**Supplemental 3** R functions for the choice of prior

**FEMetadata.R**

## Create FE meta-analysis data

Takes in necessary parameters and creates fixed effect Meta-analysis data-sets

@description FEMetadata function creates meta-analysis data using probability of control group (p\_con), number of experiments (n\_exp), odds ratio (OR), number of trials in each meta-analysis file (ntrials), and the ratio of experimental group versus control (ratio, e.g. 1:1 etc.) and provides any number of data-sets we need using (nsim). The function saves the result in folderdata created in designated directory.

@param folderdata The name of the folder where implemented data files will be saved p\_con Probability of event in control group (baseline risk), n\_exp Total number of patients in experimental group, OR True Odds Ratio nsim Number of Meta-analysis data files to be created with specified characteristics ntrials Number of studies in each meta-analysis data-set ratio The balance between number of patients in experiment vs. control groups

@return Returns FE Meta-analysis data files with the specified characteristics

FEMetadata <- function(folderdata, Min.p\_con, Max.p\_con, Min.n\_exp, Max.n\_exp, logOR, nsim, ntrials, ratio) {

 *#' Create folder*
 **dir.create**(folderdata)

 path\_results <- **file.path**(**getwd**(), folderdata)

 for (i in 1:nsim) {
 **set.seed**(456+i)

 OR<-**exp**(logOR)
 p\_con<-**round**(**runif**(n=ntrials, min=Min.p\_con, max=Max.p\_con), 4)
 n\_exp<-**runif**(n=ntrials, min=Min.n\_exp, max=Max.n\_exp)
 odds\_con<-p\_con/(1-p\_con)
 odds\_exp<-odds\_con\*OR
 p\_exp<-**round**((odds\_exp/(1+odds\_exp)), 4)

 *#' rounding the number of events*
 n\_exp<-**round**(n\_exp)

 *#' set the ratio of experimental group vs. control group*
 n\_con<-**round**(ratio\*n\_exp)

 ev\_exp<- **rbinom**(n=ntrials,size=n\_exp,prob=p\_exp)
 ev\_con<- **rbinom**(n=ntrials,size=n\_con,prob=p\_con)
 Data <- **as.data.frame**(**cbind**(ev\_exp, n\_exp, ev\_con, n\_con, **round**(OR, 4), **round**(p\_con, 4)))
 **colnames**(Data)<-**c**("ev\_exp", "n\_exp", "ev\_con", "n\_con", "OR", "p\_con")
 filename <-**paste0**(i, ".txt", collapse=NULL)
 **write.table**(Data, file = **file.path**(path\_results,filename), sep = "\t", row.names = FALSE, col.names = TRUE)
 **rm**(Data)
 }
}

**REMetadata.R**

## Create RE meta-analysis data

Takes in necessary parameters and creates random effects Meta-analysis data-sets

@description REMetadata function creates RE meta-analysis data using probability in control group (p\_con), number of experiments (n\_exp), odds ratio (OR), number of trials in each meta-analysis file (ntrials) and the ratio of control group versus experimental (ratio) and provides any number of data-set we want using (nsim). The function saves the result in folderdata created in respective working directory.

@param folderdata The name of the folder implemented data files will be saved p\_con Probability of event in control group (baseline risk), n\_exp Total number of patients in experimental group, OR True Odds Ratio nsim Number of Meta-analysis data files to be created with specified characteristics ntrials Number of studies in each data-set ratio The balance between number of patients in experiment vs. control groups

@return Returns RE Meta-analysis data files with the specified characteristics

REMetadata <- function (folderdata, Min.p\_con, Max.p\_con, Min.n\_exp, Max.n\_exp, meanlogOR, sdlogOR, nsim, ntrials, ratio) {

 *#' Create folder*
 **dir.create**(folderdata)

 path\_results <- **file.path**(**getwd**(), folderdata)

 for (i in 1:nsim) {
 **set.seed**(456+i)

 *#' preparation of initials to create meta-analysis data-set*

 OR<-**exp**(**rnorm**(n=ntrials, mean=meanlogOR, sd=sdlogOR))
 p\_con<-**round**(**runif**(n=ntrials, min=Min.p\_con, max=Max.p\_con), 4)
 n\_exp<-**runif**(n=ntrials, min=Min.n\_exp, max=Max.n\_exp)
 odds\_con<-p\_con/(1-p\_con)
 odds\_exp<-odds\_con\*OR
 p\_exp<-**round**(odds\_exp/(1+odds\_exp), 4)

 *#' rounding the number of events*
 n\_exp<-**round**(n\_exp)

 *#' ratio of experimental group to control group*
 n\_con<-**round**(ratio\*n\_exp)

 ev\_exp<- **rbinom**(n=ntrials,size=n\_exp,prob=p\_exp)
 ev\_con<- **rbinom**(n=ntrials,size=n\_con,prob=p\_con)

