semi_simulation

```r
library(MASS)
library(Iso)
```

```
## Iso 0.0-17
```

```r
library(mvtnorm)
```

Model overview

Basic model

Recently Wang and Zhou (2018) proposed a parametric transformation model for biomarker data, of the form

$$f(\mathbf{t}|\mathbf{x},\mathbf{z}) = \sum_{d=0}^{L-1} \pi_d(\mathbf{z}) \prod_{k=1}^{K} J(t_k,\lambda_k) g(H_k(t_k,\lambda_k|\mathbf{x}),$$

where $g(\cdot|\mathbf{x})$ is a known parametric density, $H(\cdot,\cdot)$ is a given transformation, $J(t_k,\lambda_k)$ is the transformation Jacobian, and $\pi_d(\mathbf{z})$ is the mixture proportion, which is specified as a logit form. In their model, the joint density is independent given the covariate $\mathbf{x}$

Our extension

Here our model is more general, the joint density is not independent, and the monotone transformations $H_k(\cdot)$'s are also unknown to be estimated The observed data is $D_n = \{(\mathbf{y}_i,\mathbf{x}_i)\} : i = 1,...,n\}$ from n subjects. $\mathbf{y}_i = (y_{i1},...,y_{ik})^T$ is the observation of the i-th subject with k biomarkers, $\mathbf{x}_i = (x_{i1},...,x_{iq})$ is the corresponding covariates, we assume the $(\mathbf{y}_i,\mathbf{x}_i)$'s are iid.

The goal is to classify each subject to one of the two groups, normal and disabled. For this, we first need to specify a model, then estimate the model parameters and finally, classify the subjects.

The normal model is easy to use but not robust to model assumption. For robustness we propose the following semi-parametric transformation model for the observed data. First, for each margin, we specify the transformation

$$g_j(y_{ij}) = \mathbf{x}_i^T\beta_j + \epsilon_{ij} \qquad (i = 1,2,...,n; j = 1,...,k)(1)$$

where $g_j(\cdot) \in \mathcal{G}$, $\mathcal{G}$ is the collection of bounded monotone non-decreasing functions, and ϵ_{ij} is the noise. We assume $\epsilon_i := (\epsilon_{i1},...,\epsilon_{ik})^T \sim N(\mathbf{0},\Omega)$, $\Omega = (\omega_{ij})_{k\times k}$. Denote $\mathbf{g}(\mathbf{y}_i) = \big(g_1(y_{i1}),...,g_k(y_{ik})\big)^T$ and $\mathbf{x_i} = (x_{i1},...,x_{iq})^T$, $\mathbf{beta}_j = (\beta_{j1},...,\beta_{jq})^T$ and $\mathbf{B}^T = (\beta_1,...,\beta_k)$. Let $\phi(\cdot|\Omega)$ be the density function of the $N(\mathbf{0},\Omega)$ distribution. Next, since the group status of each subject is unknown, the joint model is specified as a mixture

$$f(\mathbf{y}_i|\mathbf{x}_i) = \sum_{d=0}^{1} \pi_d(\mathbf{x}_i)\phi(\mathbf{g}(\mathbf{y}_i) - \mathbf{x}_i^T\mathbf{B}|\Omega)(2)$$

where

$$\pi_0(\mathbf{x}_i|\eta) = 1 - \pi_1(\mathbf{x_i}|\eta) = \frac{\exp(\mathbf{x}_i^T\eta)}{1+\exp(\mathbf{x}_i^T\eta)}$$

is the mixing proportions. $\pi_d(\mathbf{x}_i|\eta)$ is the probability subject i belongs to group d.

Simulation settings:

- Assumption1: joint density is not independent, and $H_k(\cdot) = I_k(\cdot)$, $k = 2$ (2 biomarkers) with $q = 2$ (2 covariates).

