Supplementary appendix 2. The WinBUGS code for random effects model.

The WinBUGS code for random effects model.

```
# Binomial likelihood, cloglog link
# Random effects model for multi-arm trials
model{                                          # *** PROGRAM STARTS
for(i in 1:ns){                                 # LOOP THROUGH STUDIES
     w[i,1] <- 0      # adjustment for multi-arm trials is zero for control arm
     delta[i,1] <- 0                  # treatment effect is zero for control arm
     mu[i]  ~ dnorm(0,.0001)                # vague priors for all trial baselines
     for (k in 1:na[i]) {                   # LOOP THROUGH ARMS
          r[i,k] ~ dbin(p[i,k],n[i,k]) # Binomial likelihood
# model for linear predictor
          cloglog(p[i,k]) <- log(time[i]) + mu[i] + delta[i,k]
          rhat[i,k] <- p[i,k] * n[i,k] # expected value of the numerators
#Deviance contribution
          dev[i,k] <- 2 * (r[i,k] * (log(r[i,k])-log(rhat[i,k]))
               +   (n[i,k]-r[i,k]) * (log(n[i,k]-r[i,k]) - log(n[i,k]-rhat[i,k])))          }
#   summed residual deviance contribution for this trial
     resdev[i] <- sum(dev[i,1:na[i]])
     for (k in 2:na[i]) {                   # LOOP THROUGH ARMS
# trial-specific LOR distributions
          delta[i,k] ~ dnorm(md[i,k],taud[i,k])
# mean of LOR distributions, with multi-arm trial correction
          md[i,k] <-   d[t[i,k]] - d[t[i,1]] + sw[i,k]
# precision of LOR distributions (with multi-arm trial correction)
          taud[i,k] <- tau *2*(k-1)/k
# adjustment, multi-arm RCTs
          w[i,k] <- (delta[i,k] - d[t[i,k]] + d[t[i,1]])
# cumulative adjustment for multi-arm trials
          sw[i,k] <- sum(w[i,1:k-1])/(k-1)
        }
  }
totresdev <- sum(resdev[])                      #Total Residual Deviance
d[1]<-0           # treatment effect is zero for reference treatment
# vague priors for treatment effects
for (k in 2:nt){    d[k] ~ dnorm(0,.0001) }
sd ~ dunif(0,2)         # vague prior for between-trial SD
tau <- pow(sd,-2)       # between-trial precision = (1/between-trial variance)

for (c in 1:(nt -1)) {
for (k in (c+1):nt) {
or[c,k] < -   exp(d[k]   -   d[c])
lor[c,k] < -   (d[k] -d[c])
}
}
#Ranking of treatments#

    for(k in 1:nt) {
        order[k]<- rank(d[],k)
# this is when the outcome is positive - omit   'nt+1-' when the outcome is negative
        most.effective[k]<-equals(order[k],1)

        for(j in 1:nt) {
            effectiveness[k,j]<- equals(order[k],j)
        }
    }
    for(k in 1:nt) {
        for(j in 1:nt) {
            cumeffectiveness[k,j]<- sum(effectiveness[k,1:j])
        }
    }
#SUCRAS#

    for(k in 1:nt) {
        SUCRA[k]<- sum(cumeffectiveness[k,1:(nt-1)]) /(nt-1)
    }
# ranking on relative scale
for (k in 1:nt) {
# rk[k] < -   nt+1 -rank(d[],k) # assumes events are “good”
rk[k] < -   rank(d[],k) # assumes events are “bad”
best[k] < -   equals(rk[k],1) #calculate probability that treat k is best
}
}                                           # *** PROGRAM ENDS

#Data
# ns= number of studies; nt=number of treatments
list(ns=36, nt=15)
t[,1] r[,1] n[,1]    t[,2] r[,2] n[,2]    t[,3] r[,3] n[,3]    t[,4] r[,4] n[,4]    na[] time[]
...
...
.
.
.
END
```