

Using RandomFields for genomic prediction

U. Ober, H. Simianer and M. Schlather

Department of Animal Sciences
Animal Breeding and Genetics Group
Georg-August-University Göttingen

March 28th, 2011
Workshop R-package “synbreed”



RandomFields

The R-package `RandomFields` of Martin Schlather – a flexible tool for:

- estimation of variance components in arbitrary linear mixed models
- performing BLUP based on geostatistical models

Genomic Selection

- Standard model:

$$\text{phenotype} = \text{fixed effect} + (\text{random effect}) + \text{GV} \\ + \text{environmental effect}$$

- Assumption: GV depending on pedigree- and SNP-information
- Aim: prediction of GV

Genomic Selection with Kriging

- Idea: use geostatistical kriging method for genomic selection
- Assumption: GV is realization of a Gaussian random field with parameterized covariance function
- Use geostatistical covariance functions to display covariance structures
- Use SNPs as coordinates at which random field is observed

Kriging

Kriging consists of two **steps**:

1. Estimation of the unknown parameters and variables (e.g. via maximum-likelihood-methods)
 - use `fitvario` of R-package `RandomFields` for parameter estimation
 - use `CovarianceFct` of R-package `RandomFields` to calculate covariance structures for given parameters
2. Prediction via BLUP using the estimated covariance structures

Example: Statistical Model

$$\mathbf{y} = \mathbf{W}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + g(\mathbf{X}) + \mathbf{e},$$

with

- \mathbf{y} phenotypes,
- $\boldsymbol{\beta}$ vector of fixed effects,
- \mathbf{W} and \mathbf{Z} incidence matrices,
- $\mathbf{u} \sim \mathcal{N}(0, \sigma_u^2 \mathbf{A})$ with \mathbf{A} being the additive relationship matrix,
- \mathbf{x}_i a vector of SNPs observed on individual i (genotype),
 $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$
- $g(\cdot)$ a Gaussian random field with $\mathbb{E}(g(\mathbf{x}_i)) = 0$ and
 $\text{Cov}(g(\mathbf{x}_i), g(\mathbf{x}_j)) = K(\mathbf{x}_i, \mathbf{x}_j)$,
with K being a covariance function depending on different parameters
- $\mathbf{e} \sim \mathcal{N}(0, \sigma_e \mathbf{I})$ a vector of residual effects

aim: prediction of GV $\mathbf{z}_0^T \mathbf{u} + g(\mathbf{x}_0)$

Possible covariance functions

e.g.

- Matérn covariance function
- Exponential covariance function
- Gaussian covariance function
- ...

Universal Kriging

Recall: $\mathbf{y} = \mathbf{W}\beta + \mathbf{Z}\mathbf{u} + g(\mathbf{X}) + \mathbf{e}$

Steps:

1. Estimation of all parameters of K via ML, considering the density function of \mathbf{y} using `fitvario`
2. BLUP-approach for $g(\mathbf{x}_0)$:
minimize $\mathbb{E}(\hat{g}(\mathbf{x}_0) - g(\mathbf{x}_0))^2$
with $\hat{g}(\mathbf{x}_0) = \mathbf{a}_g^T \mathbf{y}$ and $\mathbf{a}_g^T \mathbf{W} = 0$
3. Solution:

$$\begin{bmatrix} \mathbf{W} & \sigma_u^2 \mathbf{ZAZ}^T + \mathbf{K} + \sigma_e^2 \mathbf{I} \\ 0 & \mathbf{W}^T \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ \mathbf{a}_g \end{bmatrix} = \begin{bmatrix} \mathbf{K}_0 \\ 0 \end{bmatrix}$$

→ use `CovarianceFct` to calculate \mathbf{K}

4. analogously for $\mathbf{z}_0^T \mathbf{u}$, finally: $\widehat{GV} = \hat{g}(\mathbf{x}_0) + \widehat{\mathbf{z}_0^T \mathbf{u}}$

Code (I)

```
# load package
library(RandomFields)

# specify model
model <- list("+",
              list("X", X=Z, cov=list("$", var=NA,
                                   list("const", M=A))),
              list("$", scale=NA, var=NA,
                    list("matern", nu=NA)),
              list("$", var=NA, list("nugget"))
              )

# use SNP-data to estimate parameters
fit <- fitvario(x=SNPs, data=y, model=model,
               trend=NA, mle.methods="ml")
```


Code (III)

```
# Prediction via BLUP
KK <- K[idx,idx]
K_0 <- K[idx,-idx]
W <- matrix(rep(1,n.est))
D_up <- cbind(W, sigma2.u.est*A[idx,idx]+KK+
              sigma2.e.est*diag(n.est))
D_down <- cbind(0, t(W))
D <- rbind(D_up, D_down)
Dinv <- solve(D)

a.g <- (Dinv%%rbind(K_0,0))[-1,]
g.est <- t(a.g)%%y

a.u <- (Dinv%%rbind(sigma2.u.est*A[idx,-idx],0))[-1,]
u.est <- t(a.u)%%y

GV <- g.est + u.est
```

Possible contributions to Synbreed-R-package

Implement genomic selection using `RandomFields`:

- automatic call for `RandomFields`
- flexible model building and model selection (GBLUP; mixed linear models with geostatistical components)
- ML-estimation of variance-components for mixed linear models using `fitvario`
(→ `fitvario` as an alternative to “ASReml”)