We need to estimate $\eta = \{\eta_1, \eta_2, ..., \eta_q\}$, $\theta = \{\eta_{jd},\ \omega_{jr} : j, r = 1, ..., k; d = 0, 1\}$ and $\mathbf{g}(\cdot) = (g_1(\cdot), ..., g_k(\cdot))$. Note that since the ω_{jr}'s are symmetric, we only need to estimate ω_{jr} for $j = 1, ..., k; r \geq j$. Let δ_i be the latent group indicator of the i-th subject, $\delta_i = 0$ if this subject belongs to group 0, and $\delta_i = 1$ otherwise. Let $D_{n,c} = \{(\mathbf{y}_i, x_i, \delta_i) : i = 1, ..., n\}$ be the "complete" data. The log-likelihood of the complete data is

$$\ell(\eta, \theta, \mathbf{g}|D_{n,c}) = \sum_{i=1}^{n} \sum_{d=0}^{1} \Big[I_d(C_i) \Big(\log \pi_d(\mathbf{x}_i|\eta) - \frac{1}{2} \big(\mathbf{g}(\mathbf{y}_i) - \mathbf{x}_i^T \mathbf{B}\big)^T \Omega^{-1} \big(\mathbf{g}(\mathbf{y}_i) - \mathbf{x}_i^T \mathbf{B}\big) \Big)$$

$$-\frac{n}{2} \log |\Omega|.(3)$$

We estimate $(\eta, \theta, \mathbf{g})$ by

$$(\hat{\eta}, \hat{\theta}, \hat{\mathbf{g}}) = \arg \max_{(\eta,\theta,\mathbf{g}) \in (\mathbf{B}, \boldsymbol{\Theta}, \mathcal{G}^k)} \ell(\eta, \theta, \mathbf{g}|D_{n,c}).(4)$$

