```
####### Analysis Code #######
## Jip Ramakers, 13-3-2025 ##

# **Important note**: analyses are to be done within a network/country, because
# maturity groups (in maize) are country specific, and the first year of testing
# may not be construed similarly across countries (see Methods in paper)

# Specific models fitted per network can be found in the paper (eqns. 2–5),
# depending on the network (see suppl. tables S1 and S2 for models used for
# each network).


################################################################################
# This code contains examples of
# 1. Fitting GxE models and computing LSD% based on variance components. This
#    code reproduces (part of) the main results in the paper.
#
# 2. Combining countries to leverage the genetic correlation and improve LSD%
#    (using the pairwise LSD method in Piepho & Malik 2024, Plant Breeding).
#    This code reproduces the results in Fig. 4 of the paper.
################################################################################



#=============   1. Computing GxE variance components and LSD%   ==============#
setwd(...)

# Wheat data:
dw <- read.csv("wheatdat.csv")
dw[,1:5] <- lapply(dw[,1:5], factor)
head(dw)
  # network (country: AT-East, AT-West, BE, CZ, DE, FR-North, FR-South, HU, CH)
  # cultivarID
  # site
  # harvest year (as factor)
  # trial: subtrial, to be nested within site:year (site:year:trial) in
  #        Czech Rep., France (south and north), Germany and Hungary
  # first year of testing (based on registration dates for most cultivars),
  #        set to 0 for first harvest year within each network
  # trial year (harvest year, as numeric variable), set 0 for the first year.
  # grain yield (t/ha, at 0% moisture)

# Maize data:
dm <- read.csv("maizedat.csv")
dm[,1:6] <- lapply(dm[,1:6], factor)
head(dm)
  # network (country: AT, BE, CZ, DE, FR, HU, CH)
  # cultivarID
  # site
  # harvest year (as factor)
  # series: like subtrials, to be nested within site:year (site:year:trial) in
  #         Belgium, Germany and France
  # localmat: local maturity group (relevant for Austria, France ,Germany,
  #         Hungary andSwitzerland)
  # first year of testing (based on first year of appearance), set to 0
  #         for first harvest year within each network
  # trial year (harvest year, as numeric variable), set 0 for the first year
  # grain yield (t/ha, at 0% moisture)


## Wheat model (3) in the manuscript, e.g. for Germany.
library(asreml)
asreml.options(random.order="user")
```

```
dw.DE <- droplevels(subset(dw, network == "Germany"))

m3 <- asreml(yld ~ fYR + TRYR,
             random = ~(site + cul + year)^2 + site:year:trial,
             residual = ~units,
             data = dw.DE, workspace="2gb", maxiter=20)

summary(m3)$varcomp            # Variance components
summary(m3, coef=T)$coef.fixed # fixed effects
wald(m3, denDF="numeric", ssType="conditional")$Wald # Wald test

## Maize model (5) in the manuscript, e.g. for Germany:
dm.DE <- droplevels(subset(dm, network == "Germany"))
m5 <- asreml(yld ~ fYR + TRYR + localmat,
             random = ~(site + cul + year)^2 + site:year:series +
                       localmat:site + localmat:year + localmat:site:year,
             residual = ~units,
             data = dm.DE, workspace="2gb", maxiter=20)
summary(m5)$varcomp            # Variance components
summary(m5, coef=T)$coef.fixed # fixed effects
predict(m5,classify="Intercept")$pvals # intercept, averaged across maturity
groups
wald(m5, denDF="numeric", ssType="conditional")$Wald # Wald test


# The above models can be fitted in lme4 as well, e.g. for m5:
library(lmerTest)
m5.lme4 <- lmerTest::lmer(yld ~ fYR + TRYR + localmat +
                          (1|site)+(1|year)+ (1|site:year) +
                          (1|cul) + (1|cul:site) + (1|cul:year) +
                          (1|site:year:series) +
                          (1|localmat:site) +  (1|localmat:year)+
                          (1|localmat:site:year),
           data=dm.DE)
summary(m5.lme4) # uses Satherthwaite approximation of degrees of freedom as
                 # default (note that ddf="Kenward-Roger" can be very slow)


## Using variance components to compute LSD (eqns. 7–9) in paper ##

# 0.95 quantiles used to scale the LSDs as LSD%:
quants.w <- data.frame(Country = c("Austria - East","Austria - West","Belgium",
                                   "CzechRepublic","France - South","France -
North",
                                   "Germany","Hungary","Switzerland" ),
                       quant.95 =
c(8.054819,8.809469,9.432455,9.484489,7.954350,
                                    8.678204,9.269152,6.889400,7.094683))
quants.m <- data.frame(Country = c("Austria","Belgium","CzechRepublic","France",
                                   "Germany","Hungary","Switzerland"),
                       quant.95 = c(12.748814,11.500109,9.875063,12.050309,
                                    11.794340,10.366169,12.636150))

# LSD% for wheat in Germany:
vc <- summary(m3)$varcomp
nY <- 3; nL <- 17 # typical number of test years and locations (Table 1 in
paper)
Q <- quants.w[7,2] # 0.95 quantile of grain yield
Vu <- vc[2,1]/nL + vc[5,1]/nY + vc[8,1]/(nY*nL)
LSD <- 2 * sqrt(2*Vu)
(LSDperc <- LSD/Q*100)

 # --> nY and nL can be varied at will to reproduce Fig. 2 in paper
```

```
#================   2. Combining networks to improve LSD%   ==================#

