R code to re-create a simulation for the exponential decay structure for equal cluster sizes

```
# Packages and source programs
library(geepack)
source('FUNCTIONS.R')

# Design
I <- 20         # number of clusters
n <- 30         # number of participants per cluster per period
T <- 5          # number of time periods
q <- I/(T-1)    # number of clusters randomized at each step
sig <- 0.05     # significance level
p0 <- 0.3       # baseline prevalence under control
beta <- c(log(p0/(1-p0)),-0.87,-0.89,-0.91,-0.93) # gently decreasing time-effect
theta <- 0.4592 # intervention effect on the log odds ratio scale
tau <- 0.06     # within-period correlation
rho <- 0.7715   # decay parameter
times <- 1:T
varrho <- rho^abs(outer(times,times,"-"))
R <- (1-tau)*diag(n*T) + kronecker(tau*varrho,matrix(1,nrow=n,ncol=n)) # exponential decay correlation
trtSeq <- matrix(0,T-1,T)
trtSeq[upper.tri(trtSeq)] <- 1  # treatment sequences

# Simulate data
set.seed(20210513)
B <- create.B(I,T,q,beta,theta,trtSeq,n,R) # Create b's as in Qaqish equation 3
y <- create.response(I,T,q,beta,theta,trtSeq,n,B) # Simulate binary outcomes (Qaqish equation 3)

# Fit the model with the true correlation structure
X <- NULL    # Create X matrix for covariates in FITEXPDECAY
for(i in 1:(T-1)){
  for(d in 1:q){
    X <- rbind(X,kronecker(cbind(diag(T),trtSeq[i,]),rep(1,n)))
  }
}
clsize <- rep(n*T,I) # cluster sizes (across all the periods)
clpersize <- clsize/T # cluster-period sizes
fit <- FITEXPDECAY(y=y, X=X, clsize=clsize, clpersize=clpersize, T=T, maxiter=100, epsilon=0.0001)
```

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

```
## Warning: package 'MASS' was built under R version 4.0.2
```

```
fit # mean model, model-based SEs, sandwich SEs, Mancl DeRouen sandwich SEs and correlations
```

```
## $coefficients
##      beta estimates model-based SE   sand SE     MD SE
## time     -0.7681932      0.1329157 0.1484488 0.1562721
## time     -1.0917163      0.1478564 0.1354834 0.1454501
## time     -1.2230109      0.1677565 0.1774223 0.1920898
## time     -1.2094374      0.1885690 0.2050209 0.2241727
## time     -1.3018957      0.2132142 0.2152212 0.2346325
## trt       0.5160340      0.1615579 0.1560288 0.1740178
##
## $corr_parameters
##     alpha estimates
## tau      0.04463962
## rho      0.81261021
##
## $model_fit
##          model converged     number of iterations corr parameters in range
##                   "TRUE"                      "8"                   "TRUE"
```

```
# Fit the model with the working exchangeable correlation structure
cluster <- rep(1:I,each=T*n)
period <- rep(rep(1:T,each=n),I)
X.ij <- kronecker(trtSeq,rep(1,q))
treatment <- rep(c(t(X.ij)),each=n)
simdata_bin <- data.frame(cbind(y,cluster,period,treatment))
fit_exch <- geeglm(y~treatment+factor(period),id=cluster,data=simdata_bin,
                 family=binomial(link = "logit"), scale.fix=TRUE, corstr="exchangeable")
summary(fit_exch)  # mean model, sandwich SEs and correlation
```

```
##
## Call:
## geeglm(formula = y ~ treatment + factor(period), family = binomial(link = "logit"),
##     data = simdata_bin, id = cluster, corstr = "exchangeable",
##     scale.fix = TRUE)
##
##  Coefficients:
##                  Estimate Std.err   Wald Pr(>|W|)
## (Intercept)       -0.7694  0.1483 26.909 2.13e-07 ***
## treatment          0.5354  0.1660 10.401  0.00126 **
## factor(period)2   -0.3281  0.1482  4.903  0.02682 *
## factor(period)3   -0.4642  0.1916  5.870  0.01540 *
## factor(period)4   -0.4552  0.2160  4.441  0.03508 *
## factor(period)5   -0.5508  0.2093  6.927  0.00849 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = exchangeable
## Scale is fixed.
##
##   Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha  0.03321 0.01221
## Number of clusters:   20  Maximum cluster size: 150
```

```
sqrt(fit_exch$geese$vbeta.naiv[2,2]) # model-based SE for the treatment effect
```

```
## [1] 0.1398
```

```
# Get Mancl DeRouen sandwich variance
alpha <- summary(fit_exch)$corr[1,1]
var <- (1-alpha)*diag(n*T) + alpha*matrix(1,nrow=n*T,ncol=n*T)
md_var <- md.var(fit_exch,simdata_bin,var)
sqrt(md_var) # Mancl DeRouen sandwich SE for the treatment effect
```

```
## [1] 0.1851
```

```
# Fit the model with the working independence correlation structure
fit_indep <- geeglm(y~treatment+factor(period),id=cluster,data=simdata_bin,
                 family=binomial(link = "logit"), scale.fix=TRUE, corstr="independence")
summary(fit_indep) # mean model and sandwich SEs
```

```
##
```

```
## Call:
## geeglm(formula = y ~ treatment + factor(period), family = binomial(link = "logit"), 
##     data = simdata_bin, id = cluster, corstr = "independence", 
##     scale.fix = TRUE)
## 
##  Coefficients:
##                 Estimate Std.err  Wald Pr(>|W|)    
## (Intercept)       -0.769   0.148 26.88  2.2e-07 ***
## treatment          0.429   0.225  3.66    0.056 .  
## factor(period)2   -0.298   0.156  3.67    0.055 .  
## factor(period)3   -0.409   0.215  3.63    0.057 .  
## factor(period)4   -0.376   0.256  2.15    0.142    
## factor(period)5   -0.445   0.278  2.56    0.110    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation structure = independence 
## Scale is fixed.
## 
## Number of clusters:   20  Maximum cluster size: 150
```

```r
sqrt(fit_indep$geese$vbeta.naiv[2,2]) # model-based SE for the treatment effect
```

```
## [1] 0.115
```

```r
# Get Mancl DeRouen sandwich variance
var <- diag(n*T)
md_var <- md.var(fit_indep,simdata_bin,var)
sqrt(md_var) # Mancl DeRouen sandwich SE for the treatment effect
```

```
## [1] 0.254
```