors df=4.00000
                0.100        0.000        0.000
        0.000        0.100        0.000
        0.000        0.000        0.100

        Psi IW hyperpriors df=4.00000
                0.100        0.000        0.000
        0.000        0.100        0.000
        0.000        0.000        0.100

        phi Unif hyperpriors
        (0.00120, 0.03000) (0.00120, 0.03000) (0.00120, 0.03000)
```

```
Metropolis tuning values:
        A tuning:
                0.010        0.010        0.010        0.010        0.010        0.010

        L tuning:
                0.010        0.010        0.010        0.010        0.010        0.010

        phi tuning
                0.300        0.300        0.300

Metropolis starting values:
        A starting
        0.200        0.000        0.000
        0.000        0.200        0.000
        0.000        0.000        0.200

        L starting
        0.200        0.000        0.000
        0.000        0.200        0.000
        0.000        0.000        0.200

        phi starting
                0.006        0.006        0.006

-------------------------------------------------
                Sampling
-------------------------------------------------
Sampled: 10 of 10, 100.00%
-------------------------------------------------

> load(file = "R-data/bef.spMvLM")
> p.samples <- bef.spMvLM$p.samples
> p.samples[, paste("phi_", 1:q, sep = "")] <- 3/p.samples[,
+     paste("phi_", 1:q, sep = "")]
> colnames(p.samples)[colnames(p.samples) == paste("phi_",
+     1:q, sep = "")] <- paste("Eff. range ", 1:q, sep = "")
> round(summary(mcmc(p.samples))$quantiles[, c(1, 3,
+     5)], 3)

                    2.5%      50%    97.5%
(Intercept).mod1  -0.987    0.511    2.059
ELEV.mod1          0.000    0.001    0.001
SLOPE.mod1        -0.018   -0.011   -0.004
SUM_02_TC1.mod1   -0.003    0.009    0.023
SUM_02_TC2.mod1   -0.001    0.007    0.015
```

```
SUM_02_TC3.mod1    0.012    0.024    0.036
(Intercept).mod2  -3.338   -1.584    0.208
ELEV.mod2          0.000    0.001    0.002
SLOPE.mod2        -0.018   -0.010   -0.003
SUM_02_TC1.mod2    0.004    0.018    0.033
SUM_02_TC2.mod2   -0.006    0.003    0.012
SUM_02_TC3.mod2    0.014    0.027    0.040
(Intercept).mod3  -5.143   -3.444   -1.784
ELEV.mod3          0.000    0.000    0.001
SLOPE.mod3        -0.012   -0.005    0.003
SUM_02_TC1.mod3    0.001    0.015    0.030
SUM_02_TC2.mod3   -0.014   -0.005    0.004
SUM_02_TC3.mod3    0.030    0.044    0.056
K[1,1]             0.011    0.022    0.043
K[2,1]             0.004    0.016    0.039
K[3,1]            -0.001    0.012    0.034
K[2,2]             0.015    0.030    0.066
K[3,2]            -0.004    0.012    0.038
K[3,3]             0.013    0.035    0.067
Psi[1,1]           0.060    0.081    0.104
Psi[2,1]           0.057    0.079    0.100
Psi[3,1]           0.041    0.063    0.080
Psi[2,2]           0.065    0.094    0.116
Psi[3,2]           0.045    0.067    0.085
Psi[3,3]           0.049    0.071    0.092
Eff. range 1     112.760  489.792 1566.887
Eff. range 2     841.387 1734.366 2412.362
Eff. range 3     597.433 1095.524 2252.878
```

Next we construct plots of the mean of the fitted values' posterior distribution.