# Using wheat as an example (for Germany and Czech Rep.), we combine GxE models
# into one (eqn. 10 in paper). Note that fYR needs to be re-computed.
# For the sake of brevity, we give the estimated variance components resulting
# from eqn. 10 here directly (complete code can be found in Peipho & Malik 2024,
# Plant Breeding, https://doi.org/10.1111/pbr.13243):

vc.DECZ <- read.table(text = "component std.error z.ratio
    cul:network!network!DE:!network!CZ.cor 0.874945 0.083582 10.46814
    cul:network!network_CZ 0.092425 0.013974 6.614022
    cul:network!network_DE 0.12281 0.013183 9.316037
    year:network!network!DE:!network!CZ.cor 0.679999 0.185736 3.661109
    year:network!network_CZ 0.528741 0.243335 2.172887
    year:network!network_DE 0.227013 0.093796 2.420279
    cul:year:network!network!DE:!network!CZ.cor 0.510027 0.260424 1.958445
    cul:year:network!network_CZ 0.050187 0.005073 9.893612
    cul:year:network!network_DE 0.020568 0.002128 9.663376
    site:network!network_CZ 0.359821 0.180227 1.996488
    site:network!network_DE 0.551331 0.120062 4.592034
    trial:year:site:network!network_CZ 0.15582 0.013292 11.72252
    trial:year:site:network!network_DE 0.079367 0.011093 7.154872
    year:site:network!network_CZ 1.365774 0.155406 8.788414
    year:site:network!network_DE 0.599432 0.044942 13.33786
    cul:site:network!network_CZ 0.02824 0.003233 8.73554
    cul:site:network!network_DE 0.014574 0.002346 6.212425
    network_CZ!R 0.20807 0.004315 48.22148
    network_DE!R 0.147374 0.002986 49.35235",
                header = TRUE, stringsAsFactors = FALSE)

# View variance components:
vc.DECZ

# Design setup of joint network:
nY  <- 3; nL <- 17; nR <- 2
gen.DE <- paste("G",  1:207, sep="") # 207 = 184 Unique + 23 common
gen.CZ <- paste("G", 185:327, sep="") # 143 = 120 Unique + 23 common
gen    <- unique(c(gen.DE, gen.CZ) )
n.gen.DE  <- length(gen.DE)
n.gen.CZ  <- length(gen.CZ)
n.gen     <- length(gen)

# Generate data using above network setup
  # (note that series/subtrials - or maturity groups in case of maize - can be
  # ignored here, as var. components in vc.DECZ have been adjusted accordingly)
data_DE <- expand.grid(network="DE", year=1:nY, site=1:nL, cul=gen.DE, rep=1:nR)
data_CZ <- expand.grid(network="CZ", year=1:nY, site=1:nL, cul=gen.CZ, rep=1:nR)
df <- rbind(data_DE, data_CZ)
df$y <- rnorm(nrow(df), mean = 0, sd = 1) # can be of any value
df[c(1:5)] <- lapply(df[c(1:5)], factor)
df$network <- factor(df$network, levels=c("CZ","DE"))

head(df)

# Fit model (10) and save starting values
m10 <- asreml(fixed  = y ~ 1 ,
              random = ~ id(cul):corgh(network)          +
                         id(year):corgh(network)         +
                         id(cul):id(year):corgh(network)  +
                         id(site):idh(network)            +
```

```
                        id(year):id(site):idh(network)    +
                        id(cul):id(site):idh(network)     ,
                residual  = ~ dsum(~id(units)|network)     ,
                data      = df,
                start.values = T)
start.values <- m10$vparameters.table

# Fix the starting values to emperical ones and fit model with or without a
# genetic correlation:
start.values$Value <- vc.DECZ$component[match(start.values$Component,
row.names(vc.DECZ))]
start.values$Constraint <- "F"
start.values2 <- start.values
start.values2[c(1,4,7),2] <- 0

# View start values:
start.values
start.values2

asreml.options(gammaPar = TRUE, Cfixed=TRUE, random.order="user")

m10.wGenCor <- asreml(fixed  = y ~ 1 ,
                      random = ~ id(cul):corgh(network)          +
