Spatial modeling using the sommer package

Giovanny Covarrubias-Pazaran

2022-04-16

The sommer package was developed to provide R users with a powerful and reliable multivariate mixed model solver for different genetic (in diploid and polyploid organisms) and non-genetic analyses. This package allows the user to estimate variance components in a mixed model with the advantages of specifying the variance-covariance structure of the random effects, specifying heterogeneous variances, and obtaining other parameters such as BLUPs, BLUEs, residuals, fitted values, variances for fixed and random effects, etc. The core algorithms of the package are coded in C++ using the Armadillo library to optimize dense matrix operations common in the direct-inversion algorithms.

This vignette is focused on showing the capabilities of sommer to fit spatial models using the two dimensional splines models.

SECTION 1: Introduction

1) Background in tensor products

SECTION 2: Spatial models

1) Two dimensional splines (multiple spatial components)
2) Two dimensional splines (single spatial component)
3) Spatial models in multiple trials at once

SECTION 1: Introduction

Backgrounds in tensor products

TBD

SECTION 2: Spatial models

1) Two dimensional splines (multiple spatial components)

In this example we show how to obtain the same results than using the SpATS package. This is achieved by using the spl2Db function which is a wrapper of the tpsmmb function.

```r
library(sommer)
data(DT_yatesoats)
DT <- DT_yatesoats
DT$row <- as.numeric(as.character(DT$row))
DT$col <- as.numeric(as.character(DT$col))
DT$R <- as.factor(DT$row)
DT$C <- as.factor(DT$col)

# SPATS MODEL
# m1.SpATS <- SpATS(response = "Y",
```
```r
# spatial = ~ PSANOVA(col, row, nseg = c(14,21), degree = 3, pord = 2),
# genotype = "V", fixed = ~ 1,
# random = ~ R + C, data = DT,
# control = list(tolerance = 1e-04))
#
# summary(m1.SpATS, which = "variances")
#
# Spatial analysis of trials with splines
#
# Response: Y
# Genotypes (as fixed): V
# Spatial: ~PSANOVA(col, row, nseg = c(14, 21), degree = 3, pord = 2)
# Fixed: ~1
# Random: ~R + C
#
# Number of observations: 72
# Number of missing data: 0
# Effective dimension: 17.09
# Deviance: 483.405
#
# Variance components:
# Variance     SD  log10(lambda)
# R           1.277e+02 1.130e+01 0.49450
# C           2.673e-05 5.170e-03 7.17366
# f(col)      4.018e-15 6.339e-08 16.99668
# f(row)      2.291e-10 1.514e-05 12.24059
# f(col):row  1.025e-04 1.012e-02  6.59013
# col:f(row)  8.789e+01 9.375e+00  0.65674
# f(col):f(row) 8.036e-04 2.835e-02  5.69565
# Residual   3.987e+02 1.997e+01

# SOMMER MODEL
m1.sommer <- mmer(Y ~ 1 + V + spl2Db(col, row, nsegments = c(14,21), degree = c(3,3), penaltyord = c(2,2), what = "base"),
                 random = ~ R + C + spl2Db(col, row, nsegments = c(14,21), degree = c(3,3), penaltyord = c(2,2), what = "base"),
                 data=DT, tolpar = 1e-6, verbose = FALSE)

## Warning: fixed-effect model matrix is rank deficient so dropping 8 columns / coefficients
summary(m1.sommer)$varcomp

# get the fitted values for the spatial kernel and plot
# ff <- fitted.mmer(m1.sommer)
# DT$fit <- as.matrix(Reduce("+", ff$zu[-c(1:2)]))
```
2) Two dimensional splines (single spatial component)

To reduce the computational burden of fitting multiple spatial kernels \texttt{sommer} provides a single spatial kernel method through the \texttt{spl2Da} function. This as will be shown, can produce similar results to the more flexible model. Use the one that fits better your needs.

\begin{verbatim}
# SOMMER MODEL
m2.sommer <- mmer(Y ~ 1 + V,
  random = -R+C+spl2Da(col, row, nsegments = c(14,21), degree = c(3,3), penaltyord = c(2,2),
  data=DT, tolpar = 1e-6, verbose = FALSE)

summary(m1.sommer)$varcomp

## VarComp VarCompSE Zratio Constraint
## R.Y-Y 125.928235 89.77330 1.4027360 Positive
## C.Y-Y -7.789528 24.29529 -0.3206189 Positive
## A:fC.Y-Y 0.000000 19.09624 0.0000000 Positive
## A:fR.Y-Y 0.000000 15.87659 0.0000000 Positive
## A:fC.R.Y-Y 0.000000 21.42763 0.0000000 Positive
## A:C.fR.Y-Y 82.177296 92.28630 0.8904604 Positive
## A:fC.fR.Y-Y 0.000000 25.46390 0.0000000 Positive
## units.Y-Y 405.900386 90.48195 4.4859820 Positive