```
> X <- bef.spMvLM$X
> beta <- t(p.samples[, 1:bef.spMvLM$p])
> w <- bef.spMvLM$sp.effects
> Y <- rowMeans(X %*% beta + w)
> bole <- Y[seq(1, length(Y), q)]
> branch <- Y[seq(2, length(Y), q)]
> foliage <- Y[seq(3, length(Y), q)]
> res <- 100
> par(mfrow = c(2, 2))
> surf <- mba.surf(cbind(coords, bole), no.X = res, no.Y = res,
+     extend = FALSE)$xyz.est
> image.plot(surf, main = "Bole fitted values")
> points(coords)
> surf <- mba.surf(cbind(coords, branch), no.X = res,
```

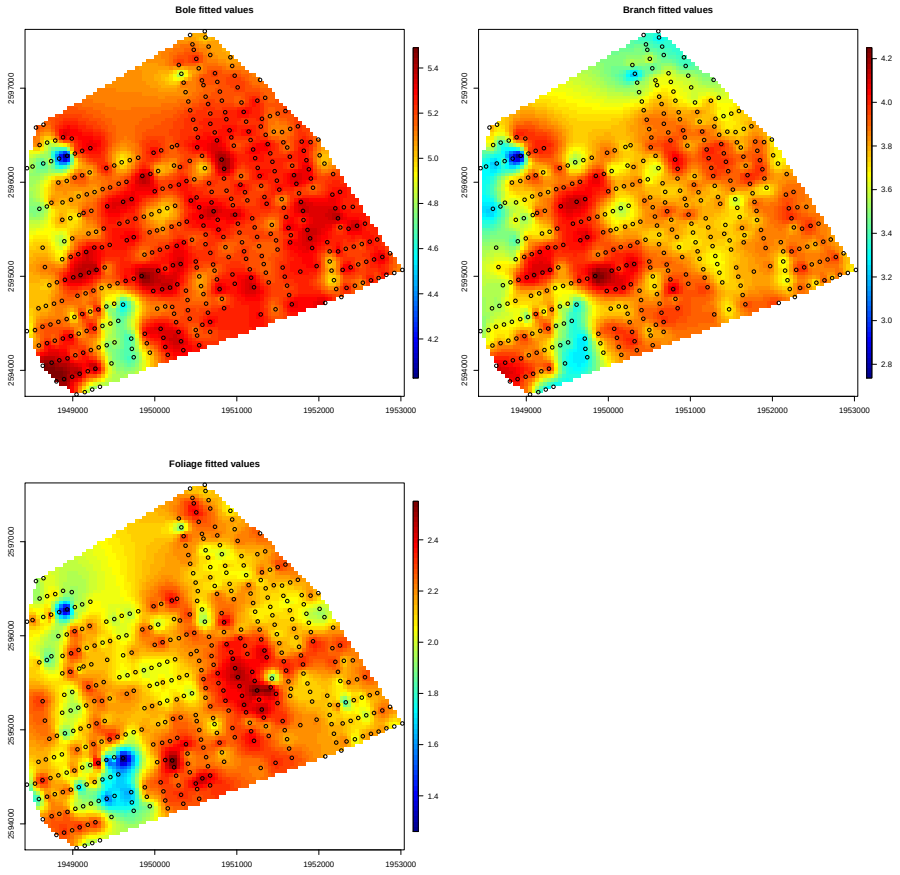

Figure 3: Mean of the fitted values' posterior distribution.

```
+     no.Y = res, extend = FALSE)$xyz.est
> image.plot(surf, main = "Branch fitted values")
> points(coords)
> surf <- mba.surf(cbind(coords, foliage), no.X = res,
+     no.Y = res, extend = FALSE)$xyz.est
> image.plot(surf, main = "Foliage fitted values")
> points(coords)
```

4 Prediction

Given the samples from the parameters' posterior distribution we can now turn to prediction. Using the `spMvLM` object and predictor variables from *new* loca-

tions, the function `spPredict` allows us to sample from the posterior predictive distribution of every pixel across the BEF. We are only interested in predictions within the BEF; however, the predictor variable grid extends well beyond the BEF bounds. Therefore, we would like to *clip* the predictor grid to the BEF bounding polygon. The code block below makes use of the `readShapePoly` function from the **maptools** package and `readGDAL` function from the **rgdal** package to read the bounding polygon and predictor variable grid stack, respectively. We then construct the prediction design matrix for the entire grid extent. Then extract the coordinates of the BEF bounding polygon vertices and use the `pointsInPoly` **spBayes** function to obtain the desired subset of the prediction design matrix and associated prediction coordinates (i.e., pixel centroids). Finally, the `spPredict` function is called and posterior predictive samples are stored in `bef.bio.pred`. Again, we are using samples from a previous `spPredict` run that was based on post burn-in and thinned posterior samples (i.e., note we specify `start=1` and `start=2` only to save computing time in this illustration). `spPredict` took ∼60 minutes to return the posterior predictive samples.