                                 id(year):corgh(network)         +
                                 id(cul):id(year):corgh(network) +
                                 id(site):idh(network)           +
                                 id(year):id(site):idh(network)  +
                                 id(cul):id(site):idh(network),
                      residual  = ~ dsum(~id(units)|network),
                      data      = df,
                      G.param = start.values, R.param = start.values)
m10.woGenCor <- asreml(fixed  = y ~ 1 ,
                      random = ~ id(cul):corgh(network)          +
                                 id(year):corgh(network)         +
                                 id(cul):id(year):corgh(network) +
                                 id(site):idh(network)           +
                                 id(year):id(site):idh(network)  +
                                 id(cul):id(site):idh(network),
                      residual  = ~ dsum(~id(units)|network),
                      data      = df,
                      G.param = start.values2, R.param = start.values2)

## Function to retrieve variance of difference
VD <- function(model, network, pred.gen){
  library(data.table)
  n.gen <- length(pred.gen)
  cul <- model$G.param$`cul:network`$cul$levels
  g.c.pred   <- predict(model, classify = "cul:network",  sed=T,
                        levels = list(network=network, cul=pred.gen), only =
"cul:network")

  BLUPs.g.c        <- data.table(g.c.pred$pvals[ ,c(1:3)])
  names(BLUPs.g.c) <- c("cul", "network", "BLUP")
  BLUPs.g.c$cul    <- as.character(BLUPs.g.c$cul)

  #### Handle model estimates #

  ### G.g (i.e. estimated G matrix of genotype main effect)

  vc.g.c     <- summary(model)$varcomp[paste0('cul:network!
network_',network),'component']   # VC genotype main effect
  G.g.c.wide <- diag(1, n.gen)*vc.g.c
  dimnames(G.g.c.wide) <- list(cul, cul)           # G.g matrix
  G.g.c.long <- data.table(reshape2::melt(G.g.c.wide))
```

```
  names(G.g.c.long) <- c("cul1","cul2","sigma")
# G.g matrix in long format

  ### Variance of a difference between genotypic BLUPs (based on C22.g/PEV
matrix)
  vd.g.c.wide           <- g.c.pred$sed^2
  dimnames(vd.g.c.wide) <- list(cul, cul)                                  #
C22.g matrix
  vd.g.c.long        <- data.table(reshape2::melt(as.matrix(vd.g.c.wide)))
  names(vd.g.c.long) <- c("cul1","cul2","vd")                              #
C22.g matrix in long format

  ### merge BLUPs, G.g and C22.g information into "var.blup" table
  g.c.var <- G.g.c.long[cul1==cul2, .(cul1, sigma)]
  names(g.c.var) <- c("cul","var")                       # variances G.g
  g.c.cov        <- G.g.c.long[cul1!=cul2]
  names(g.c.cov) <- c("cul1","cul2","cov")               # covariances of G.g
  var.blup       <- merge(vd.g.c.long, g.c.cov, all=T)

  for (i in 1:2){
    temp <- merge(BLUPs.g.c, g.c.var, by="cul")
    names(temp) <- c(paste0(names(temp),i))               # merge BLUPs and
variances for each genotye
    var.blup  <- merge(var.blup, temp, by=paste0("cul",i)) # merge this for both
cul1 and cul2, respectively, to result table
  }

  ### formatting
  setcolorder(var.blup,
c("cul1","BLUP1","cul2","BLUP2","var1","var2","cov","vd"))
  var.blup <- var.blup[order(cul1, cul2)]
  var.blup[, i.is.j     := cul1==cul2]                         # i=j is not a
pair
  var.blup[, i.larger.j := as.numeric(substring(cul1, 2)) >
             as.numeric(substring(cul2, 2))] # i>j is a duplicate pair


  # Return objects
  return(list(g.pred = g.c.pred, blup= BLUPs.g.c, G.g.c.long=G.g.c.long,