# get the fitted values for the spatial kernel and plot
# ff <- fitted.mmer(m2.sommer)
# DT$fit <- as.matrix(Reduce("+",ff$Zu[-c(1:2)]))
# lattice::levelplot(fit~row*col,data=DT)
\end{verbatim}

3) Spatial models in multiple trials at once

Sometimes we want to fit heterogeneous variance components when e.g., have multiple trials or different locations. The spatial models can also be fitted that way using the \texttt{at.var} and \texttt{at.levels} arguments. The first argument expects a variable that will define the levels at which the variance components will be fitted. The second argument is a way for the user to specify the levels at which the spatial kernels should be fitted if the user doesn’t want to fit it for all levels (e.g., trials or fields).

\begin{verbatim}
DT2 <- rbind(DT,DT)
DT2$Y <- DT2$Y + rnorm(length(DT2$Y))
DT2$trial <- c(rep("A",nrow(DT)),rep("B",nrow(DT)))
head(DT2)

## row col Y N V B MP R C trial
## 1 1 1 91.79843 0.2 Victory B2 Victory 1 1 A
## 2 2 1 61.85086 0 Victory B2 Victory 2 1 A
## 3 3 1 120.55643 0.4 Marvellous B2 Marvellous 3 1 A
## 4 4 1 143.55323 0.6 Marvellous B2 Marvellous 4 1 A
## 5 5 1 149.01331 0.6 GoldenRain B2 GoldenRain 5 1 A
## 6 6 1 106.56385 0.2 GoldenRain B2 GoldenRain 6 1 A

# SOMMER MODEL
m3.sommer <- mmer(Y ~ 1 + V,
  random = ~vs(ds(trial), R)+vs(ds(trial), C)+spl2Da(col, row, nsegments = c(14,21), degree = c(3,3), penaltyord = c(2,2), at.var =
\end{verbatim}
rcov = ~vs(ds(trial),units),
data=DT2, tolpar = 1e-6, verbose = FALSE)

summary(m3.sommer)$varcomp

## VarComp VarCompSE Zratio Constraint
## A:R.Y-Y 107.48007 82.12826 1.3086855 Positive
## B:R.Y-Y 98.26652 80.47655 1.2210578 Positive
## A:C.Y-Y 144.95281 138.74448 1.0447465 Positive
## B:C.Y-Y 138.91292 134.98994 1.0290613 Positive
## A:all.Y-Y 403.81707 879.19318 0.4593041 Positive
## B:all.Y-Y 418.54730 901.30369 0.4643799 Positive
## A:units.Y-Y 385.64550 202.89149 1.9007475 Positive
## B:units.Y-Y 396.86541 208.15464 1.9065893 Positive

# get the fitted values for the spatial kernel and plot
# ff <- fitted.mmer(m3.sommer)
# DT2$fit <- as.matrix(Reduce("+", ff$Zu[-c(1:4)]))
# lattice::levelplot(fit~row*col|trial,data=DT2)

Final remarks

Keep in mind that sommer uses a direct inversion (DI) algorithm which can be very slow for large datasets. The package is focused on problems of the type p > n (more random effect levels than observations) and models with dense covariance structures. For example, for experiments with dense covariance structures with low-replication (i.e. 2000 records from 1000 individuals replicated twice with a covariance structure of 1000x1000) sommer will be faster than MME-based software. Also for genomic problems with large number of random effect levels, i.e. 300 individuals (n) with 100,000 genetic markers (p). For highly replicated trials with small covariance structures or n > p (i.e. 2000 records from 200 individuals replicated 10 times with covariance structure of 200x200) asreml or other MME-based algorithms will be much faster and we recommend you to opt for those. When datasets are big, the installation of the OpenBLAS library can make sommer quite fast and sometimes faster than asreml given the capability of sommer to take advantage of the multi-processor architecture of some systems.

Literature


Covarrubias-Pazaran G. 2018. Software update: Moving the R package sommer to multivariate mixed models for genome-assisted prediction. doi: https://doi.org/10.1101/354639


