Illustrating Guided BAC codes

Below we will illustrate how to use the guided BAC codes from https://github.com/jantonelli111/Guided-BAC

In this example we will restrict attention to a binary treatment and continuous outcome so we will use the function called ContinuousBinary(). First we will simulate a small data set to illustrate the software.

```
## Our main study includes 800 = 1000 - 200 subjects
## Our validation study has 200 subjects
N = 1000
n = 200

StudyInd = c(rep(1, n), rep(0, N-n))

## 5 covariates fully observed and 5 missing
P = 10
M=5

Cobs = matrix(rnorm(M*N), N,M)
Cmis = matrix(rnorm((P-M)*N), N, P-M)

C = cbind(Cobs, Cmis)

## simulate an exposure
X.lin = -1 + 0.5*C[,1] + 0.6*C[,6]
X = rbinom(N, 1, p=pnorm(X.lin))

## simulate an outcome
Y.lin = 3 + 1*X + 0.3*X*StudyInd + 0.5*C[,1] + 0.6*C[,6]
sigmaY = 2
Y = rnorm(N, mean=Y.lin, sd=sqrt(sigmaY))
```

So now we have generated a matrix of covariates and simulated an exposure an outcome where the 1st and 6th covariates are confounders. Now for a very important part of how the software works. The functions to calculate the treatment effects take in two matrices of covariates. The first, called Cobs, is the fully observed covariates and should just be an N by M matrix of covariates where the first n rows must represent the subjects in the validation study. The second, called Cmis, represents the covariates missing in the main study. It should be of dimension N by (P-M) and the first n subjects should be observed while the remaining rows should be all NAs. Importantly each row of Cobs should correspond to the same row of Cmis, i.e the row indices represent the same subjects in both matrices. Below we see how the Cmis matrix should look:

```
Cmis[(n+1) : N,] = NA

head(Cmis)
```

```
##             [,1]       [,2]       [,3]        [,4]        [,5]
## [1,] -0.97400425  0.5363179 -0.4682980 -1.43232829 -0.39749073
## [2,]  0.06695866  1.7409370  1.3686487 -0.25262394 -0.09534817
## [3,]  0.86629336 -1.1666656 -1.0241405  1.00934407 -1.19351412
## [4,] -0.33813742  0.8954829  0.0959469 -0.35944317 -0.61598590
## [5,] -1.33247258  0.5706661  0.8320258  0.03144104 -0.55617422
## [6,]  0.87715198 -0.9536663  0.6951957 -0.01549453 -0.42055228
```

```
tail(Cmis)
```

```
##          [,1] [,2] [,3] [,4] [,5]
## [995,]    NA   NA   NA   NA   NA
## [996,]    NA   NA   NA   NA   NA
## [997,]    NA   NA   NA   NA   NA
## [998,]    NA   NA   NA   NA   NA
## [999,]    NA   NA   NA   NA   NA
## [1000,]   NA   NA   NA   NA   NA
```

Now that we have our data we can fit the model. The functions to calculate the ATE should only require 5 inputs:

1. Cobs
2. Cmis
3. X the vector of treatment values
4. Y the vector of outcome values
5. covType, a vector of length P taking values either "Continuous" or "Binary" representing whether each covariate in the data is continuous or binary

In our simulated example, CovType looks like the following:

```
covType = rep("Continuous", P)
covType
```

```
##  [1] "Continuous" "Continuous" "Continuous" "Continuous" "Continuous"
##  [6] "Continuous" "Continuous" "Continuous" "Continuous" "Continuous"
```

And then there are optional inputs

1. omega, which dictates the strength of dependence between treatment and outcome model inclusion probabilities
2. nScans, the total number of MCMC scans
3. burn, the number of scans we drop as a burn in
4. thin, how often we thin the MCMC samples
5. savePosterior, which is a true/false variable of whether we should save the whole posterior so that diagnostics and plots can be made. If FALSE then only the posterior means and CI will be reported.

Now let's fit Guided BAC to our simulated data. To do so you will need to have installed the MASS, mvtnorm, and truncnorm packages. Also you will need to enter into ATEfunctions.R and set the working directory (first line of code) to the appropriate folder where all of these R codes live on your local machine.

```
## First read in file, which contains all of the R functions
source("ATEfunctions.R")

## Now fit Guided BAC
burn = 100
nScans = 500
thin = 1
omega = 1000

GBAC = ContinuousBinary(Cobs=Cobs,
                        Cmis=Cmis,
                        X=X,
                        Y=Y,
                        covType=covType,
                        omega=omega,
                        burn=burn,
                        nScans=nScans,
                        thin=thin,
                        savePosterior=FALSE)
```

```
## [1] 100
## [1] 200
## [1] 300
## [1] 400
## [1] 500
```

```
GBAC
```

```
## $InclusionProbX
##  [1] 1 0 0 0 0 1 0 0 0 0
##
## $InclusionProbY
##  [1] 1.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.1500 0.0625 0.0450 0.0375
##
## $ATE
## [1] 0.7949889
##
## $ATE_CI
##      2.5%     97.5%
## 0.4146723 1.1893256
```

And we see that the model selects variables 1 and 6 with high probability because they're confounders and not the other variables. The ATE is around 1, which is the true value in this instance.