              vd.g.c.long=vd.g.c.long))
}


## Prediction in DE: comparison i, iii, v  and vii in Fig. 4
 # Prediction in CZ: comparison ii, iv, vi and viii "  "   "
vd.DE_woGenCor <- VD(model=m10.woGenCor, network="DE", pred.gen = gen)
vd.DE_wGenCor <-  VD(model=m10.wGenCor,  network="DE", pred.gen = gen)
vd.CZ_woGenCor <- VD(model=m10.woGenCor, network="CZ", pred.gen = gen)
vd.CZ_wGenCor <-  VD(model=m10.wGenCor,  network="CZ", pred.gen = gen)

## Retrieve variances of differences below:

###  1: NO genetic correlation between networks #####

# (i) two genotypes tested in both countries A and B, focus on comparison in
country A
comp.i.woGenCor <- vd.DE_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))>=
185
                                    & as.numeric(substring(cul1, 2)) <= 207,]
comp.i.woGenCor <-       comp.i.woGenCor[as.numeric(substring(cul2, 2)) >= 185 &
                                           as.numeric(substring(cul2, 2)) <=
207,]
# (ii) two genotypes tested in both countries A and B, focus on comparison in
country B
```

```
comp.ii.woGenCor <- vd.CZ_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))>=
185
                                               & as.numeric(substring(cul1, 2))
<= 207,]
comp.ii.woGenCor <-       comp.ii.woGenCor[as.numeric(substring(cul2, 2)) >= 185
&
                                           as.numeric(substring(cul2, 2)) <=
207,]
# (iii) two genotypes tested only in country A, focus on their comparison in
country A
comp.iii.woGenCor <- vd.DE_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))
<= 184,]
comp.iii.woGenCor <- comp.iii.woGenCor[as.numeric(substring(cul2, 2)) <= 184,]
# (iv) two genotypes tested only in country A, focus on their comparison in
country B
comp.iv.woGenCor <- vd.CZ_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
184,]
comp.iv.woGenCor <- comp.iv.woGenCor[as.numeric(substring(cul2, 2)) <= 184,]
# (v) two genotypes tested only in country B, focus on their comparison in
country A
comp.v.woGenCor <- vd.DE_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) >=
208,]
comp.v.woGenCor <- comp.v.woGenCor[as.numeric(substring(cul2, 2)) >= 208,]
# (vi) two genotypes tested only in country B, focus on their comparison in
country B
comp.vi.woGenCor <- vd.CZ_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) >=
208,]
comp.vi.woGenCor <- comp.vi.woGenCor[as.numeric(substring(cul2, 2)) >= 208,]
# (vii) one genotype tested only in country A, one only in country B, focus on
their comparison in country A
comp.vii.woGenCor <- vd.DE_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))
<= 184,]
comp.vii.woGenCor <- comp.vii.woGenCor[as.numeric(substring(cul2, 2)) >= 208,]
# (viii) one genotype tested only in country A, one only in country B, focus on
their comparison in country B
comp.viii.woGenCor <- vd.CZ_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))
<= 184,]
comp.viii.woGenCor <- comp.viii.woGenCor[as.numeric(substring(cul2, 2)) >= 208,]

# Get mean variance of a difference and express as LSD%
LSD.i    <- (2 * sqrt(mean(comp.i.woGenCor$vd,na.rm=T)))   / quants.w[7,2] * 100
LSD.ii   <- (2 * sqrt(mean(comp.ii.woGenCor$vd,na.rm=T)))  / quants.w[4,2] * 100
LSD.iii  <- (2 * sqrt(mean(comp.iii.woGenCor$vd,na.rm=T))) / quants.w[7,2] * 100
LSD.iv   <- (2 * sqrt(mean(comp.iv.woGenCor$vd,na.rm=T)))  / quants.w[4,2] * 100
LSD.v    <- (2 * sqrt(mean(comp.v.woGenCor$vd,na.rm=T)))   / quants.w[7,2] * 100
LSD.vi   <- (2 * sqrt(mean(comp.vi.woGenCor$vd,na.rm=T)))  / quants.w[4,2] * 100
LSD.vii  <- (2 * sqrt(mean(comp.vii.woGenCor$vd,na.rm=T))) / quants.w[7,2] * 100