 Data <- **as.data.frame**(**cbind**(ev\_exp, n\_exp, ev\_con, n\_con, **round**(OR, 4), **round**(p\_con, 4)))
 **colnames**(Data)<-**c**("ev\_exp", "n\_exp", "ev\_con", "n\_con", "OR", "p\_con")
 filename <-**paste0**(i, ".txt", collapse=NULL)
 **write.table**(Data, file = **file.path**(path\_results,filename), sep = "\t", row.names = FALSE, col.names = TRUE)
 **rm**(Data)
 }
}

**Fcheck.R**

## Sanity check for implemented data

To check if the simulated data set(s) has the characteristics we defined in FEMetadata

@description We implement data using FEMetadata & REMetadata and to make sure the data-files contain the characteristics that we already introduced to both functions, we recalculate log(OR) and also relevant standard deviation. Corresponding file is being saved in result folder.

@param folderdata It is the folder which contains the meta-analysis data files, folderresults The folder we want to save the result filename Name of the file to save the result of this function, nsim Number of meta-analysis data-sets that we want to check

@return It reads the data-files and calculates the mean log(OR) and standard deviation. It also creates a txt file with the corresponding information for each datafile.

datacheck <-function(folderdata, folderresults, filename, nsim){

 *# Read the generated data*

 **dir.create**(folderresults)

 path\_data <- **file.path**(**getwd**(), folderdata)

 path\_results <- **file.path**(**getwd**(), folderresults)
 results <- **matrix**(**rep**(NA, 2\*nsim), nrow = nsim, ncol = 2)

 for (i in 1:nsim) {

 files <- **list.files**(path\_data)

 files <- **namesort**(files)
 datafiles<-files[i]

 Metaanalysis<-**read.table**(**file.path**(path\_data, datafiles), header=TRUE)
 meanlogOR <-**round**(**mean**(**log**(Metaanalysis$OR)), 5)
 sdlogOR <-**round**(**sd**(**log**(Metaanalysis$OR)), 5)

 *# Write the file in the result folder*

 results[i, 1] <- meanlogOR
 results[i, 2] <- sdlogOR

 }
 *#' Read the result file from result path*
 *# results <-read.table(file=file.path(path\_results, filename), header=T)*
 **colnames**(results) <- **c**("meanlogOR", "sdlogOR")
 **write.table**(results, file=**file.path**(path\_results, filename), sep="\t", col.names = T, row.names = F)
 **summary**(results)
}

**Fcountzeros.R**

## Count the number of trials with zero in both arms and one arm (experimental & control)

@description Reads the meta-analysis data set(s) and returns the summary for number of trials with zero in both arms, number of zeros in experimental group and control group, respectively. It restores the result file under the filename we give it in the folder which creates using folderresults. It will give a warning message if folder already exists but It will restore the result.

@param folderdata It is the folder which contains the meta-analysis data files, folderresults The folder we want to save the result file of the current function filename Name of the file to save the result of the function, ntrials Number of studies in each meta-analysis data file nsim Number of meta-analysis data-sets that we want to analyse

@return It reads the meta-analysis data-files and counts the number of zeros in each data set. It gives the result for zero in both arms and also number of zeros in Experimental group and also control group.

countzs<-function(folderdata, folderresults, filename, ntrials, nsim){
 **require**(dplyr)
 *#' Read the generated data*

 path\_data <- **file.path**(**getwd**(), folderdata)

 **dir.create**(folderresults)
 path\_results <- **file.path**(**getwd**(), folderresults)

 for (i in 1:nsim) {

 datafiles<-**list.files**(path\_data)
 files <- **namesort**(datafiles)
 dfiles<-files[i]
 Metaanalysis<-**read.table**(**file.path**(path\_data, dfiles), header=TRUE)

 *#' count number of zeros*
 dl.Metaanalysis<-**subset**(Metaanalysis, (ev\_exp == 0 & ev\_con == 0)==0)
 nzeros.dl<-**nrow**(dl.Metaanalysis)
 nzeros<-(ntrials-nzeros.dl)
 nzexp<-**length**(Metaanalysis$ev\_exp[Metaanalysis$ev\_exp==0])
 nzcon<-**length**(Metaanalysis$ev\_con[Metaanalysis$ev\_con==0])

 nzdata<-**as.data.frame**(**cbind**(nzeros, nzexp, nzcon))
 **colnames**(nzdata)<-**c**("nzeros", "nz\_exp", "nz\_con")

 *#' Write the file in the result folder*

 if(i == 1) {
 **write.table**(nzdata,file=**file.path**(path\_results, filename), sep="\t", col.names = T, row.names = F)
 } else {
 **write.table**(nzdata,file=**file.path**(path\_results, filename), append=T, sep="\t", col.names = F, row.names = F)
 }

