

Code summary (Eng): models and PEVD

Eva Lavrencic

Contents

1	IMPORTING DATA	1
2	MODEL DESCRIPTION	2
2.1	SPATIAL EFFECT MODELLING	2
3	HYPOTHESES	2
4	DATA PREPARATION	3
4.1	REGIONAL NEIGHBORHOOD MATRIX (accounting for neighborhood structure in BESAG/CAR model)	3
4.2	DEFINING PRIORS	3
4.3	PREPARING SUBSET OF DATA	3
4.4	SPDE MODEL PREPARATION (STACK) (Maria)	4
4.5	SET MODEL FORMULAS	6
5	FIT MODELS	6
6	MODEL COMPARISON (VARIANCE COMPONENTS, DIC VALUES)	7
7	INLA POSTERIOR SAMPLING	8

1 IMPORTING DATA

```
# load("import_data.RData")
#
# load("~/UPORABNA STATISTIKA/Spatial Modelling/SM/SlovenianCattleAnalysis/SlovenianCattleAnalysis/Proc
load("workspaces/import_data.RData")
load("workspaces/model_comparison.RData")
```

- phenoData = original dataset, 30314x30, data on Brown-Swiss cattle from Slovenia (years 2004-2019 from 2012 different herds)
- Ainv = inverse of the relationship matrix (56465x56465)

- boundaryOB = map of municipalities (sf data.frame class)
- boundaryUE = map of administrative units (sf data.frame class)
- mapOB = map of municipalities (SpatialPolygonsDataFrame class)
- mapUE = map of administrative units (SpatialPolygonsDataFrame class)
- sloveniaBoundary = map of Slovenia

```
# importing results
# load("workspaces/full_models.RData")
# load("workspaces/Eucl_models.RData")
# load("workspaces/spde_big.RData")
# load("workspaces/spde_small.RData")
# load("workspaces/UE_models.RData")
# load("workspaces/mc_samples.RData")
# load("workspaces/small_models_with_MC.RData")
# Eucl_dist <- read.table("workspaces/Euclidean_dist_small.txt", sep=" ")
# Eucl_dist <- data.matrix(Eucl_dist)
```

2 MODEL DESCRIPTION

We analysed a physical trait describing a body confirmation measure as a response variable (last004_scaled). In all models we included 3 fixed effects: year and scorer of the data (vpliv1), cattle age and stage of lactation (vpliv2) and year and month of calving (vpliv3). We included 3 fixed effects: genetic/pedigree effect (G), herd effect (H) and herd location effect (S) in different combinations (see below):

$$G: y_i = x_i\beta + a_i + e_i;$$

$$GH: y_i = x_i\beta + a_i + h_i + e_i;$$

$$GS: y_i = x_i\beta + a_i + s_i + e_i;$$

$$GHS: y_i = x_i\beta + a_i + h_i + s_i + e_i$$

- β = vector of fixed covariate effects, x_i = covariate vector
- a_i = additive genetic effects (breeding value); modelled using pedigree relationship matrix A
- h_i = herd effect; modelled using herd ids
- s_i = spatial effect; modelled in 3 different ways
- e_i = residual effect

2.1 SPATIAL EFFECT MODELLING

- BESAG/CAR (regional model)
- using Euclidean distances covariance matrix (geostatistical model)
- SPDE (geostatical model)

BESAG/CAR and model based on Euclidean distances were fitted in INLA-R and BLUPF90. SPDE model was fitted in INLA-R.

3 HYPOTHESES

- Accounting for spatial relationships can improve genetic evaluation [low genetic connectedness between small herds, relatively high environmental variation setting]
- Accounting for spatial relationships can improve genetic connectedness between animals