LSD.viii <- (2 * sqrt(mean(comp.viii.woGenCor$vd,na.rm=T)))/ quants.w[4,2] * 100

# Save LSDs
LSD.woGenCor <- cbind(LSD.i,LSD.ii,LSD.iii,LSD.iv,LSD.v,LSD.vi,LSD.vii,LSD.viii)


      #### 2: WITH a genetic correlation between networks ####

# (i) two genotypes tested in both countries A and B, focus on comparison in
country A
comp.i.wGenCor <- vd.DE_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))>= 185
                                               & as.numeric(substring(cul1, 2))
<= 207,]
comp.i.wGenCor <-       comp.i.wGenCor[as.numeric(substring(cul2, 2)) >= 185 &
                                           as.numeric(substring(cul2, 2)) <=
207,]
# (ii) two genotypes tested in both countries A and B, focus on comparison in
```

```
country B
comp.ii.wGenCor <- vd.CZ_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))>=
185
                                                & as.numeric(substring(cul1, 2))
<= 207,]
comp.ii.wGenCor <-       comp.ii.wGenCor[as.numeric(substring(cul2, 2)) >= 185 &
                                           as.numeric(substring(cul2, 2)) <=
207,]
# (iii) two genotypes tested only in country A, focus on their comparison in
country A
comp.iii.wGenCor <- vd.DE_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
184,]
comp.iii.wGenCor <- comp.iii.wGenCor[as.numeric(substring(cul2, 2)) <= 184,]
# (iv) two genotypes tested only in country A, focus on their comparison in
country B
comp.iv.wGenCor <- vd.CZ_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
184,]
comp.iv.wGenCor <- comp.iv.wGenCor[as.numeric(substring(cul2, 2)) <= 184,]
# (v) two genotypes tested only in country B, focus on their comparison in
country A
comp.v.wGenCor <- vd.DE_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) >=
208,]
comp.v.wGenCor <- comp.v.wGenCor[as.numeric(substring(cul2, 2)) >= 208,]
# (vi) two genotypes tested only in country B, focus on their comparison in
country B
comp.vi.wGenCor <- vd.CZ_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) >=
208,]
comp.vi.wGenCor <- comp.vi.wGenCor[as.numeric(substring(cul2, 2)) >= 208,]
# (vii) one genotype tested only in country A, one only in country B, focus on
their comparison in country A
comp.vii.wGenCor <- vd.DE_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
184,]
comp.vii.wGenCor <- comp.vii.wGenCor[as.numeric(substring(cul2, 2)) >= 208,]
# (viii) one genotype tested only in country A, one only in country B, focus on
their comparison in country B
comp.viii.wGenCor <- vd.CZ_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
184,]
comp.viii.wGenCor <- comp.viii.wGenCor[as.numeric(substring(cul2, 2)) >= 208,]

# Get mean variance of a difference and express as LSD%
LSD.i    <- (2 * sqrt(mean(comp.i.wGenCor$vd,na.rm=T)))   / quants.w[7,2] * 100
LSD.ii   <- (2 * sqrt(mean(comp.ii.wGenCor$vd,na.rm=T)))  / quants.w[4,2] * 100
LSD.iii  <- (2 * sqrt(mean(comp.iii.wGenCor$vd,na.rm=T))) / quants.w[7,2] * 100
LSD.iv   <- (2 * sqrt(mean(comp.iv.wGenCor$vd,na.rm=T)))  / quants.w[4,2] * 100
LSD.v    <- (2 * sqrt(mean(comp.v.wGenCor$vd,na.rm=T)))   / quants.w[7,2] * 100
LSD.vi   <- (2 * sqrt(mean(comp.vi.wGenCor$vd,na.rm=T)))  / quants.w[4,2] * 100
LSD.vii  <- (2 * sqrt(mean(comp.vii.wGenCor$vd,na.rm=T))) / quants.w[7,2] * 100
LSD.viii <- (2 * sqrt(mean(comp.viii.wGenCor$vd,na.rm=T)))/ quants.w[4,2] * 100

# Save LSDs
LSD.wGenCor <- cbind(LSD.i,LSD.ii,LSD.iii,LSD.iv,LSD.v,LSD.vi,LSD.vii,LSD.viii)


## Display LSDs (%) without and with genetic correlation
# (see Methods for interpretation):
round(rbind(LSD.woGenCor, LSD.wGenCor),3)




