**Appendix 2.** The OpenBUGS code for random effects model

```openbugs
# Binomial likelihood, cloglog link
# Random effects model for multi-arm trials
model{  
  # *** PROGRAM STARTS
  for(i in 1:ns){  
    # adjustment for multi-arm trials is zero for control arm
    w[i,1] <- 0
    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]
    }  
    #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]){  
    # 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 * pow(k,-2)
    # 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[i])  
  # *** PROGRAM ENDS
}
```

# vague priors for treatment effects
for (k in 2:nt) {  
  d[k] ~ dnorm(0,0.0001)  
}

# vague prior for between-trial SD
sd ~ dunif(0,5)

# between-trial precision = (1/between-trial variance)
for (c in 1:(nt-1)) {  
  tau <- pow(sd,-2)
  for (k in (c+1):nt) {  
    or[c,k] <- exp(d[k] - d[c])
    or[c,k] <- (d[k]-d[c])
  }
}

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

for(i in 1:nt) {  
  order[k] <- rank(d[k],k)
  effectiveness[k,i] <- equals(order[k],i)
}

for(k 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)
}

# *** PROGRAM ENDS