4 DATA PREPARATION

4.1 REGIONAL NEIGHBORHOOD MATRIX (accounting for neighborhood structure in BESAG/CAR model)

Neighborhood matrix can be constructed using 'inla.read.graph' function. Construted graph contains information on number and ids of neighbouring regions for every region. Then, the region id (UE_MID and OB_MID) in the phenoData dataset is merged with the consecutive id number of the region (1 to number of regions).

```
# administrative units
nb.mapUE <- poly2nb(mapUE) #class nb
nb2INLA("map.graphUE", nb.mapUE)
gUE <- inla.read.graph(filename = "map.graphUE")

#obcine
nb.mapOB <- poly2nb(mapOB)
nb2INLA("map.graphOB", nb.mapOB)
gOB <- inla.read.graph(filename = "map.graphOB")

# assinging a sequence number to regional id
grafID_UE <- data.frame("UE" = mapUE[, "UE_MID"], "idUE" = mapUE[, "UE_ID"])
names(grafID_UE) <- c("UE", "idUE")
grafID_OB <- data.frame("OB" = mapOB[, "OB_MID"], "idOB" = mapOB[, "OB_ID"])
names(grafID_OB) <- c("OB", "idOB")

phenoData <- as.data.frame(merge(phenoData, grafID_UE, all.x=T))

## Warning in '[<-factor'('*tmp*', ri, value = c(10051851L, 10051860L,
## 10051878L, : invalid factor level, NA generated

phenoData <- as.data.frame(merge(phenoData, grafID_OB, all.x=T))

## Warning in '[<-factor'('*tmp*', ri, value = c(11026923L, 11027776L,
## 11027997L, : invalid factor level, NA generated
```

4.2 DEFINING PRIORS

```
# priors
hyperVarPed = list(theta = list(prior="pc.prec", param=c(0.1,0.5)))
hyperVarIdlok.GH = list(theta = list(prior="pc.prec", param=c(0.25,0.5)))
hyperVarIdlok.GHS = list(theta = list(prior="pc.prec", param=c(0.15,0.5)))
hyperResVarG = list(theta = list(prior="pc.prec", param=c(0.3,0.5)))
hyperResVarGHS = list(theta = list(prior="pc.prec", param=c(0.15,0.5)))
```

4.3 PREPARING SUBSET OF DATA

- n = 3800

- sampling 3800 individuals without replacement, with sampling probability equal to the inverse herd size

```
# podset podatkov
herd_n <- data.frame(table(phenoData$idlok))
colnames(herd_n) <- c("idlok", "herd_n")
phenoData <- merge(phenoData, herd_n, all.x=T)
set.seed(85940)
phenoData.small <- phenoData[sample(1:nrow(phenoData), size = 3800, replace = FALSE, prob = 1/phenoData$herd_n), ]
phenoData.small <- drop.levels(phenoData.small)
```

4.4 SPDE MODEL PREPARATION (STACK) (Maria)

Full data.

```
# Prior values
hyperRange = c(50, 0.8) # <
hyperVarSpdeS = c(1, 0.05) # >
hyperVarSpdeHS = c(0.6, 0.05) # >

hyperVarPed = list(theta = list(prior="pc.prec", param=c(0.4,0.05)))
hyperVarIdlokH = list(theta = list(prior="pc.prec", param=c(1,0.05)))
hyperVarIdlokHS = list(theta = list(prior="pc.prec", param=c(0.6,0.05)))

hyperResVarG = list(theta = list(prior="pc.prec", param=c(1,0.1)))
hyperResVarRest = list(theta = list(prior="pc.prec", param=c(0.6,0.05)))

# Use model matrix to fit factor effects
M = model.matrix(~1 + vpliv1 + vpliv2 + vpliv3, phenoData)
# Store in a stack
FactorNames = tail(colnames(M), ncol(M)-1)
FactorStack = paste0("list(", FactorNames, " = M[,", 2:(ncol(M)), "]),", collapse = " ")
FactorStack = (gsub('.{1}$', '', FactorStack))
FactorStack = paste0("c(", FactorStack, ")")

# Make the formula
FormulaStack = paste(" + ", FactorNames, collapse = " ")

formulaS.1 = 'last004scaled ~ intercept + f(rowNumberAinv, model = "generic0", Cmatrix = Ainv, hyper = hyperRange, hyperVar = hyperVarSpdeS)'
formulaHS.1 = 'last004scaled ~ intercept + f(rowNumberAinv, model = "generic0", Cmatrix = Ainv, hyper = hyperRange, hyperVar = hyperVarSpdeHS)'

formulaS = as.formula(paste0(formulaS.1, FormulaStack))
formulaHS = as.formula(paste0(formulaHS.1, FormulaStack))

# Make mesh and SPDE
mesh = inla.mesh.2d(cbind(phenoData$xpos, phenoData$ypos), max.edge=c(10, 20), cutoff = 2.5, offset = 3)
A = inla.spde.make.A(mesh = mesh, loc = cbind(phenoData$xpos, phenoData$ypos))
spdeStatS = inla.spde2.pcmatern(mesh = mesh, alpha = 2, prior.range = hyperRange, prior.sigma = hyperVarSpdeS)
meshIndexS = inla.spde.make.index(name = "fieldID", n.spde = spdeStatS$n.spde)
spdeStatHS = inla.spde2.pcmatern(mesh = mesh, alpha = 2, prior.range = hyperRange, prior.sigma = hyperVarSpdeHS)
meshIndexHS = inla.spde.make.index(name = "fieldID", n.spde = spdeStatHS$n.spde)
```

```

# Make stack
stack = inla.stack(data = list(last004scaled = phenoData$last004scaled),
  A = list(A,1, 1),
  effects = list(c(meshIndexS, list(intercept = 1)),
    c(list(rowNumberAinv = phenoData$rowNumberAinv),list(idlok = phenoData$
    c(eval(parse(text = FactorStack ) ) )), tag = "est")

stackHS = inla.stack(data = list(last004scaled = phenoData$last004scaled),
  A = list(A,1, 1),
  effects = list(c(meshIndexHS, list(intercept = 1)),
    c(list(rowNumberAinv = phenoData$rowNumberAinv),list(idlok = phenoData$
    c(eval(parse(text = FactorStack ) ) ) ), tag = "est")

