

GenePair: Statistical Methods for Modeling Spatially-Referenced Paired Genetic Relatedness Data

Trans_Prob_Example

[1] Simulate data from the proposed model:

- Setting the reproducibility seed and initializing packages for data simulation:

```
set.seed(2411)

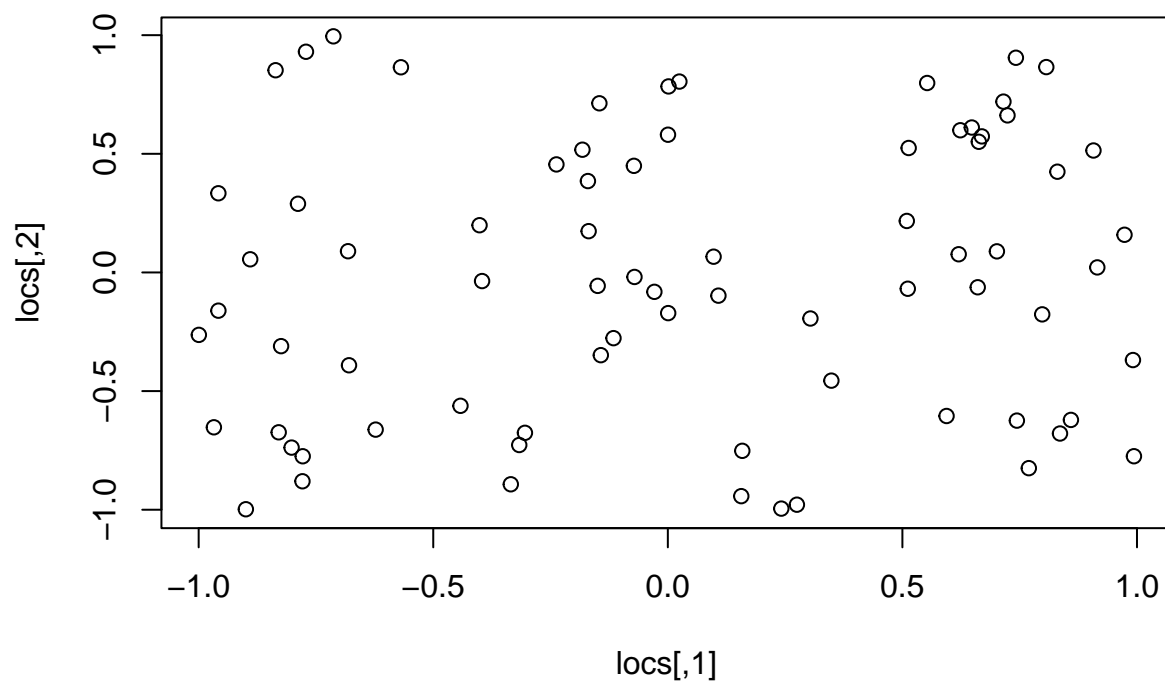
library(GenePair)
library(mnormt) #Multivariate normal distribution
```

- Setting the global data values:

```
n<-75 #Number of individuals
n_star<-0 #Number of paired responses
for(j in 1:n){
  for(k in 1:n){
    if(j != k){
      n_star<-n_star +
        1
    }
  }
}

#Unique locations (one for each individual)
locs<-matrix(runif((2*n),
                  min = -1.00,
                  max = 1.00),
             nrow = n,
             ncol = 2)

plot(locs)
```



```
v<-diag(n)

spatial_dists<-as.matrix(dist(locs,
                              diag = TRUE,
                              upper = TRUE))

#x_pair (intercept included)
x_pair<-matrix(1,
               nrow = n_star,
               ncol = 2)
x<-rnorm(n = n)
counter<-1
for(j in 1:n){
  for(k in 1:n){
    if(j != k){

      x_pair[counter, 2]<-abs(x[j] - x[k])
      counter<-counter +
        1

    }
  }
}
x_pair[,2]<-scale(x_pair[,2])

#x_ind_g, x_ind_r (no intercept)
```

```

x1<-rnorm(n = n)
x2<-rnorm(n = n)
x_ind_g<-
x_ind_r<-matrix(0.00,
                nrow = n_star,
                ncol = 2)

counter<-1
for(j in 1:n){
  for(k in 1:n){
    if(j != k){

      x_ind_g[counter, 1]<-x1[k]
      x_ind_g[counter, 2]<-x2[k]
      x_ind_r[counter, 1]<-x1[j]
      x_ind_r[counter, 2]<-x2[j]
      counter<-counter +
        1

    }
  }
}
x_ind_g<-scale(x_ind_g)
x_ind_r<-scale(x_ind_r)

#z_g, z_r
z_g<-matrix(0,
            nrow = n_star,
            ncol = n)
z_r<-matrix(0,
            nrow = n_star,
            ncol = n)

counter<-1
for(j in 1:n){
  for(k in 1:n){
    if(j != k){

      z_g[counter, k]<-1
      z_r[counter, j]<-1
      counter<-counter +
        1

    }
  }
}

