R code to implement the power calculation method

```r
# Source the R functions needed
source('FUNCTIONS.R')

# Design parameters for the illustrative example
I <- 96      # number of clusters
n <- 40      # number of participants per cluster per period
T <- 5       # number of time periods
p0 <- 0.08   # prevalence under control
p1 <- 0.064 # prevalence under intervention
theta_star <- log(p1/(1-p1))-log(p0/(1-p0)) # log odds ratio
sig <- 0.05 # alpha significance level
rho <- 0.1   # inter-cluster correlation

# Power from linear model equations in Web Appendix A
linear_no_time <- function(p0,p1,rho,n,T,I,sig){
  sigma_sq <- (p1*(1-p1)+p0*(1-p0))/2
  var <- 12*sigma_sq*(1-rho)*(1-rho+n*T*rho)/(n*I*T*(3*(1-rho)+2*n*rho*(T+1)))
  test_stat <- abs((p1-p0)/sqrt(var))
  power <- pnorm(test_stat-qnorm(1-sig/2))
  return(power)
}
linear_no_time(p0,p1,rho,n,T,I,sig)
```

```
## [1] 0.982436
```

```r
# Power from logistic model equations in Web Appendix A
logistic_no_time <- function(p0,p1,theta_star,rho,n,T,I,sig){
  a <- sqrt(p0*(1-p0))
  b <- sqrt(p1*(1-p1))
  var <- (6*(1-rho)*(a^2+b^2)+2*n*rho*(T+1)*(a-b)^2)*(1-rho+n*T*rho)/
         (n*I*T*a^2*b^2*(3*(1-rho)+2*n*rho*(T+1)))
  test_stat <- abs(theta_star/sqrt(var))
  power <- pnorm(test_stat-qnorm(1-sig/2))
  return(power)
}
logistic_no_time(p0,p1,theta_star,rho,n,T,I,sig)
```

```
## [1] 0.9775285
```

```r
# Power from equations 2 and 3
linear_time <- function(p0,p1,rho,n,T,I,sig){
  sigma_sq <- (p1*(1-p1)+p0*(1-p0))/2
  var <- 12*sigma_sq*(T-1)*(1-rho)*(1-rho+n*T*rho)/(n*I*T*(T-2)*(2*(1-rho)+n*rho*(T+1)))
  test_stat <- abs((p1-p0)/sqrt(var))
  power <- pnorm(test_stat-qnorm(1-sig/2))
  return(power)
}
linear_time(p0,p1,rho,n,T,I,sig)
```

```
## [1] 0.7096853
```

```r
# Power from equations 4 and 5
logistic_time <- function(beta,theta_star,R,n,T,I,sig){
  invR <- solve(R)
  trtSeq <- matrix(0,T-1,T)
```

```
  trtSeq[upper.tri(trtSeq)] <- 1
  Z_part <- kronecker(diag(T),rep(1,n))
  q <- I/(T-1)
  var <- var.logistic(I,T,q,beta,theta_star,trtSeq,n,invR,Z_part)
  test_stat <- abs(theta_star/sqrt(var))
  power <- pnorm(test_stat - qnorm(1-sig/2))
  return(power)
}
R_exch <- (1-rho)*diag(n*T) + rho*matrix(1,nrow=n*T,ncol=n*T) # exchangeable correlation matrix
beta <- c(log(p0/(1-p0)), # decrease in probability of an event only at step 1 (then maintained)
          rep(log(p0/(1-p0))-0.15,T-1))
logistic_time(beta,theta_star,R_exch,n,T,I,sig)
```

```
## [1] 0.6442509
```

```
# Power from equations 5 and 6
rho1 <- 0.1  # within-period correlation
rho2 <- 0.01 # inter-period correlation
R_nest <- (1-rho1)*diag(n*T) +
          (rho1-rho2)* kronecker(diag(T),matrix(1,nrow=n,ncol=n)) +
          rho2*matrix(1,nrow=n*T,ncol=n*T) # nested exchangeable correlation matrix
logistic_time(beta,theta_star,R_nest,n,T,I,sig)
```

```
## [1] 0.2397403
```

```
# Power from equation 5 and 7
tau <- 0.1  # within-period correlation
rho <- 0.56 # decay parameter
times <- 1:T