- Simulation function

```
no_gold_standard_simu<- function(eta.r, theta.r, b.0.r, b.1.r, o.r, D){
  iter = 1
  res = c()
  res[iter] = 1000
  while(res[iter]>0.001){
  print(iter)
  par_old = c(eta.r, theta.r)
  print(iter)
  # if(iter == 1){
    pi1.r = exp(as.matrix(D$X)%*%eta.r)/(1+exp(as.matrix(D$X)%*%eta.r))
    pi0.r = 1-pi1.r

    p0.r = dmvnorm(g.r-as.matrix(D$X)%*%b.0.r,mean = c(0,0), sigma = o.r)
    p1.r = dmvnorm(g.r-as.matrix(D$X)%*%b.1.r,mean = c(0,0), sigma = o.r)
    P.r = (pi1.r*p1.r)/(pi1.r*p1.r+pi0.r*p0.r)
    # res_eta = mean(abs(D$d_prob - P.r))
  # }


  # label.new = ifelse(P.r>1-P.r,1,0)
  # mean.0 = apply((g.r-as.matrix(D$X))[label.new==0,],2,mean)
  # mean.1 = apply((g.r-as.matrix(D$X))[label.new==1,],2,mean)
  #
  # P.r.cat = as.factor(ifelse(P.r>0.5,1,0))
    eta.r<- glm(P.r~.-1, data = D$X,family = binomial)$coef

  sprintf("Estimated eta: %s", eta.r)
  sprintf("True Estimated eta: %s", eta)

  mytheta = function(par){
      o = matrix(0,2,2)
      o[1,1]<-par[1]
      o[2,2]<-par[2]
      o[1,2] <- par[3]*prod(sqrt(diag(o)))
      o[2,1] <- o[1,2]
```

```
    beta1 <- par[4:5]
    beta2 <- par[6:7]
    b0 <- matrix(c(beta1,beta2),2)

    beta1 <- par[8:9]
    beta2 <- par[10:11]
    b1 <- matrix(c(beta1,beta2),2)

    o.det = prod(par[1],par[2])-o[1,2]^2
    o.det = det(o)
    o.inv = matrix(0,2,2)
    o.inv[1,1]<-par[2]/o.det
    o.inv[2,2]<-par[1]/o.det
    o.inv[1,2] <- -o[1,2]/o.det
    o.inv[2,1] <- -o[2,1]/o.det
    # o.inv = solve(o)
    # sum(c(log(P.r))*(-0.5)*c(apply(as.matrix(g.r)-as.matrix(D$X)%*%b1,1,function(x) t(as.matrix(x))
    #     c(1-log(P.r))*(-0.5)*c(pply(as.matrix(g.r)-as.matrix(D$X)%*%b0,1,function(x) t(as.matrix(x)
    #     -N/2*log(o.det)
    sum(
      P.r*-0.5*c(apply(as.matrix(g.r)-as.matrix(D$X)%*%b1,1,function(x) t(as.matrix(x))%*%o.inv%*%as.m
        ,
        (1-P.r)*-0.5*c(apply(as.matrix(g.r)-as.matrix(D$X)%*%b0,1,function(x) t(as.matrix(x))%*%o.inv%
       ,-N/2*log(o.det))
}
result = try(optim(theta.r,mytheta, method = 'BFGS',
                   control= list(fnscale=-1)))


if(inherits(result, "try-error")==TRUE){
  res = res
#   result2.1 = try(pava(diag(as.matrix(D$X)%*%as.matrix(apply(P.r,1, function(x) (1-x)*as.matrix(b.0
#                   +0.5*(diag(as.matrix(D$X)%*%as.matrix(apply(P.r,1, function(x) (1-x)*as.matrix(b.
#                   -g.r[,2])*o.inv[1,2]/o.inv[1,1]))
#
#
#   result2.2 = try(pava(diag(as.matrix(D$X)%*%as.matrix(apply(1-P.r,1, function(x) (1-x)*as.matrix(b
# +0.5*(diag(as.matrix(D$X)%*%as.matrix(apply(P.r,1, function(x) (1-x)*as.matrix(b.0.r[,1],nrow=2))+a
#       -g.r[,1])*o.inv[1,2]/o.inv[2,2]))
#
#   if(inherits(result2.1, "try-error")==TRUE | inherits(result2.2, "try-error")==TRUE){
#     res = res
#   }else{
#     g.r.1 <- matrix(0, nrow = 1000, ncol = 2)
#     g.r.1[,1] <- pava(diag(as.matrix(D$X)%*%as.matrix(apply(P.r,1, function(x) (1-x)*as.matrix(b.0.