```

- Setting the values for the statistical model parameters:

```

beta_z_true<- c(1.00,
               -0.30)
gamma_z_g_true<-c(2.10,
                 -1.24)
gamma_z_r_true<-c(1.05,
                 0.56)
sigma2_zeta_z_g_true<-0.01

```

```

sigma2_zeta_z_r_true<-0.01

sigma2_epsilon_true<-0.01
beta_w_true<-c(-2.40,
               0.50)
gamma_w_g_true<-c(0.80,
                  -0.35)
gamma_w_r_true<-c(-1.10,
                  -0.75)
sigma2_zeta_w_g_true<-0.01
sigma2_zeta_w_r_true<-0.01

Sigma_true<-chol2inv(chol(rWishart(n = 1,
                                   df = 5,
                                   Sigma = diag(4))[,,1]))

phi_true<-5.00
Sigma_full_true<-Sigma_true%x%exp(-phi_true*spatial_dists)
eta_full_true<-rmnorm(n = 1,
                      mean = rep(0.00,
                                   times = (n*4)),
                      varcov = Sigma_full_true)

theta_z_g_true<-eta_full_true[1:n] +
  rnorm(n = n,
        mean = 0.00,
        sd = sqrt(sigma2_zeta_z_g_true))
theta_z_g_true<-theta_z_g_true -
  mean(theta_z_g_true)

theta_z_r_true<-eta_full_true[(n + 1):(2*n)] +
  rnorm(n = n,
        mean = 0.00,
        sd = sqrt(sigma2_zeta_z_r_true))
theta_z_r_true<-theta_z_r_true -
  mean(theta_z_r_true)

theta_w_g_true<-eta_full_true[(2*n + 1):(3*n)] +
  rnorm(n = n,
        mean = 0.00,
        sd = sqrt(sigma2_zeta_w_g_true))
theta_w_g_true<-theta_w_g_true -
  mean(theta_w_g_true)

theta_w_r_true<-eta_full_true[(3*n + 1):(4*n)] +
  rnorm(n = n,
        mean = 0.00,
        sd = sqrt(sigma2_zeta_w_r_true))
theta_w_r_true<-theta_w_r_true -
  mean(theta_w_r_true)

mu_z<-x_pair%%beta_z_true +
  x_ind_g%%gamma_z_g_true +
  x_ind_r%%gamma_z_r_true +

