

# Employing `asremlPlus`, in conjunction with `asreml`, to calculate and use information criteria

Chris Brien

05 February, 2024

This vignette illustrates the facilities in `asremlPlus` (Brien, 2024), in conjunction with `asreml` (Butler et al., 2023), for calculating and using information. Here, `asremlPlus` and `asreml` are packages for the R Statistical Computing environment (R Core Team, 2024).

It is divided into the following main sections:

1. Set up the maximal model for this experiment
2. Obtaining information criteria for separate models
3. Obtaining information criteria for a prescribed sequence of model changes
4. Using information criteria to decide model changes

## 1. Set up the maximal model for this experiment

```
library(knitr)
opts_chunk$set("tidy" = FALSE, comment = NA)
suppressMessages(library(asreml, quietly=TRUE))

## Offline License checked out Mon Feb 5 18:43:15 2024
```

```
packageVersion("asreml")
```

```
## [1] '4.2.0.276'
```

```
suppressMessages(library(asremlPlus))
packageVersion("asremlPlus")
```

```
## [1] '4.4.27'
```

```
options(width = 100)
```

### Get data available in `asremlPlus`

The data are from a 1976 spring wheat experiment and are taken from Gilmour et al. (1995). An analysis is presented in the `asreml` manual by Butler et al. (2023, Section 7.6), although they suggest that it is a barley experiment.

```
data(Wheat.dat)
```

## Fit the maximal model

In the following a model is fitted that has the terms that would be included for a balanced lattice. In addition, a term WithinColPairs has been included to allow for extraneous variation arising between pairs of adjacent lanes. Also, separable ar1 residual autocorrelation has been included. This model represents the maximal anticipated model,

```
max.asr <- asreml(yield ~ WithinColPairs + Variety,  
                random = ~ Rep/(Row + Column) + units,  
                residual = ~ ar1(Row):ar1(Column),  
                data=Wheat.dat)
```

Warning in asreml(yield ~ WithinColPairs + Variety, random = ~Rep/(Row + : Some components changed by more than 1% on the last iteration

The warning from asreml is probably due to a bound term.

## Initialize a testing sequence by loading the current fit into an asrtests object

```
max.asrt <- as.asrtests(max.asr, NULL, NULL)
```

## Check for and remove any boundary terms

```
max.asrt <- rmboundary(max.asrt)  
summary(max.asrt$asreml.obj)$varcomp
```

	component	std.error	z.ratio	bound	%ch
Rep:Row	4.293282e+03	3.199458e+03	1.3418779	P	0.0
Rep:Column	1.575689e+02	1.480357e+03	0.1064398	P	0.7
units	5.742689e+03	1.652457e+03	3.4752438	P	0.0
Row:Column!R	4.706787e+04	2.515832e+04	1.8708669	P	0.0
Row:Column!Row!cor	7.920301e-01	1.014691e-01	7.8056280	U	0.0
Row:Column!Column!cor	8.799559e-01	7.370402e-02	11.9390486	U	0.0

```
print(max.asrt, which = "testsummary")
```

```
#### Sequence of model investigations
```

(If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)

```
terms DF denDF p AIC BIC action  
1 Rep 1 NA NA NA NA Boundary
```

Rep has been removed because it has been constrained to zero. Following the recommendation of Littel et al. (2006, p. 150), the bound on all variance components is set to unconstrained (U) using `setvariances.asreml` so as to avoid bias in the estimate of the residual variance. Alternatively, one could move Rep to the fixed model.

## Unbind Rep, Row and Column components and reload into an `asrtests` object

```
max.asr <- setvarianceterms(max.asr$call,
                           terms = c("Rep", "Rep:Row", "Rep:Column"),
                           bounds = "U")
```

Warning in `asreml(fixed = yield ~ WithinColPairs + Variety, random = ~Rep/(Row + : Some components changed by more than 1% on the last iteration`

```
max.asrt <- as.asrtests(max.asr, NULL, NULL)
max.asrt <- rmboundary(max.asrt)
summary(max.asrt$asreml.obj)$varcomp
```

	component	std.error	z.ratio	bound	%ch
Rep	-2458.3485841	1.197491e+03	-2.0529167	U	0.0
Rep:Row	5008.7151486	3.401335e+03	1.4725732	U	0.0
Rep:Column	916.4641198	1.699576e+03	0.5392309	U	0.2
units	5959.0220817	1.609649e+03	3.7020634	P	0.0
Row:Column!R	46637.6303429	2.724392e+04	1.7118545	P	0.0
Row:Column!Row!cor	0.8150590	1.000281e-01	8.1483012	U	0.0
Row:Column!Column!cor	0.8856824	7.492514e-02	11.8208968	U	0.0

```
print(max.asrt, which = "testsummary")
```

```
#### Sequence of model investigations
```

(If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)

```
[1] terms DF denDF p AIC BIC action
<0 rows> (or 0-length row.names)
```

Now the Rep component estimate is negative.

The `test.summary` output shows that no changes have been made to the model loaded using `as.asrtests`. The pseudo-anova table shows that Varieties are highly significant ( $p < 0.001$ )

## 2. Obtaining information criteria for separate models

The method `infoCriteria` has two methods for calculating information criteria. One, `infoCriteria.asreml`, is a method for `asreml` objects and the other, `infoCriteria.list`, is for 'listobjects, the components of `thelistbeingasreml`' objects.