# Using MAIZE as an example (for Germany and France), we combine GxE models
# into one (eqn. 10 in paper). Note that fYR needs to be re-computed.
# For the sake of brevity, we give the estimated variance components resulting
# from eqn. 10 here:
```

```
vc.DEFR <- read.table(text = "component
                              cul:network!network!DE:!network!FR.cor   0.7719
                              cul:network!network_FR 0.0845
                              cul:network!network_DE 0.0637
                              year:network!network!DE:!network!FR.cor  0.3138
                              year:network!network_FR0.1558
                              year:network!network_DE0.767
                              cul:year:network!network!DE:!network!FR.cor
      0.6932
                              cul:year:network!network_FR  0.03
                              cul:year:network!network_DE  0.0373
                              site:network!network_FR0.2092
                              site:network!network_DE0.5789
                              year:site:network!network_FR 0.674
                              year:site:network!network_DE 1.0941
                              cul:site:network!network_FR  0.0434
                              cul:site:network!network_DE  0.0359
                              network_FR!R     0.1559
                              network_DE!R     0.2617",
                       header = TRUE, stringsAsFactors = FALSE)

# View variance components:
vc.DEFR

# Design setup of joint network:
nY  <- 2; nL <- 12; nR <- 2
gen.DE <- paste("G",  1:176, sep="") # 176 = 145 Unique + 31 common
gen.FR <- paste("G", 146:832, sep="") # 687= 656 Unique + 31 common
gen    <- unique(c(gen.DE, gen.FR) )
n.gen.DE  <- length(gen.DE)
n.gen.FR  <- length(gen.FR)
n.gen     <- length(gen)

# Generate data using above network setup
# (note that series/subtrials - or maturity groups in case of maize - can be
# ignored here, as var. components in vc.DEFR have been adjusted accordingly)
data_DE <- expand.grid(network="DE", year=1:nY, site=1:nL, cul=gen.DE, rep=1:nR)
data_FR <- expand.grid(network="FR", year=1:nY, site=1:nL, cul=gen.FR, rep=1:nR)
df <- rbind(data_DE, data_FR)
df$y <- rnorm(nrow(df), mean = 0, sd = 1) # can be of any value
df[c(1:5)] <- lapply(df[c(1:5)], factor)
df$network <- factor(df$network, levels=c("FR","DE"))

head(df)

# Fit model (10) and save starting values
m10 <- asreml(fixed  = y ~ 1 ,
              random = ~ id(cul):corgh(network)          +
                id(year):corgh(network)          +
                id(cul):id(year):corgh(network)  +
                id(site):idh(network)             +
                id(year):id(site):idh(network)    +
                id(cul):id(site):idh(network)     ,
              residual  = ~ dsum(~id(units)|network)     ,
              data      = df,
              start.values = T)
start.values <- m10$vparameters.table

# Fix the starting values to emperical ones and fit model with or without a
# genetic correlation:
start.values$Value <- vc.DEFR$component[match(start.values$Component,
row.names(vc.DEFR))]
start.values$Constraint <- "F"
start.values2 <- start.values
```

```
start.values2[c(1,4,7),2] <- 0

# View start values:
start.values
start.values2

asreml.options(gammaPar = TRUE, Cfixed=TRUE, random.order="user")

m10.wGenCor <- asreml(fixed  = y ~ 1 ,
                      random = ~ id(cul):corgh(network)          +
                        id(year):corgh(network)         +
                        id(cul):id(year):corgh(network) +
                        id(site):idh(network)           +
                        id(year):id(site):idh(network)  +
                        id(cul):id(site):idh(network),
                      residual  = ~ dsum(~id(units)|network),
                      data      = df,
                      G.param = start.values, R.param = start.values)