```
> x.range <- range(coords[, 1])
> y.range <- range(coords[, 2])
> BEF.shp <- readShapePoly("BEF-data/BEF_bound.shp")
> BEF.poly <- as.matrix(BEF.shp@polygons[[1]]@Polygons[[1]]@coords)
> BEF.grids <- readGDAL("BEF-data/dem_slope_lolosptc_clip_60.img")

BEF-data/dem_slope_lolosptc_clip_60.img has GDAL driver GTiff
and has 81 rows and 81 columns

> pred.covars <- cbind(BEF.grids[["band1"]], BEF.grids[["band2"]],
+     BEF.grids[["band3"]], BEF.grids[["band4"]], BEF.grids[["band5"]])
> pred.covars <- cbind(rep(1, nrow(pred.covars)), pred.covars)
> pred.coords <- SpatialPoints(BEF.grids)@coords
> pred.covars <- pred.covars[pointsInPoly(BEF.poly, pred.coords),
+     ]
> pred.coords <- pred.coords[pointsInPoly(BEF.poly, pred.coords),
+     ]
> pred.X <- mkMvX(list(pred.covars, pred.covars, pred.covars))
> bef.bio.pred <- spPredict(bef.spMvLM, start = 1, end = 2,
+     pred.coords = pred.coords, pred.covars = pred.X,
+     verbose = TRUE)

-------------------------------------------------
               Starting prediction
-------------------------------------------------

> load(file = "R-data/bef.bio.pred")
```

With access to each pixel's posterior predictive distribution we can map any summary statistics of interest. In Figure 4 we compare the log metric tons of biomass interpolated over the observed plots to that of the pixel-level prediction.