 }
 *#' Read the result file from result path*
 count <-**read.table**(file=**file.path**(path\_results, filename), header=T)

 **return**(**list**(Botharms=**summary**(count$nzeros), Experimental=**summary**(count$nz\_exp), Control=**summary**(count$nz\_con)))
}

**FERJAGS.R**

## Fixed effect Meta-analysis of rare events clinical trials

Using fixed effect JAGS models, function analysis the data-sets for meta-analysis of rare events clinical trials with binary outcome

@description FERJAGS reads the meta-analysis data set(s) and using the JAGS model (runs JAGS through R using jags() command of r2jags package) we introduced in the args returns the result file in numerical order for each data-set and saves the log(OR) (mu) and standard deviation (sd), 95% credible interval, n.eff and R.hat

@param folderdata It is the folder which contains the meta-analysis data files, folderresults The folder which the JAGS result will be saved, nsim Number of meta-analysis data-sets that we want to analyse, DL LOGICAL If TRUE deletes studies with zero in both arm and analysis the rest of the data, n.iter number of iteration for MCMC machinery, n.burnin number of burnning in for MCMC, pars Parameters that we need for our meta-analysis model(s), initials the initial values for relative priors in the corresponding model(s), modelfile Text file of the model written in JAGS, filename Name of the file to save the result of the Bayesian approach.

@return Fixed effect RJAGS (FERJAGS) function reads the meta-analysis data-sets and based on the model we define and also the characteristics for MCMC machinery (check the arguments) of JAGS calculates the posterior information for odds ratio (mu=log(OR))

FERJAGS<- function (folderdata, folderresults,nsim, DL, n.iter, n.burnin, pars, initials, modelfile, filename) {

 Results<-NULL

 *#' create result folder*
 **dir.create**(folderresults)

 *#' Read the generated data*

 path\_data <- **file.path**(**getwd**(), folderdata)

 path\_results <- **file.path**(**getwd**(), folderresults)

 files <- **list.files**(path\_data)

 files <- **namesort**(files)

 *# Read the data*

 for (i in 1:nsim) {
 datafiles<-files[i]

 if (DL==FALSE) {

 Metaanalysis<-**read.table**(**file.path**(path\_data, datafiles), header=TRUE)

 J <-**nrow**(Metaanalysis)

 names <-**colnames**(Metaanalysis)

 Data <- **list**(nstudies=J, ev\_exp=Metaanalysis$ev\_exp, n\_exp=Metaanalysis$n\_exp, ev\_con=Metaanalysis$ev\_con, n\_con=Metaanalysis$n\_con)
 }

 *#' Delete trials with zero in both arms*

 Metaanalysis<-**read.table**(**file.path**(path\_data, datafiles), header=TRUE)

 dl.Metaanalysis<-**subset**(Metaanalysis, (ev\_exp == 0 & ev\_con == 0)==0)

 J <-**nrow**(dl.Metaanalysis)

 dl.names <-**colnames**(dl.Metaanalysis)

 Data <- **list**(nstudies=J, ev\_exp=dl.Metaanalysis$ev\_exp, n\_exp=dl.Metaanalysis$n\_exp, ev\_con=dl.Metaanalysis$ev\_con, n\_con=dl.Metaanalysis$n\_con)

 *#' Intract JAGS and R: Run the JAGS model in R*

 **system.time** ( jg.fit<- **jags**(data = Data,
 inits = initials,
 parameters.to.save = **c**('mu', 'pc'),
 model.file = modelfile,
 n.chains = 4,
 n.iter = n.iter,
 n.burnin = n.burnin,
 n.thin = 1,
 jags.seed = seed,
 progress.bar = "text") )

 *#' Results*

 jg.fit.sum<-jg.fit$BUGSoutput$summary

 RJAGS<-**as.data.frame**(jg.fit.sum, rownames=TRUE)

 Results<-**rbind**(Results, RJAGS["mu",**c**("mean", "sd", "2.5%", "50%", "97.5%", "Rhat", "n.eff")])

 *#' Remove unnecessary info from memory*

 **rm**(jg.fit, jg.fit.sum)

 *#' Garbage collection*

 **gc**()
 }

 *#' In case we need to round the result and save it*

 Result<-**round**(Results, 4)
 **write.table**(Result, file=**file.path**(path\_results, filename), col.names=T, row.names = F, sep="\t")
}

**Fnamesort.R**

## Sort the data files in ascending order

To order simulated data sequentially Two steps are necessary in order for this function to work which are done inside of all the other functions 1. After giving the name of the folder which meta-analysis data files are in (e.g. folderdata="OR1\_n10\_ratio1-4\_40p") we define respective path path\_data <- file.path(getwd(), folderdata) 2. We must list the files datafiles<-list.files(path\_data)

@description We name our simulated data using numbers and the default sorting of R recognizes them as a string and orders them in a mixed way. To be able to use our data sequentially we use function namesort.