```

Data subset.

```

# Use model matrix to fit factor effects
M.small = model.matrix(~1 + vpliv1 + vpliv2 + vpliv3, phenoData.small)
# Store in a stack
FactorNames.small = tail(colnames(M.small), ncol(M.small)-1)
FactorStack.small = paste0( "list(", FactorNames.small, " = M.small[,", 2:(ncol(M.small)), "]),", collapse = ",")
FactorStack.small = ( gsub('.{1}$', '', FactorStack.small ) )
FactorStack.small = paste0("c(", FactorStack.small, ")")

# Make the formula
FormulaStack.small = paste("+", FactorNames.small, collapse = " ")

formulaS.1.small = 'last004scaled ~ intercept + f(rowNumberAinv, model = "generic0", Cmatrix = Ainv, hyper = 1)'
formulaHS.1.small = 'last004scaled ~ intercept + f(rowNumberAinv, model = "generic0", Cmatrix = Ainv, hyper = 1)'

formulaS.small = as.formula( paste0(formulaS.1.small, FormulaStack.small))
formulaHS.small = as.formula( paste0(formulaHS.1.small, FormulaStack.small))

# Make mesh and SPDE
mesh.small = inla.mesh.2d(cbind(phenoData.small$xpos, phenoData.small$ypos), max.edge=c(10, 20), cutoff=100)
A.small = inla.spde.make.A(mesh = mesh.small, loc = cbind(phenoData.small$xpos, phenoData.small$ypos) )
spdeStatS.small = inla.spde2.pcmatern(mesh = mesh.small, alpha = 2 , prior.range = hyperRange, prior.var = 1)
meshIndexS.small = inla.spde.make.index(name = "fieldID", n.spde = spdeStatS.small$n.spde)
spdeStatHS.small = inla.spde2.pcmatern(mesh = mesh.small, alpha = 2 , prior.range = hyperRange, prior.var = 1)
meshIndexHS.small = inla.spde.make.index(name = "fieldID", n.spde = spdeStatHS.small$n.spde)

# Make stack
stack.small = inla.stack(data = list(last004scaled = phenoData.small$last004scaled),
  A = list(A.small,1, 1),
  effects = list(c(meshIndexS.small, list(intercept = 1)),
    c(list(rowNumberAinv = phenoData.small$rowNumberAinv),list(idlok = phenoData$
    c(eval(parse(text = FactorStack.small ) ) )), tag = "est")

stackHS.small = inla.stack(data = list(last004scaled = phenoData.small$last004scaled),
  A = list(A.small,1, 1),
  effects = list(c(meshIndexHS.small, list(intercept = 1)),
    c(list(rowNumberAinv = phenoData.small$rowNumberAinv),list(idlok = phenoData$
    c(eval(parse(text = FactorStack.small ) ) ) ), tag = "est")