```
> y.pred.mu <- apply(bef.bio.pred$y.pred, 1, mean)
> y.pred.sd <- apply(bef.bio.pred$y.pred, 1, sd)
> bole.pred.mu <- y.pred.mu[seq(1, length(y.pred.mu),
+     q)]
> branch.pred.mu <- y.pred.mu[seq(2, length(y.pred.mu),
+     q)]
> foliage.pred.mu <- y.pred.mu[seq(3, length(y.pred.mu),
+     q)]
> bole.pred.sd <- y.pred.sd[seq(1, length(y.pred.sd),
+     q)]
> branch.pred.sd <- y.pred.sd[seq(2, length(y.pred.sd),
+     q)]
> foliage.pred.sd <- y.pred.sd[seq(3, length(y.pred.sd),
+     q)]
> bio.pred.grid <- as.data.frame(list(x = pred.coords[,
+     1], y = pred.coords[, 2], bole.mu = bole.pred.mu,
+     branch.mu = branch.pred.mu, foliage.mu = foliage.pred.mu,
+     bole.sd = bole.pred.sd, branch.sd = branch.pred.sd,
+     foliage.sd = foliage.pred.sd))
> coordinates(bio.pred.grid) <- c("x", "y")
> gridded(bio.pred.grid) <- TRUE
> toImage <- function(x) {
+     as.image.SpatialGridDataFrame(x)
+ }
> res <- 100
> par(mfrow = c(3, 2))
> surf <- mba.surf(cbind(coords, log.bio[, "log.bole.mt"]),
+     no.X = res, no.Y = res, extend = FALSE)$xyz.est
> z.lim <- range(surf[["z"]], na.rm = TRUE)
> image.plot(surf, xaxs = "r", yaxs = "r", main = "Observed bole biomass")
> image.plot(toImage(bio.pred.grid["bole.mu"]), zlim = z.lim,
+     xaxs = "r", yaxs = "r", main = "Predicted bole biomass")
> surf <- mba.surf(cbind(coords, log.bio[, "log.branch.mt"]),
+     no.X = res, no.Y = res, extend = FALSE)$xyz.est
> z.lim <- range(surf[["z"]], na.rm = TRUE)
> image.plot(surf, xaxs = "r", yaxs = "r", main = "Observed branch biomass")
> image.plot(toImage(bio.pred.grid["branch.mu"]), zlim = z.lim,
+     xaxs = "r", yaxs = "r", main = "Predicted branch biomass")
> surf <- mba.surf(cbind(coords, log.bio[, "log.foliage.mt"]),
+     no.X = res, no.Y = res, extend = FALSE)$xyz.est
> z.lim <- range(surf[["z"]], na.rm = TRUE)
> image.plot(surf, xaxs = "r", yaxs = "r", main = "Observed foliage biomass")
> image.plot(toImage(bio.pred.grid["foliage.mu"]), zlim = z.lim,
+     xaxs = "r", yaxs = "r", main = "Predicted foliage biomass")
```

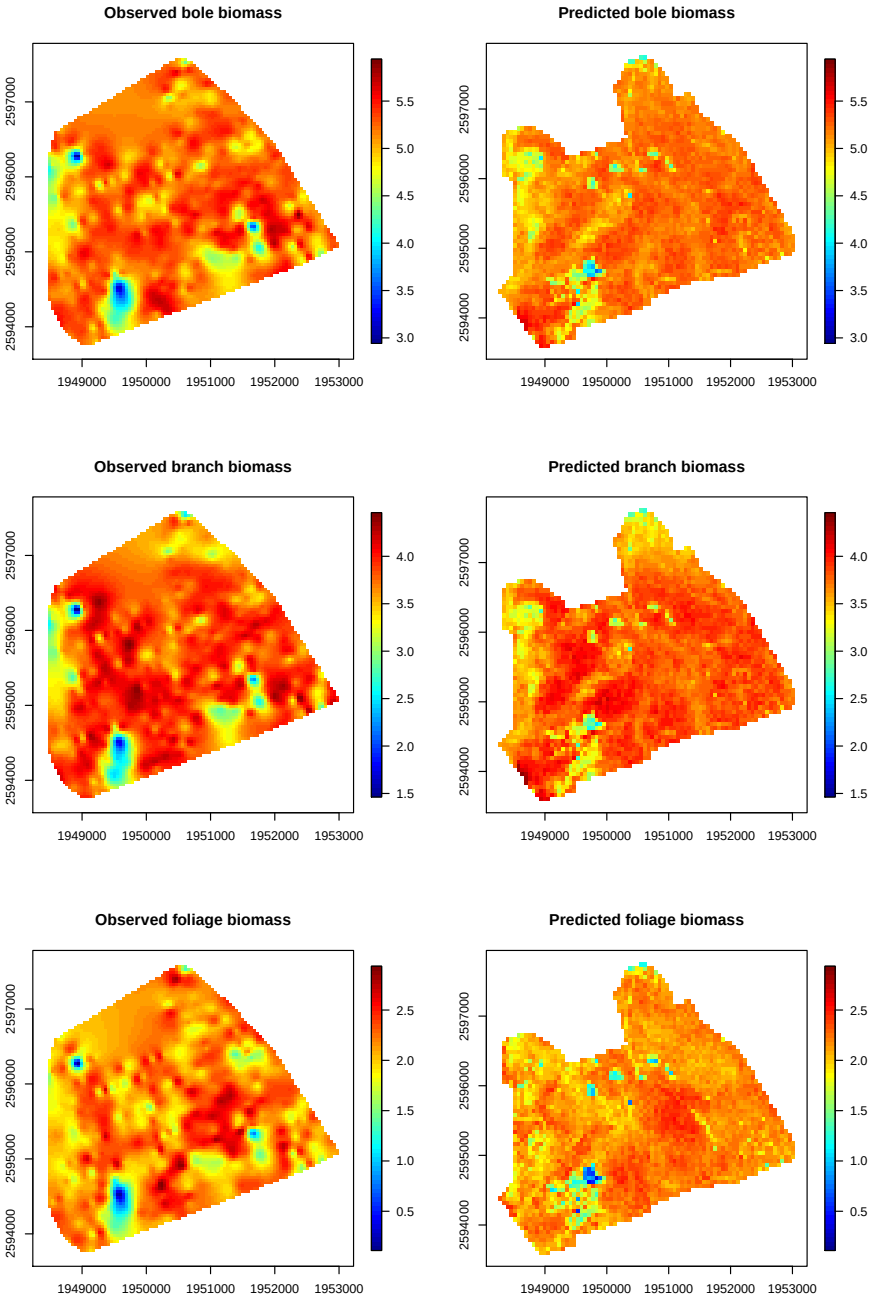

Figure 4: Mean of the posterior predictive distributions.

5 References

Banerjee, S., Carlin, B.P., and Gelfand, A.E. (2004). *Hierarchical Modeling and Analysis for Spatial Data*, Boca Raton, FL: Chapman and Hall/CRC Press.

Bivand, R.B., Pebesma, E.J., and Gómez-Rubio, V. (2008). *Applied Spatial Data Analysis with R*, UseR! Series, Springer.

Diggle, P.J. and Riberio, P.J. (2007). *Model-based Geostatistics*, Series in Statistics, Springer.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (2004). *Bayesian Data Analysis*. Second Edition. Boca Raton, FL: Chapman and Hall/CRC Press.

Spiegelhalter, D.J., Best, N.G., Carlin, B.P., and van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B* **64**, 583–639.