#                   +0.5*(diag(as.matrix(D$X)%*%as.matrix(apply(P.r,1, function(x) (1-x)*as.matrix(b.
#                   -g.r[,2])*o.inv[1,2]/o.inv[1,1])
#
#
#     g.r.1[,2] <- pava(diag(as.matrix(D$X)%*%as.matrix(apply(1-P.r,1, function(x) (1-x)*as.matrix(b.
# +0.5*(diag(as.matrix(D$X)%*%as.matrix(apply(P.r,1, function(x) (1-x)*as.matrix(b.0.r[,1],nrow=2))+a
#       -g.r[,1])*o.inv[1,2]/o.inv[2,2])
#   g.r <- g.r.1
```

```
#   }
}else{
  theta.opt = optim(theta.r,mytheta, method = 'BFGS',
                    control= list(fnscale=-1))
  theta.r <- theta.opt$par
theta.conv <- theta.opt$convergence
print(ifelse(theta.conv==0,'success','fail'))
sprintf("Estimated theta: %s, %s and %s ", theta.r[1],theta.r[2], theta.r[3])
sprintf("True theta: %s, %s and %s", O[1,1],O[2,2], rho)

sprintf("Estimated beta0: %s, %s, %s,%s",
        theta.r[4],theta.r[5],theta.r[6],theta.r[7])

sprintf("True beta0: %s, %s, %s,%s",
       B.0[1],B.0[2],B.0[3],B.0[4])

sprintf("Estimated beta1: %s, %s, %s,%s ",
       theta.r[8],theta.r[9],theta.r[10],theta.r[11])

sprintf("True beta1: %s, %s, %s,%s",
       B.1[1],B.1[2],B.1[3],B.1[4])

b.0.r = matrix(theta.r[4:7],2,2)
b.1.r = matrix(theta.r[8:11],2,2)

o.r = matrix(0,2,2)
o.r[1,1]<-theta.r[1]
o.r[2,2]<-theta.r[2]
o.r[1,2] <- theta.r[3]*prod(sqrt(diag(o.r)))
o.r[2,1] <- o.r[1,2]
o.det = prod(diag(o.r))-o.r[1,2]^2
o.inv = matrix(0,2,2)
o.inv[1,1]<-o.r[2,2]/o.det
o.inv[2,2]<-o.r[1,1]/o.det
o.inv[1,2] <- -o.r[2,1]/o.det
o.inv[2,1] <- -o.r[2,1]/o.det

# g.r.1 <- matrix(0, nrow = 1000, ncol = 2)
# g.r.1[,1] <- pava(diag(as.matrix(D$X)%*%as.matrix(apply(P.r,1, function(x) (1-x)*as.matrix(b.0.r[,1
#                    +0.5*(diag(as.matrix(D$X)%*%as.matrix(apply(P.r,1, function(x) (1-x)*as.matrix(b.
#                    -g.r[,2])*o.inv[1,2]/o.inv[1,1])
#
# g.r.1[,2] <-  pava(diag(as.matrix(D$X)%*%as.matrix(apply(1-P.r,1, function(x) (1-x)*as.matrix(b.0.r
# +0.5*(diag(as.matrix(D$X)%*%as.matrix(apply(P.r,1, function(x) (1-x)*as.matrix(b.0.r[,1],nrow=2))+a
#       -g.r[,1])*o.inv[1,2]/o.inv[2,2])
#
# res2 = mean(abs(g.r-g.r.1))
# g.r <- g.r.1
# res[iter] = mean(abs(g.r-D$Y))
 par_new = c(eta.r, theta.r)

 # par_new = c(res_eta, theta.r)
res = append(res, max(abs(par_old-par_new)), after = length(res))
```

```
  # res = append(res,max(res2,abs(par_old-par_new)), after = length(res))
  # g.r <- g.r.1
  print(res[iter])
  iter = iter+1
  }
}



# mean(as.matrix(D$X[P.r<0.5,])%*%B.0 - D$Y[P.r<0.5,])


print(paste("Converge in %s iteration:", iter))
print("Converge covariance matrix: ")
print(o.r)

print("Original covariance matrix: ")
print(O)
return(list(eta.r = eta.r, theta.r = theta.r))
}
```