```

4.5 SET MODEL FORMULAS

Full dataset models.

```
# G MODEL (G)
formulaB.G = last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "generic0", Cmat:
# H MODEL (G + H: herd effect)
formulaB.H = last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "generic0", Cmat:
# S MODEL (G + field (regional effect)) BESAG
# UE = administrative regions
# OB = municipalities
formulaB.S.UE = last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "generic0", C
formulaB.S.OB = last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "generic0", C
# HS MODEL (G + location + field) BESAG
formulaB.HS.UE <- last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "generic0", C
formulaB.HS.OB <- last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "generic0", C

# Euclidean
formulaB.S.Eucl = last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "generic0",
formulaB.HS.Eucl <- last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "generic0",
```

Data subset models.

```
# Euclidean
formulaB.S.Eucl.small = last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "gene
formulaB.HS.Eucl.small <- last004scaled ~ 1 + vpliv1 + vpliv2 + vpliv3 + f(rowNumberAinv, model = "gene
```

5 FIT MODELS

Full dataset.

```
model.G <- inla(formulaB.G, family = "gaussian", data = phenoData, control.compute = list(dic=T, cpo=F,
model.H <- inla(formulaB.H, family = "gaussian", data = phenoData, control.compute = list(dic=T, cpo=F,
model.besag.S.UE <- inla(formulaB.S.UE, family = "gaussian", data = phenoData, control.compute = list(
model.besag.S.OB <- inla(formulaB.S.OB, family = "gaussian", data = phenoData, control.compute = list(
model.besag.HS.UE <- inla(formulaB.HS.UE, family = "gaussian", data = phenoData, control.compute = list(
model.besag.HS.OB <- inla(formulaB.HS.OB, family = "gaussian", data = phenoData, control.compute = list(
# Euclidean
model.Eucl.S <- inla(formulaB.S.Eucl, family = "gaussian", data = phenoData, control.compute = list(di
model.Eucl.HS <- inla(formulaB.HS.Eucl, family = "gaussian", data = phenoData, control.compute = list(
# SPDE
model.SPDE.S = inla(formula = formulaS, data = inla.stack.data(stack),
  family = "normal", control.predictor = list(A=inla.stack.A(stack), compute = T),
  control.family=list(list(hyper=hyperResVarRest)),
  control.compute = list(dic=T, cpo=F), verbose=T)
model.SPDE.HS = inla(formula = formulaHS, data = inla.stack.data(stackHS),
  family = "normal", control.predictor = list(A=inla.stack.A(stackHS), compute = T),
  control.family=list(list(hyper=hyperResVarRest)),
  control.compute = list(dic=T, cpo=F), verbose=T)
```

Data subset model fitting.

```

model.G.small <- inla(formulaB.G.small, family = "gaussian", data = phenoData.small, control.compute =
model.H.small <- inla(formulaB.H.small, family = "gaussian", data = phenoData.small, control.compute =
model.besag.S.UE.small <- inla(formulaB.S.UE.small, family = "gaussian", data = phenoData.small, contr
model.besag.S.OB.small <- inla(formulaB.S.OB.small, family = "gaussian", data = phenoData.small, contr
model.besag.HS.UE.small <- inla(formulaB.HS.UE.small, family = "gaussian", data = phenoData.small, con
model.besag.HS.OB.small <- inla(formulaB.HS.OB.small, family = "gaussian", data = phenoData.small, con
# Euclidean
model.Eucl.S.small <- inla(formulaB.S.Eucl.small, family = "gaussian", data = phenoData.small, control
model.Eucl.HS.small <- inla(formulaB.HS.Eucl.small, family = "gaussian", data = phenoData.small, contr
# SPDE
model.SPDE.S.small = inla(formula = formulaS.small, data = inla.stack.data(stack.small),
family = "normal", control.predictor =list(A=inla.stack.A.small(stack.small),compute = T),
control.family=list(list(hyper=hyperResVarRest)),
control.compute = list(dic=T,cpo=F, config=T), verbose=T)

model.SPDE.HS.small = inla(formula = formulaHS.small, data = inla.stack.data(stackHS.small),
family = "normal", control.predictor =list(A=inla.stack.A.small(stackHS.small),compute = T),
control.family=list(list(hyper=hyperResVarRest)),
control.compute = list(dic=T,cpo=F, config=T), verbose=T)