```

```

z_g%%theta_z_g_true +
z_r%%theta_z_r_true

probs_z<-1.00/(1.00 + exp(-mu_z))
bin<-rbinom(n = n_star,
           size = 1,
           prob = probs_z)

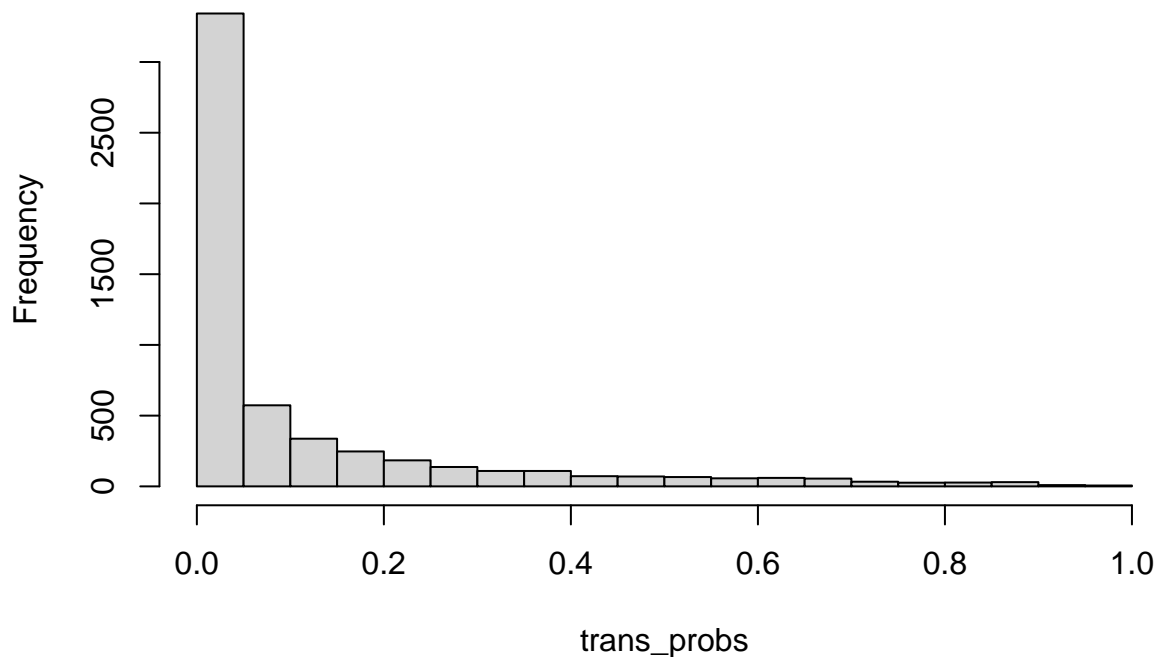
mu_w<-x_pair%%beta_w_true +
      x_ind_g%%gamma_w_g_true +
      x_ind_r%%gamma_w_r_true +
      z_g%%theta_w_g_true +
      z_r%%theta_w_r_true

w<-rnorm(n = n_star,
         mean = mu_w,
         sd = sqrt(sigma2_epsilon_true))
w<-1.00/(1.00 + exp(-w))
trans_probs<-bin*w

hist(trans_probs)

```

Histogram of trans_probs



[2] Apply the Trans_Prob Function within the GenePair Package:

```

results<-Trans_Prob(mcmc_samples = 10000,
                    trans_probs = trans_probs,

```

```

x_pair = x_pair,
x_ind_g = x_ind_g,
x_ind_r = x_ind_r,
z_g = z_g,
z_r = z_r,
spatial_dists = spatial_dists,
v = v,
metrop_var_phi_trans = 0.20)

```

```

## Progress: 10%
## phi Acceptance: 28%
## *****
## Progress: 20%
## phi Acceptance: 28%
## *****
## Progress: 30%
## phi Acceptance: 29%
## *****
## Progress: 40%
## phi Acceptance: 29%
## *****
## Progress: 50%
## phi Acceptance: 29%
## *****
## Progress: 60%
## phi Acceptance: 30%
## *****
## Progress: 70%
## phi Acceptance: 29%
## *****
## Progress: 80%
## phi Acceptance: 29%
## *****
## Progress: 90%
## phi Acceptance: 29%
## *****
## Progress: 100%
## phi Acceptance: 29%
## *****

