# SIM-18-0289: Code demonstration of spatiotemporal singal detection using continuous shrinkage priors

## Demonstration of a sample analysis

This document demonstrates how the authors fit the Gaussian model and the spatialtemporal horseshoe model using a simulated data for 10 patients. We follow the same way to generate the data as described in Section 6 (simulation study) of the article. We choose the effect size $\beta_0 = 1$ and proprtion of non-null slopes $\pi_0 = 0.05$.

## Setup

We source the two R functions for the Gaussian and spatiotemporal models. The second part of codes show how we generate the response (i.e. periodontal pocket depth, PPD) under the conditions specified above.

```r
library(MCMCpack)
```

```
## Loading required package: coda
```

```
## Loading required package: MASS
```

```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## ## Copyright (C) 2003-2019 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
## ##
## ## Support provided by the U.S. National Science Foundation
```

```
## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```r
library(mvtnorm)
library(FastGP)

setwd("D:/STHS paper/Code/")
source("STHS_Gaussian_CV_new.R")
source("STHS_CV_new.R")
```

```r
#Generate data
np <- 10        #number of patients
ns <- 84        #number of sites
nv <- 5         #number of visits

#Specifiy the CAR covariance
# ur: upright, ul: upleft
# Location as below
```

```r
# ur1 | ul1
# ur2 | ul2
ur1 <- rep(seq(3,1),7)+rep(seq(0,36,by=6),each=3)
ur2 <- ur1+3

ul1 <- rep(seq(1,3),7)+rep(seq(42,78,by=6),each=3)
ul2 <- ul1+3

A <- matrix(0,ns,ns)
A[cbind(c(ur1,ul1),c(ur2,ul2))] <- 1
A[cbind(c(ur2,ul2),c(ur1,ul1))] <- 1

nghb <- cbind(c(ur1,ul1),c(ur2,ul2))
for (k in 1:nrow(nghb)){
  if (k==1){
    A[cbind(nghb[k,],nghb[k+1,])] <- 1
  } else if (k==nrow(nghb)){
    A[cbind(nghb[k,],nghb[k-1,])] <- 1
  } else {
    A[cbind(nghb[k,],nghb[k+1,])] <- 1
    A[cbind(nghb[k,],nghb[k-1,])] <- 1
  }
}

m   <- rowSums(A)
S   <- chol2inv(chol(diag(m)-.99*A))

pi0   <- .05
beta0 <- 1

Y    <- matrix(0,ns,np*nv)
Beta <- matrix(0,ns,np)

for (i in 1:np){
  set.seed(398*i-7)
  alpha    <- as.vector(rcpp_rmvnorm(1,(.5^2)*S,rep(1,ns)))
  beta     <- sample(c(beta0,0),14,replace=TRUE,prob=c(pi0,1-pi0))
  Beta[,i] <- rep(beta,each=6)
  for (j in 1:nv){
    set.seed(11*i-4*j)
    y_star <- rnorm(ns,alpha+(j-1)*beta,1)
    Y[,((i-1)*nv+j)]<- ifelse(round(y_star,0)>0,yes=round(y_star,0),no=0)
  }
}
```

## Run the model

Next, we run the Gaussian and SHS models considering two low-rank representations of the sample covariance, $L = 13$ and $L = 18$, corresponding to 70% and 80% explained variation in the sample covariance. Then we calculate mean squared error (MSE) of predicted PPD values at patients' last dental visits. It takes around 0.93 and 1.23 minutes to run a simulated dataset with $L = 18$ using a computer equipped with Intel i7-4790 3.6GHz processor and 32GB RAM.

```r
#Set up for model fit
Time      <- rep(0:4,np)
Y4        <- Y
Y4[,seq(5,np*nv,5)] <- NA      #Set the 5th visit for each patient as NA
covY      <- cov(t(Y4),use="complete.obs")
EVdecmp   <- eigen(covY)
Q          <- EVdecmp$vectors
d          <- EVdecmp$values

#Calculate proportion of explained variaiton in the sample coveraince
prop <- cumsum(EVdecmp$values)/sum(EVdecmp$values)
L1    <- min(which(prop>=.7))
L2    <- min(which(prop>=.8))

#Fit the models
fit_Gaussian1 <- sths0_new(Y4, np, Time, Q[,1:L1], d[1:L1], 1:np)
fit_Gaussian2 <- sths0_new(Y4, np, Time, Q[,1:L2], d[1:L2], 1:np)
fit_HS1       <- sths_new(Y4, np, Time, Q[,1:L1], d[1:L1], 1:np)
fit_HS2       <- sths_new(Y4, np, Time, Q[,1:L2], d[1:L2], 1:np)


#Summarize the results
pred.mse <- matrix(c(mean((fit_Gaussian1$pred.mn-Y[,seq(5,np*nv,5)])^2),
                     mean((fit_Gaussian2$pred.mn-Y[,seq(5,np*nv,5)])^2),
                     mean((fit_HS1$pred.mn-Y[,seq(5,np*nv,5)])^2),
                     mean((fit_HS2$pred.mn-Y[,seq(5,np*nv,5)])^2)),
                   2,2,byrow=TRUE)
colnames(pred.mse) <- c("L=13","L=18")
rownames(pred.mse) <- c("Gaussian","SHS")
print(pred.mse)
```

```
##                 L=13      L=18
## Gaussian  2.001625  1.791328
## SHS       1.443795  1.399572
```