m10.woGenCor <- asreml(fixed  = y ~ 1 ,
                       random = ~ id(cul):corgh(network)          +
                         id(year):corgh(network)         +
                         id(cul):id(year):corgh(network) +
                         id(site):idh(network)           +
                         id(year):id(site):idh(network)  +
                         id(cul):id(site):idh(network),
                       residual  = ~ dsum(~id(units)|network),
                       data      = df,
                       G.param = start.values2, R.param = start.values2)

## Prediction in DE: comparison i, iii, v  and vii in Fig. 4
# Prediction in FR: comparison ii, iv, vi and viii "   "   "
vd.DE_woGenCor <- VD(model=m10.woGenCor, network="DE", pred.gen = gen)
vd.DE_wGenCor <-  VD(model=m10.wGenCor,  network="DE", pred.gen = gen)
vd.FR_woGenCor <- VD(model=m10.woGenCor, network="FR", pred.gen = gen)
vd.FR_wGenCor <-  VD(model=m10.wGenCor,  network="FR", pred.gen = gen)

## Retrieve variances of differences below:

###  1: NO genetic correlation between networks #####

# (i) two genotypes tested in both countries A and B, focus on comparison in
country A
comp.i.woGenCor <- vd.DE_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))>=
146
                                              & as.numeric(substring(cul1, 2))
<= 176,]
comp.i.woGenCor <-       comp.i.woGenCor[as.numeric(substring(cul2, 2)) >= 146 &
                                           as.numeric(substring(cul2, 2)) <=
167,]
# (ii) two genotypes tested in both countries A and B, focus on comparison in
country B
comp.ii.woGenCor <- vd.FR_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))>=
146
                                               & as.numeric(substring(cul1, 2))
<= 176,]
comp.ii.woGenCor <-       comp.ii.woGenCor[as.numeric(substring(cul2, 2)) >= 146
&
                                            as.numeric(substring(cul2, 2)) <=
176,]
# (iii) two genotypes tested only in country A, focus on their comparison in
country A
comp.iii.woGenCor <- vd.DE_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))
<= 145,]
comp.iii.woGenCor <- comp.iii.woGenCor[as.numeric(substring(cul2, 2)) <= 145,]
```

```
# (iv) two genotypes tested only in country A, focus on their comparison in
country B
comp.iv.woGenCor <- vd.FR_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
145,]
comp.iv.woGenCor <- comp.iv.woGenCor[as.numeric(substring(cul2, 2)) <= 145,]
# (v) two genotypes tested only in country B, focus on their comparison in
country A
comp.v.woGenCor <- vd.DE_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) >=
177,]
comp.v.woGenCor <- comp.v.woGenCor[as.numeric(substring(cul2, 2)) >= 177,]
# (vi) two genotypes tested only in country B, focus on their comparison in
country B
comp.vi.woGenCor <- vd.FR_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) >=
177,]
comp.vi.woGenCor <- comp.vi.woGenCor[as.numeric(substring(cul2, 2)) >= 177,]
# (vii) one genotype tested only in country A, one only in country B, focus on
their comparison in country A
comp.vii.woGenCor <- vd.DE_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))
<= 145,]
comp.vii.woGenCor <- comp.vii.woGenCor[as.numeric(substring(cul2, 2)) >= 177,]
# (viii) one genotype tested only in country A, one only in country B, focus on
their comparison in country B
comp.viii.woGenCor <- vd.FR_woGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))
<= 145,]
comp.viii.woGenCor <- comp.viii.woGenCor[as.numeric(substring(cul2, 2)) >= 177,]

# Get mean variance of a difference and express as LSD%
LSD.i    <- (2 * sqrt(mean(comp.i.woGenCor$vd,na.rm=T)))   / quants.m[5,2] * 100
LSD.ii   <- (2 * sqrt(mean(comp.ii.woGenCor$vd,na.rm=T)))  / quants.m[4,2] * 100
LSD.iii  <- (2 * sqrt(mean(comp.iii.woGenCor$vd,na.rm=T))) / quants.m[5,2] * 100