- Basic Settings: 2 biomarkers, 2 covairates and 2 classes of diseases(0-no disease, 1-disease)

```
set.seed(123)

k <-2; #number of biomarker
m <-2; #covariates dimension
d <-2; #number of class(0-no disease 1-disease)
N<- 1000 #number of observation

## B
beta1 <- c(0.1,0.3)
beta2 <- c(0.2,0.4)
B.0 <- matrix(c(beta1,beta2),2)

beta1 <- c(-0.1,0.2)
beta2 <- c(-0.2,0.3)
B.1 <- matrix(c(beta1,beta2),2)

# B.1 = B.0

## Delta
# delta <- sample(c(0,1),N, replace = T)


## X
x1 = rnorm(N,0,1)
x2 = rnorm(N,0,2)
X = data.frame(x1 = x1, x2 = x2)

## g.y
eta<- c(1,2)
d_prob = exp(as.matrix(X)%*%eta)/(1+exp(as.matrix(X)%*%eta))
d_cat = rbinom(N,1,d_prob)
```

```r
d_prob.0 = 1-d_prob[d_cat==0]
d_prob.1 = d_prob[d_cat==1]


X.0 = X[d_cat==0,]
X.1 = X[d_cat==1,]


## D
D <- list()
D$X <- rbind(X.0,X.1)
d_cat <- c(rep(0,nrow(X.0)),rep(1,nrow(X.1)))
D$d_cat <-  d_cat
D$d_prob <- c(d_prob.0, d_prob.1)
```