```

6 MODEL COMPARISON (VARIANCE COMPONENTS, DIC VALUES)

Full data.

```

G <- c(round(1/model.G$summary.hyperpar[2,1],2), "/", "/", round(1/model.G$summary.hyperpar[1,1],2), ro
GH <- c(round(1/model.H$summary.hyperpar[2,1],2), round(1/model.H$summary.hyperpar[3,1],2), "/", round(
GS.besag.OB <- c(round(1/model.besag.S.OB$summary.hyperpar[2,1],2), "/", round(1/model.besag.S.OB$summa
GS.besag.UE <- c(round(1/model.besag.S.UE$summary.hyperpar[2,1],2), "/", round(1/model.besag.S.UE$summa
GHS.besag.OB <- c(round(1/model.besag.HS.OB$summary.hyperpar[2,1],2), round(1/model.besag.HS.OB$summary
GHS.besag.UE <- c(round(1/model.besag.HS.UE$summary.hyperpar[2,1],2), round(1/model.besag.HS.UE$summary
GS.Eucl <- c(round(1/model.Eucl.S$summary.hyperpar[2,1],2), "/", round(1/model.Eucl.S$summary.hyperpar[
# GHS.Eucl <- c(round(1/model.Eucl.HS$summary.hyperpar[2,1],2), round(1/model.Eucl.HS$summary.hyperpar[
GS.SPDE <- c(round(1/model.SPDE.S$summary.hyperpar[2,1],2), "/", round(model.SPDE.S$summary.hyperpar[4,
GHS.SPDE <- c(round(1/model.SPDE.HS$summary.hyperpar[2,1],2), round(1/model.SPDE.HS$summary.hyperpar[3,

Modeli <- cbind(G, GH, GS.besag.OB, GS.besag.UE, GHS.besag.OB, GHS.besag.UE, GS.Eucl, GS.SPDE, GHS.SPDE)
rownames(Modeli) <- c("genetic effect", "herd effect", "spatial effect", "residual", "DIC")
kable(Modeli,caption = "Variance components for random effects") %>%
  kable_styling("striped",full_width = F, font_size=15)

```

Small data.

Table 1: Variance components for random effects

	G	GH	GS.besag.OB	GS.besag.UE	GHS.besag.OB	GHS.besag
genetic effect	0.77	0.2	0.69	0.72	0.19	0.2
herd effect	/	0.27	/	/	0.24	0.25
spatial effect	/	/	0.37	0.15	0.13	0.13
residual	0.31	0.49	0.35	0.32	0.49	0.49
DIC	67307	70939	68092	68000	71048	70902

```

# rm(list=ls())
# load("workspaces/import_data.RData")
# load("workspaces/spde_small.RData")
# load("workspaces/Eucl_models.RData")
# load("workspaces/UE_models.RData")

G.small <- c(round(1/model.G.small$summary.hyperpar[2,1],2), "/", "/", round(1/model.G.small$summary.hyperpar[3,1],2))
GH.small <- c(round(1/model.H.small$summary.hyperpar[2,1],2), round(1/model.H.small$summary.hyperpar[3,1],2))
GS.besag.OB.small <- c(round(1/model.besag.S.OB.small$summary.hyperpar[2,1],2), "/", round(1/model.besag.S.OB.small$summary.hyperpar[3,1],2))
GS.besag.UE.small <- c(round(1/model.besag.S.UE.small$summary.hyperpar[2,1],2), "/", round(1/model.besag.S.UE.small$summary.hyperpar[3,1],2))
GHS.besag.OB.small <- c(round(1/model.besag.HS.OB.small$summary.hyperpar[2,1],2), round(1/model.besag.HS.OB.small$summary.hyperpar[3,1],2))
GHS.besag.UE.small <- c(round(1/model.besag.HS.UE.small$summary.hyperpar[2,1],2), round(1/model.besag.HS.UE.small$summary.hyperpar[3,1],2))
GS.Eucl.small <- c(round(1/model.Eucl.S.small$summary.hyperpar[2,1],2), "/", round(1/model.Eucl.S.small$summary.hyperpar[3,1],2))
GHS.Eucl.small <- c(round(1/model.Eucl.HS.small$summary.hyperpar[2,1],2), round(1/model.Eucl.HS.small$summary.hyperpar[3,1],2))
GS.SPDE.small <- c(round(1/model.SPDE.S.small$summary.hyperpar[2,1],2), "/", paste(round(model.SPDE.S.small$summary.hyperpar[3,1],2), "x", round(model.SPDE.S.small$summary.hyperpar[4,1],2)))
GHS.SPDE.small <- c(round(1/model.SPDE.HS.small$summary.hyperpar[2,1],2), round(1/model.SPDE.HS.small$summary.hyperpar[3,1],2))

Modeli <- cbind(G.small, GH.small, GS.besag.OB.small, GS.besag.UE.small, GHS.besag.OB.small, GHS.besag.UE.small, GS.Eucl.small, GHS.Eucl.small, GS.SPDE.small, GHS.SPDE.small)
rownames(Modeli) <- c("genetic effect", "herd effect", "spatial effect", "residual", "DIC")
kable(Modeli, caption = "Variance components for random effects (small models)") %>%
  kable_styling("striped", full_width = F, font_size=15)