```

[3] Analyzing Output:

```

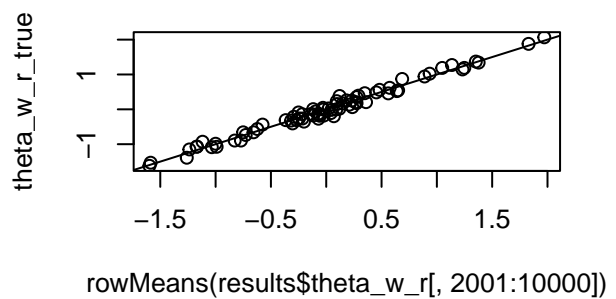
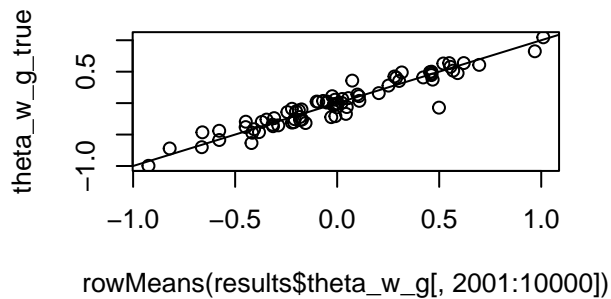
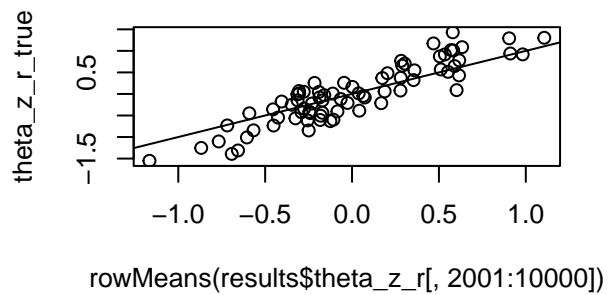
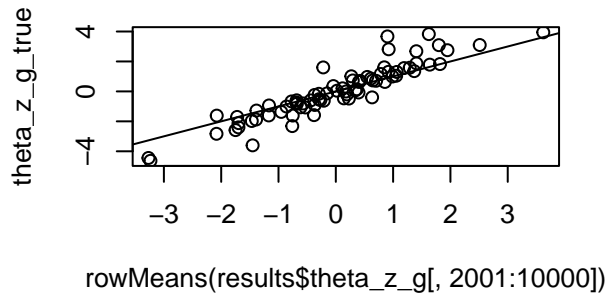
#Compare random effects
par(mfrow = c(2,2))
plot(rowMeans(results$theta_z_g[,2001:10000]),
     theta_z_g_true)
abline(0,1)

plot(rowMeans(results$theta_z_r[,2001:10000]),
     theta_z_r_true)
abline(0,1)

plot(rowMeans(results$theta_w_g[,2001:10000]),
     theta_w_g_true)
abline(0,1)

```

```
plot(rowMeans(results$theta_w_r[,2001:10000]),
     theta_w_r_true)
abline(0,1)
```



```
#Compare \beta
round(rowMeans(results$beta_z[,2001:10000]), 2)

## [1] 0.86 -0.35
round(beta_z_true, 2)

## [1] 1.0 -0.3
round(rowMeans(results$beta_w[,2001:10000]), 2)

## [1] -2.39 0.50
round(beta_w_true, 2)

## [1] -2.4 0.5

#Compare \gamma_z_g
round(rowMeans(results$gamma_z_g[,2001:10000]), 2)

## [1] 1.92 -1.21
round(gamma_z_g_true, 2)

## [1] 2.10 -1.24
```

```

round(rowMeans(results$gamma_z_r[,2001:10000]), 2)

## [1] 0.96 0.59
round(gamma_z_r_true, 2)

## [1] 1.05 0.56
round(rowMeans(results$gamma_w_g[,2001:10000]), 2)

## [1] 0.71 -0.31
round(gamma_w_g_true, 2)

## [1] 0.80 -0.35
round(rowMeans(results$gamma_w_r[,2001:10000]), 2)

## [1] -1.09 -0.65
round(gamma_w_r_true, 2)

## [1] -1.10 -0.75
#Compare \sigma2_epsilon
round(mean(results$sigma2_epsilon[2001:10000]), 2)

## [1] 0.01
sigma2_epsilon_true

## [1] 0.01
#Compare \Sigma
Sigma<-array(0.00,
             dim = c(10000, 4, 4))
for(j in 1:10000){
  Sigma[j,,]<-results$Sigma[[j]]
}
round(colMeans(Sigma[2001:10000,,]), 2)

##      [,1] [,2] [,3] [,4]
## [1,]  1.15 -0.16  0.28 -0.55
## [2,] -0.16  0.17 -0.01  0.02
## [3,]  0.28 -0.01  0.22 -0.18
## [4,] -0.55  0.02 -0.18  0.56
Sigma_true

##      [,1]      [,2]      [,3]      [,4]
## [1,]  3.4165470 -0.73126656  0.3379183 -0.95289423
## [2,] -0.7312666  0.75024506  0.0862960  0.05214396
## [3,]  0.3379183  0.08629600  0.2399986 -0.16795824
## [4,] -0.9528942  0.05214396 -0.1679582  0.51692972

```