- Simulation1: ###Initial var1 = var2 = 0.3, and correlation = 0, without transformation

```
## [1] 1
## [1] 1
## [1] "success"
## [1] 1000
## [1] 2
## [1] 2
## [1] "success"
## [1] 1.728492
## [1] 3
## [1] 3
## [1] "success"
## [1] 0.4599164
## [1] 4
## [1] 4
## [1] "success"
## [1] 0.2193756
## [1] 5
## [1] 5
## [1] "success"
## [1] 0.11384
## [1] 6
## [1] 6
## [1] "success"
## [1] 0.1233321
## [1] 7
## [1] 7
## [1] "success"
## [1] 0.1098868
## [1] 8
## [1] 8
## [1] "success"
## [1] 0.09223654
## [1] 9
## [1] 9
## [1] "success"
## [1] 0.07796341
## [1] 10
```

```
## [1] 10
## [1] "success"
## [1] 0.06292622
## [1] 11
## [1] 11
## [1] "success"
## [1] 0.05196838
## [1] 12
## [1] 12
## [1] "success"
## [1] 0.04503555
## [1] 13
## [1] 13
## [1] "success"
## [1] 0.04075815
## [1] 14
## [1] 14
## [1] "success"
## [1] 0.0379811
## [1] 15
## [1] 15
## [1] "success"
## [1] 0.03599946
## [1] 16
## [1] 16
## [1] "success"
## [1] 0.03442759
## [1] 17
## [1] 17
## [1] "success"
## [1] 0.03302762
## [1] 18
## [1] 18
## [1] "success"
## [1] 0.03170706
## [1] 19
## [1] 19
## [1] "success"
## [1] 0.03041261
## [1] 20
## [1] 20
## [1] "success"
## [1] 0.02911869
## [1] 21
## [1] 21
## [1] "success"
## [1] 0.02783414
## [1] 22
## [1] 22
## [1] "success"
## [1] 0.02655791
## [1] 23
## [1] 23
## [1] "success"
```

```
## [1] 0.02528767
## [1] 24
## [1] 24
## [1] "success"
## [1] 0.02404576
## [1] 25
## [1] 25
## [1] "success"
## [1] 0.02283555
## [1] 26
## [1] 26
## [1] "success"
## [1] 0.02166203
## [1] 27
## [1] 27
## [1] "success"
## [1] 0.02052229
## [1] 28
## [1] 28
## [1] "success"
## [1] 0.01943253
## [1] 29
## [1] 29
## [1] "success"
## [1] 0.01838952
## [1] 30
## [1] 30
## [1] "success"
## [1] 0.01739259
## [1] 31
## [1] 31
## [1] "success"
## [1] 0.01643598
## [1] 32
## [1] 32
## [1] "success"
## [1] 0.01552809
## [1] 33
## [1] 33
## [1] "success"
## [1] 0.01466609
## [1] 34
## [1] 34
## [1] "success"
## [1] 0.01384839
## [1] 35
## [1] 35
## [1] "success"
## [1] 0.01307351
## [1] 36
## [1] 36
## [1] "success"
## [1] 0.01233986
## [1] 37
```

```
## [1] 37
## [1] "success"
## [1] 0.01164575
## [1] 38
## [1] 38
## [1] "success"
## [1] 0.01112185
## [1] 39
## [1] 39
## [1] "success"
## [1] 0.01039033
## [1] 40
## [1] 40
## [1] "success"
## [1] 0.009793049
## [1] 41
## [1] 41
## [1] "success"
## [1] 0.009224181
## [1] 42
## [1] 42
## [1] "success"
## [1] 0.008706356
## [1] 43
## [1] 43
## [1] "success"
## [1] 0.008225371
## [1] 44
## [1] 44
## [1] "success"
## [1] 0.00776114
## [1] 45
## [1] 45
## [1] "success"
## [1] 0.007321293
## [1] 46
## [1] 46
## [1] "success"
## [1] 0.006914192
## [1] 47
## [1] 47
## [1] "success"
## [1] 0.006521922
## [1] 48
## [1] 48
## [1] "success"
## [1] 0.006150948
## [1] 49
## [1] 49
## [1] "success"
## [1] 0.005801235
## [1] 50
## [1] 50
## [1] "success"
```

```
## [1] 0.005467542
## [1] 51
## [1] 51
## [1] "success"
## [1] 0.005154058
## [1] 52
## [1] 52
## [1] "success"
## [1] 0.004861552
## [1] 53
## [1] 53
## [1] "success"
## [1] 0.004585468
## [1] 54
## [1] 54
## [1] "success"
## [1] 0.004325461
## [1] 55
## [1] 55
## [1] "success"
## [1] 0.004080221
## [1] 56
## [1] 56
## [1] "success"
## [1] 0.003848988
## [1] 57
## [1] 57
## [1] "success"
## [1] 0.003630914
## [1] 58
## [1] 58
## [1] "success"
## [1] 0.003424358
## [1] 59
## [1] 59
## [1] "success"
## [1] 0.003255876
## [1] 60
## [1] 60
## [1] "success"
## [1] 0.003043787
## [1] 61
## [1] 61
## [1] "success"
## [1] 0.002872846
## [1] 62
## [1] 62
## [1] "success"
## [1] 0.002724177
## [1] 63
## [1] 63
## [1] "success"
## [1] 0.002563944
## [1] 64
```

```
## [1] 64
## [1] "success"
## [1] 0.002411615
## [1] 65
## [1] 65
## [1] "success"
## [1] 0.002288566
## [1] 66
## [1] 66
## [1] "success"
## [1] 0.00213144
## [1] 67
## [1] 67
## [1] "success"
## [1] 0.002034819
## [1] 68
## [1] 68
## [1] "success"
## [1] 0.001902512
## [1] 69
## [1] 69
## [1] "success"
## [1] 0.001814476
## [1] 70
## [1] 70
## [1] "success"
## [1] 0.001699965
## [1] 71
## [1] 71
## [1] "success"
## [1] 0.001602
## [1] 72
## [1] 72
## [1] "success"
## [1] 0.001506953
## [1] 73
## [1] 73
## [1] "success"
## [1] 0.001440294
## [1] 74
## [1] 74
## [1] "success"
## [1] 0.001333832
## [1] 75
## [1] 75
## [1] "success"
## [1] 0.001277076
## [1] 76
## [1] 76
## [1] "success"
## [1] 0.001183229
## [1] 77
## [1] 77
## [1] "success"
```

```
## [1] 0.001137852
## [1] 78
## [1] 78
## [1] "success"
## [1] 0.001049331
## [1] 79
## [1] 79
## [1] "success"
## [1] 0.001016419
## [1] "Converge in %s iteration: 80"
## [1] "Converge covariance matrix: "
##             [,1]        [,2]
## [1,]  0.29712420 -0.01902223
## [2,] -0.01902223  0.29836743
## [1] "Original covariance matrix: "
##      [,1] [,2]
## [1,]  0.3  0.0
## [2,]  0.0  0.3
```