```

7 INLA POSTERIOR SAMPLING

Table 2: Variance components for random effects (small models)

	G.small	GH.small	GS.besag.OB.small	GS.besag.UE.small	GHS.besag
genetic effect	0.34	0.16	0.33	0.28	0.16
herd effect	/	0.26	/	/	0.23
spatial effect	/	/	0.06	0.14	0.06
residual	0.63	0.51	0.6	0.67	0.51
DIC	10205	9735	10112	10296	9717

```

# import Euclidean distance matrix
# import posterior samplings
load("workspaces/mc_samples.RData")
load("workspaces/Euclidean_dist_small.RData")

# xxG, xxH, xxS, xxHS, xxS.spde, xxHS.spde

pipeline_conn_1 <- function(nEff=56465, nSample=100, samples_model, data=phenoData.small, nameEff="Geff")

effects_sample <- matrix(NA, nrow=nEff, ncol=nSample)
nEff_full <- length(samples_model[[1]]$latent)
for(i in 1:nSample){
  effects_sample_full <- samples_model[[i]]$latent[1:nEff_full,]
  effects_sample[,i] <- effects_sample_full[grepl("^rowNumber", names(effects_sample_full))]
}
df_Row_eff <- data.frame("rowNumberAinv" = 1:nEff, nameEff = effects_sample)
rowNumber <- data.frame("rowNumberAinv" = data$rowNumberAinv)
df_Row_eff <- as.data.frame(merge(rowNumber, df_Row_eff, all.x=T))
row.names(df_Row_eff) <- df_Row_eff$rowNumberAinv
df_Row_eff$rowNumberAinv <- NULL
df_Row_eff <- data.matrix(df_Row_eff)
df_Row_eff

sampled.G.gen.small <- pipeline_conn_1(samples_model=sampled.G.small)
sampled.HS.SPDE.gen.small <- pipeline_conn_1(samples_model=sampled.HS.SPDE.small)

RowVar <- function(x, ...) {
  rowSums((x - rowMeans(x, ...))^2, ...)/(dim(x)[2] - 1)
}

variance_animal.G <- RowVar(sampled.G.gen.small)
variance_animal.HS.SPDE <- RowVar(sampled.HS.SPDE.gen.small)

pipeline_PVD <- function(var_animal, samples){
  sum_var = outer(var_animal, -var_animal, '-')
  varcovM <- cov(t(samples), t(samples))
  pvd <- sum_var - varcovM
}

```

```

    pevd[lower.tri(pevd,diag=TRUE)] <- NA
    pevd_num <- as.vector(pevd)
    pevd_num <- pevd_num[!is.na(pevd_num)]
    pevd_num
  }
  pevd.G.small <- pipeline_PEVD(var_animal=variance_animal.G, samples=sampled.G.gen.small)
  pevd.HS.SPDE.small <- pipeline_PEVD(var_animal=variance_animal.HS.SPDE, samples=sampled.HS.SPDE.gen.sma

pipeline_Eucl <- function(dist){
  dist[lower.tri(dist,diag=TRUE)] <- NA
  Eucl_dist_num <- as.vector(dist)
  Eucl_dist_num <- Eucl_dist_num[!is.na(Eucl_dist_num)]
  Eucl_dist_num
}

pevd.Eucl.small <- pipeline_Eucl(dist=paired.distances)

cor(pevd.Eucl.small, pevd.G.small)

## [1] -0.00334999

cor(pevd.Eucl.small, pevd.HS.SPDE.small)

## [1] 0.004101414

cor(pevd.G.small, pevd.HS.SPDE.small)

## [1] 0.3556385

```