varrho <- rho^abs(outer(times,times,"-"))
R_exp <- (1-tau)*diag(n*T) +
          kronecker(tau*varrho,matrix(1,nrow=n,ncol=n)) # exponential decay correlation matrix
logistic_time(beta,theta_star,R_exp,n,T,I,sig)
```

```
## [1] 0.264653
```

```
# Power from equation 5 and 8
logistic_trt_het_time <- function(beta,theta_star,rho1,rho2,rho3,n,T,I,sig){
  trtSeq <- matrix(0,T-1,T)
  trtSeq[upper.tri(trtSeq)] <- 1
  Z_part <- kronecker(diag(T),rep(1,n))
  q <- I/(T-1)
  var <- var.logistic.trt.het(I,T,q,beta,theta_star,trtSeq,n,rho1,rho2,rho3,Z_part)
  test_stat <- abs(theta_star/sqrt(var))
  power <- pnorm(test_stat - qnorm(1-sig/2))
  return(power)
}
rho1 <- 0.1  # inter-cluster correlation (control, control)
rho2 <- 0.02 # inter-cluster correlation (control, treatment)
rho3 <- 0.08 # inter-cluster correlation (treatment, treatment)
logistic_trt_het_time(beta,theta_star,rho1,rho2,rho3,n,T,I,sig)
```

```
## [1] 0.2484286
```

```
# Varying cluster sizes
n_mod <- c(62,63,64,65,65,66,67,68,
           35,36,37,38,39,40,41,42,43,44,45,46,47,rep(48,6),49,50,51,52,53,54,55,56,57,58,59,60,61,
           32,33,34,35,36,37,38,rep(35,10),rep(30,14),rep(40,14),
           14,15,16,17,18,19,20,21,22,23,24) # moderate cluster size variation
mean(n_mod)
```

```
## [1] 40
```

```r
sd(n_mod)/mean(n_mod)
```

```
## [1] 0.3196215
```

```r
n_lar <- c(70,74,76,92,134,172,213,302,
           35,36,37,38,38,39,39,40,40,41,42,43,44,45,46,47,rep(48,3),49,50,51,52,53,54,55,56,57,58,
           59,60,61,
           12,15,18,19,20,20,21,21,22,22,23,25,25,25,25,26,27,28,30,31,31,31,34,35,37,38,39,39,39,
           rep(35,2),rep(30,3),rep(40,3),45,
           1,2,2,2,3,3,3,4,4,4,5,6,7,8,10,10,10,11) # large cluster size variation
mean(n_lar)
```

```
## [1] 40
```

```r
sd(n_lar)/mean(n_lar)
```

```
## [1] 1.036403
```

```r
# Design parameters
rho <- 0.1
q <- I/(T-1)
trtSeq <- matrix(0,T-1,T)
trtSeq[upper.tri(trtSeq)] <- 1
X.ij <- kronecker(trtSeq,rep(1,q))

# Power from equations 11 and 12
set.seed(3656)
M <- 1000   # Number of randomization sequences to be used
power <- matrix(NA,M,2)
colnames(power) <- c("moderate","large")
for (m in 1:M) {
  first <- sample(1:I,q)
  second <- sample((1:I)[-first],q)
  third <- sample((1:I)[-c(first,second)],q)
  fourth <- (1:I)[-c(first,second,third)]
  S.ij <- X.ij[c(first,second,third,fourth),]   # randomization sequence
  var <- var.logistic.vary(I,T,q,beta,theta_star,S.ij,n_mod,rho)
  test_stat <- abs(theta_star/sqrt(var))
  power[m,1] <- pnorm(test_stat - qnorm(1-sig/2))
  var <- var.logistic.vary(I,T,q,beta,theta_star,S.ij,n_lar,rho)
  test_stat <- abs(theta_star/sqrt(var))
  power[m,2] <- pnorm(test_stat - qnorm(1-sig/2))
}
colMeans(power) # mean power
```

```
##  moderate     large
## 0.6433369 0.6402897
```

```r
apply(power,2,min) # minimum power
```

```
##  moderate     large
## 0.6342235 0.6178242
```

```r
apply(power,2,max) # maximum power
```

```
##  moderate     large
## 0.6479599 0.6489909
```