- Simulation2: ###Initial var1 = 0.3, var2 = 0.4, and correlation = 0.2, without transformation

```
## [1] 1
## [1] 1
## [1] "success"
## [1] 1000
## [1] 2
## [1] 2
## [1] "success"
## [1] 0.8128374
## [1] 3
## [1] 3
## [1] "success"
## [1] 0.2323024
## [1] 4
## [1] 4
## [1] "success"
## [1] 0.1657146
## [1] 5
## [1] 5
## [1] "success"
## [1] 0.1722373
## [1] 6
## [1] 6
## [1] "success"
## [1] 0.1582521
## [1] 7
## [1] 7
## [1] "success"
## [1] 0.1171126
## [1] 8
## [1] 8
## [1] "success"
## [1] 0.07215318
## [1] 9
## [1] 9
```

```
## [1] "success"
## [1] 0.03808022
## [1] 10
## [1] 10
## [1] "success"
## [1] 0.03121417
## [1] 11
## [1] 11
## [1] "success"
## [1] 0.02954183
## [1] 12
## [1] 12
## [1] "success"
## [1] 0.02815532
## [1] 13
## [1] 13
## [1] "success"
## [1] 0.02691246
## [1] 14
## [1] 14
## [1] "success"
## [1] 0.02573327
## [1] 15
## [1] 15
## [1] "success"
## [1] 0.02456725
## [1] 16
## [1] 16
## [1] "success"
## [1] 0.02338681
## [1] 17
## [1] 17
## [1] "success"
## [1] 0.02218437
## [1] 18
## [1] 18
## [1] "success"
## [1] 0.02096742
## [1] 19
## [1] 19
## [1] "success"
## [1] 0.01975683
## [1] 20
## [1] 20
## [1] "success"
## [1] 0.01857386
## [1] 21
## [1] 21
## [1] "success"
## [1] 0.01743454
## [1] 22
## [1] 22
## [1] "success"
## [1] 0.01636327
```

```
## [1] 23
## [1] 23
## [1] "success"
## [1] 0.01535528
## [1] 24
## [1] 24
## [1] "success"
## [1] 0.01444561
## [1] 25
## [1] 25
## [1] "success"
## [1] 0.01353992
## [1] 26
## [1] 26
## [1] "success"
## [1] 0.01273448
## [1] 27
## [1] 27
## [1] "success"
## [1] 0.01205673
## [1] 28
## [1] 28
## [1] "success"
## [1] 0.01140102
## [1] 29
## [1] 29
## [1] "success"
## [1] 0.0107942
## [1] 30
## [1] 30
## [1] "success"
## [1] 0.01023169
## [1] 31
## [1] 31
## [1] "success"
## [1] 0.009711789
## [1] 32
## [1] 32
## [1] "success"
## [1] 0.009236765
## [1] 33
## [1] 33
## [1] "success"
## [1] 0.008792494
## [1] 34
## [1] 34
## [1] "success"
## [1] 0.008379779
## [1] 35
## [1] 35
## [1] "success"
## [1] 0.007993271
## [1] 36
## [1] 36
```

```
## [1] "success"
## [1] 0.007628589
## [1] 37
## [1] 37
## [1] "success"
## [1] 0.007284865
## [1] 38
## [1] 38
## [1] "success"
## [1] 0.006965935
## [1] 39
## [1] 39
## [1] "success"
## [1] 0.006666856
## [1] 40
## [1] 40
## [1] "success"
## [1] 0.006385533
## [1] 41
## [1] 41
## [1] "success"
## [1] 0.006120233
## [1] 42
## [1] 42
## [1] "success"
## [1] 0.005869553
## [1] 43
## [1] 43
## [1] "success"
## [1] 0.0056323
## [1] 44
## [1] 44
## [1] "success"
## [1] 0.005407436
## [1] 45
## [1] 45
## [1] "success"
## [1] 0.005194045
## [1] 46
## [1] 46
## [1] "success"
## [1] 0.004991306
## [1] 47
## [1] 47
## [1] "success"
## [1] 0.004799889
## [1] 48
## [1] 48
## [1] "success"
## [1] 0.004628415
## [1] 49
## [1] 49
## [1] "success"
## [1] 0.004455203
```