LSD.iv   <- (2 * sqrt(mean(comp.iv.woGenCor$vd,na.rm=T)))  / quants.m[4,2] * 100
LSD.v    <- (2 * sqrt(mean(comp.v.woGenCor$vd,na.rm=T)))   / quants.m[5,2] * 100
LSD.vi   <- (2 * sqrt(mean(comp.vi.woGenCor$vd,na.rm=T)))  / quants.m[4,2] * 100
LSD.vii  <- (2 * sqrt(mean(comp.vii.woGenCor$vd,na.rm=T))) / quants.m[5,2] * 100
LSD.viii <- (2 * sqrt(mean(comp.viii.woGenCor$vd,na.rm=T)))/ quants.m[4,2] * 100

# Save LSDs
LSD.woGenCor <- cbind(LSD.i,LSD.ii,LSD.iii,LSD.iv,LSD.v,LSD.vi,LSD.vii,LSD.viii)


#### 2: WITH a genetic correlation between networks ####

# (i) two genotypes tested in both countries A and B, focus on comparison in
country A
comp.i.wGenCor <- vd.DE_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))>= 146
                                            & as.numeric(substring(cul1, 2)) <=
176,]
comp.i.wGenCor <-       comp.i.wGenCor[as.numeric(substring(cul2, 2)) >= 146 &
                                       as.numeric(substring(cul2, 2)) <= 167,]
# (ii) two genotypes tested in both countries A and B, focus on comparison in
country B
comp.ii.wGenCor <- vd.FR_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2))>=
146
                                             & as.numeric(substring(cul1, 2)) <=
176,]
comp.ii.wGenCor <-       comp.ii.wGenCor[as.numeric(substring(cul2, 2)) >= 146 &
                                         as.numeric(substring(cul2, 2)) <=
176,]
# (iii) two genotypes tested only in country A, focus on their comparison in
country A
comp.iii.wGenCor <- vd.DE_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
145,]
comp.iii.wGenCor <- comp.iii.wGenCor[as.numeric(substring(cul2, 2)) <= 145,]
# (iv) two genotypes tested only in country A, focus on their comparison in
```

```
country B
comp.iv.wGenCor <- vd.FR_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
145,]
comp.iv.wGenCor <- comp.iv.wGenCor[as.numeric(substring(cul2, 2)) <= 145,]
# (v) two genotypes tested only in country B, focus on their comparison in
country A
comp.v.wGenCor <- vd.DE_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) >=
177,]
comp.v.wGenCor <- comp.v.wGenCor[as.numeric(substring(cul2, 2)) >= 177,]
# (vi) two genotypes tested only in country B, focus on their comparison in
country B
comp.vi.wGenCor <- vd.FR_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) >=
177,]
comp.vi.wGenCor <- comp.vi.wGenCor[as.numeric(substring(cul2, 2)) >= 177,]
# (vii) one genotype tested only in country A, one only in country B, focus on
their comparison in country A
comp.vii.wGenCor <- vd.DE_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
145,]
comp.vii.wGenCor <- comp.vii.wGenCor[as.numeric(substring(cul2, 2)) >= 177,]
# (viii) one genotype tested only in country A, one only in country B, focus on
their comparison in country B
comp.viii.wGenCor <- vd.FR_wGenCor$vd.g.c.long[as.numeric(substring(cul1, 2)) <=
145,]
comp.viii.wGenCor <- comp.viii.wGenCor[as.numeric(substring(cul2, 2)) >= 177,]

# Get mean variance of a difference and express as LSD%
LSD.i    <- (2 * sqrt(mean(comp.i.wGenCor$vd,na.rm=T)))   / quants.m[5,2] * 100
LSD.ii   <- (2 * sqrt(mean(comp.ii.wGenCor$vd,na.rm=T)))  / quants.m[4,2] * 100
LSD.iii  <- (2 * sqrt(mean(comp.iii.wGenCor$vd,na.rm=T))) / quants.m[5,2] * 100
LSD.iv   <- (2 * sqrt(mean(comp.iv.wGenCor$vd,na.rm=T)))  / quants.m[4,2] * 100
LSD.v    <- (2 * sqrt(mean(comp.v.wGenCor$vd,na.rm=T)))   / quants.m[5,2] * 100
LSD.vi   <- (2 * sqrt(mean(comp.vi.wGenCor$vd,na.rm=T)))  / quants.m[4,2] * 100
LSD.vii  <- (2 * sqrt(mean(comp.vii.wGenCor$vd,na.rm=T))) / quants.m[5,2] * 100
LSD.viii <- (2 * sqrt(mean(comp.viii.wGenCor$vd,na.rm=T)))/ quants.m[4,2] * 100
# Save LSDs
LSD.wGenCor <- cbind(LSD.i,LSD.ii,LSD.iii,LSD.iv,LSD.v,LSD.vi,LSD.vii,LSD.viii)


## Display LSDs (%) without and with genetic correlation
# (see Methods for interpretation):
round(rbind(LSD.woGenCor, LSD.wGenCor),3)
```