@param datafiles the data-sets that we want to analyse @return namesort reads the data files and sorts them in numerical order

namesort <- function(datafiles){
 file <- **gsub**(".txt", "", datafiles)
 n <- **as.numeric**(file)
 files <- datafiles[**order**(n)]
}

**RERJAGS.R**

## Random effects Meta-analysis of rare events clinical trials

Using random effects JAGS models, function analysis the data-sets for meta-analysis of rare events clinical trials with binary outcome

@description RERJAGS reads the data-sets and using the random effects JAGS model (runs JAGS through R using jags() function of r2jags package) which we introduced in the args returns the result file in sequential order for each data-set and saves the log(OR) (mu), tau, standard deviations (sd), 95% credible intervals. n.effs and R.hats

@param folderdata It is the folder which contains the meta-analysis data files, folderresults The folder we want to save the result, nsim Number of meta-analysis data-sets that we want to analyse, DL If TRUE deletes studies with zero in both arm and analysis the data, n.iter number of iteration for MCMC machinery, n.burnin number of burnning in for MCMC, pars Parameters that we need for our meta-analysis model(s), initials the initial values for relative priors in the corresponding model(s), modelfile Text file of the model written in JAGS, filename Name of the file to save the result of the function.

@return Random effects RJAGS (RERJAGS) function reads the meta-analysis data-sets and based on the model we define and also the characteristics for MCMC machinery (check the arguments) of JAGS calculates the posterior information for odds ratio (mu=log(OR)) and heterogeity (tau)

RERJAGS<- function (folderdata, folderresults, nsim, DL, n.iter, n.burnin, n.chains, pars, initials, modelfile, filename.mu, filename.tau) {

 Resultsmu<-NULL
 Resultstau<-NULL

 *#' create result folder*
 **dir.create**(folderresults)

 *#' Read the generated data*

 path\_data <- **file.path**(**getwd**(), folderdata)

 path\_results <- **file.path**(**getwd**(), folderresults)

 files <- **list.files**(path\_data)

 files <- **namesort**(files)

 *# Read the data*

 for (i in 1:nsim) {
 datafiles<-files[i]
 if (!DL) {

 Metaanalysis<-**read.table**(**file.path**(path\_data, datafiles), header=TRUE)

 J <-**nrow**(Metaanalysis)

 names <-**colnames**(Metaanalysis)

 Data <- **list**(nstudies=J, ev\_exp=Metaanalysis$ev\_exp, n\_exp=Metaanalysis$n\_exp)
 }

 *#' Delete trials with zero in both arms*

 Metaanalysis<-**read.table**(**file.path**(path\_data, datafiles), header=TRUE)
 dl.Metaanalysis<-**subset**(Metaanalysis, (ev\_exp == 0 & ev\_con == 0)==0)

 J <-**nrow**(dl.Metaanalysis)

 dl.names <-**colnames**(dl.Metaanalysis)

 Data <- **list**(nstudies=J, ev\_exp=dl.Metaanalysis$ev\_exp, n\_exp=dl.Metaanalysis$n\_exp, ev\_con=dl.Metaanalysis$ev\_con, n\_con=dl.Metaanalysis$n\_con)

 *#' Intract JAGS and R: Run the JAGS model in R*

 **system.time** ( jg.fit<- **jags**(data = Data,
 inits = initials,
 parameters.to.save = pars,
 model.file = modelfile,
 n.chains = n.chains,
 n.iter = n.iter,
 n.burnin = n.burnin,
 n.thin = 1,
 jags.seed = seed,
 progress.bar = "text") )

 *#' Results*

 jg.fit.sum<-jg.fit$BUGSoutput$summary

 RJAGS<-**as.data.frame**(jg.fit.sum, rownames=TRUE)

 Resultsmu<-**rbind**(Resultsmu, RJAGS["mu",**c**("mean", "sd", "2.5%", "50%", "97.5%", "Rhat", "n.eff")])

 Resultstau<-**rbind**(Resultstau, RJAGS["tau",**c**("mean", "sd", "2.5%", "50%", "97.5%", "Rhat", "n.eff")])

 *#' Remove unnecessary info from memory*

 **rm**(jg.fit, jg.fit.sum)

 *#' Garbage collection*

 **gc**()
 }

 *#' In case we need to round the result and save it*

 Resultmu<-**round**(Resultsmu, 4)
 Resulttau<-**round**(Resultstau, 4)

 **write.table**(Resultmu, file=**file.path**(path\_results, filename.mu), col.names=T, row.names = F, sep="\t")
 **write.table**(Resulttau, file=**file.path**(path\_results, filename.tau), col.names=T, row.names = F, sep="\t")
}