```
## [1] 50
## [1] 50
## [1] "success"
## [1] 0.00428713
## [1] 51
## [1] 51
## [1] "success"
## [1] 0.004125046
## [1] 52
## [1] 52
## [1] "success"
## [1] 0.003970022
## [1] 53
## [1] 53
## [1] "success"
## [1] 0.003826344
## [1] 54
## [1] 54
## [1] "success"
## [1] 0.003685582
## [1] 55
## [1] 55
## [1] "success"
## [1] 0.003550674
## [1] 56
## [1] 56
## [1] "success"
## [1] 0.0034207
## [1] 57
## [1] 57
## [1] "success"
## [1] 0.003300296
## [1] 58
## [1] 58
## [1] "success"
## [1] 0.003182617
## [1] 59
## [1] 59
## [1] "success"
## [1] 0.003068864
## [1] 60
## [1] 60
## [1] "success"
## [1] 0.002959284
## [1] 61
## [1] 61
## [1] "success"
## [1] 0.002857227
## [1] 62
## [1] 62
## [1] "success"
## [1] 0.002757599
## [1] 63
## [1] 63
```

```
## [1] "success"
## [1] 0.002660879
## [1] 64
## [1] 64
## [1] "success"
## [1] 0.002567914
## [1] 65
## [1] 65
## [1] "success"
## [1] 0.002478173
## [1] 66
## [1] 66
## [1] "success"
## [1] 0.00239463
## [1] 67
## [1] 67
## [1] "success"
## [1] 0.002313499
## [1] 68
## [1] 68
## [1] "success"
## [1] 0.002231832
## [1] 69
## [1] 69
## [1] "success"
## [1] 0.002154434
## [1] 70
## [1] 70
## [1] "success"
## [1] 0.002084045
## [1] 71
## [1] 71
## [1] "success"
## [1] 0.002013914
## [1] 72
## [1] 72
## [1] "success"
## [1] 0.001942687
## [1] 73
## [1] 73
## [1] "success"
## [1] 0.001877301
## [1] 74
## [1] 74
## [1] "success"
## [1] 0.001816235
## [1] 75
## [1] 75
## [1] "success"
## [1] 0.001753724
## [1] 76
## [1] 76
## [1] "success"
## [1] 0.001697159
```

```
## [1] 77
## [1] 77
## [1] "success"
## [1] 0.001636895
## [1] 78
## [1] 78
## [1] "success"
## [1] 0.0015867
## [1] 79
## [1] 79
## [1] "success"
## [1] 0.001536963
## [1] 80
## [1] 80
## [1] "success"
## [1] 0.001485344
## [1] 81
## [1] 81
## [1] "success"
## [1] 0.001438275
## [1] 82
## [1] 82
## [1] "success"
## [1] 0.001388909
## [1] 83
## [1] 83
## [1] "success"
## [1] 0.001345161
## [1] 84
## [1] 84
## [1] "success"
## [1] 0.001297333
## [1] 85
## [1] 85
## [1] "success"
## [1] 0.001253688
## [1] 86
## [1] 86
## [1] "success"
## [1] 0.001212494
## [1] 87
## [1] 87
## [1] "success"
## [1] 0.00117456
## [1] 88
## [1] 88
## [1] "success"
## [1] 0.001137352
## [1] 89
## [1] 89
## [1] "success"
## [1] 0.001096916
## [1] 90
## [1] 90
```

```
## [1] "success"
## [1] 0.001063122
## [1] 91
## [1] 91
## [1] "success"
## [1] 0.00102956
## [1] "Converge in %s iteration: 92"
## [1] "Converge covariance matrix: "
##             [,1]       [,2]
## [1,] 0.28339449 0.05800791
## [2,] 0.05800791 0.36813036
## [1] "Original covariance matrix: "
##             [,1]       [,2]
## [1,] 0.30000000 0.06928203
## [2,] 0.06928203 0.40000000
```