## Single models

Firstly, `infoCriteria` is called with the default `IClikelihood`, which is `REML`. Then it is called with `IClikelihood` set to `full` (Verbyla, 2019).

```
infoCriteria(max.asr)
```

```
  fixedDF varDF NBound      AIC      BIC  loglik
1         0     7       0 1396.34 1416.082 -691.17
```

```
infoCriteria(max.asr, IClikelihood = "full")
```

```
Warning in asreml(fixed = yield ~ WithinColPairs + Variety, random = ~Rep/(Row + : Log-likelihood
not converged
```

```
  fixedDF varDF NBound      AIC      BIC  loglik
1        26     7       0 1647.191 1746.542 -790.5957
```

## A list of models

Now, a second model, from which the `withinColPairs` term has been omitted, is fitted; to be consistent, the variance components are unconstrained using `setvariances.asreml`. Then the `asreml` objects for this model and the maximal model are combined into a `list` and a `data.frame` produced that includes their information criteria.

```
m1.asr <- asreml(yield ~ Variety,
               random = ~ Rep/(Row + Column) + units,
               residual = ~ ar1(Row):ar1(Column),
               data=Wheat.dat)
```

```
Warning in asreml(yield ~ Variety, random = ~Rep/(Row + Column) + units, : Some components changed
by more than 1% on the last iteration
```

```
m1.asr <- setvarianceterms(m1.asr$call,
                          terms = c("Rep", "Rep:Row", "Rep:Column"),
                          bounds = "U")
```

```
Warning in asreml(fixed = yield ~ Variety, random = ~Rep/(Row + Column) + : Some components changed
by more than 1% on the last iteration
```

```
mods <- list(max = max.asr, m1 = m1.asr)
ic <- infoCriteria(mods, IClikelihood = "full")
print(ic)
```

```
  fixedDF varDF NBound      AIC      BIC  loglik
max        26     7       0 1647.191 1746.542 -790.5957
m1         25     7       0 1645.318 1741.658 -790.6588
```

### 3. Obtaining information criteria for a prescribed sequence of model changes

The use of `changeTerms.asrtests` is demonstrated for a sequence of models, starting with the maximal model.

#### Drop the term for within Column pairs (a post hoc factor)

```
current.asrt <- as.asrtests(max.asrt$asreml.obj, NULL, NULL,  
                           label = "Maximal model", IClkelihood = "full")
```

Warning in `asreml(fixed = yield ~ WithinColPairs + Variety, random = ~Rep/(Row + : Log-likelihood not converged`

```
current.asrt <- changeTerms(current.asrt, dropFixed = "WithinColPairs",  
                            label = "Drop withinColPairs", IClkelihood = "full")
```

Warning in `asreml(fixed = yield ~ Variety, random = ~Rep + units + Rep:Row + : Some components changed by more than 1% on the last iteration`

Warning in `asreml(fixed = yield ~ Variety, random = ~Rep + units + Rep:Row + : Some components changed by more than 1% on the last iteration`

```
print(current.asrt, which = "testsummary", omit.columns = "p")
```

```
#### Sequence of model investigations
```

(If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)

	terms	DF	denDF	AIC	BIC	action
1	Maximal model	26	7	1647.191	1746.542	Starting model
2	Drop withinColPairs	25	7	1645.325	1741.666	Changed fixed

So the same values of the information criteria have been obtained as when `infoCriteria.list` was used on a list containing the `asreml` objects for the two models. The difference is that here there is ultimately only one fitted model, the model stored in the `asreml` object in the `asrtests` object named `current.asrt`: this is the model with `withinColPairs` omitted.

Note this use of the `omit.columns` argument from `print.test.summary` to omit the irrelevant column `p` from the `test.summary`.

#### Drop nugget term

```
current.asrt <- changeTerms(current.asrt, dropRandom = "units",  
                            label = "Drop units", IClkelihood = "full")
```

```
Warning in asreml(fixed = yield ~ Variety, random = ~Rep + Rep:Row + Rep:Column, : Some components
changed by more than 1% on the last iteration
```

```
Warning in asreml(fixed = yield ~ Variety, random = ~Rep + Rep:Row + Rep:Column, : Some components
changed by more than 1% on the last iteration
```

## Check Row autocorrelation

```
current.asrt <- changeTerms(current.asrt, newResidual = "Row:ar1(Column)",
                             label="Row autocorrelation", ICLikelihood = "full")
print(current.asrt, which = "testsummary", omit.columns = "p")
```

```
#### Sequence of model investigations
```

(If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)

	terms	DF	denDF	AIC	BIC	action
1	Maximal model	26	7	1647.191	1746.542	Starting model
2	Drop withinColPairs	25	7	1645.325	1741.666	Changed fixed
3	Drop units	25	6	1650.126	1743.456	Changed random
4	Row autocorrelation	25	5	1660.882	1751.201	Changed residual

## 4. Using information criteria to decide model changes

This sections illustrates the use of `changeModelOnIC.asrtests` to decide between consecutive models in a sequence of models. The default information criterion to use for this is the AIC. However, `which.IC` can be used to specify the use of the BIC or both. Here we use the AIC and the full likelihood.

### Check the term for within Column pairs (a post hoc factor)

As before, we start with the maximal model, in which the variance components have been unconstrained and look to decide whether of not to drop the `withinColPairs` term.

```
current.asrt <- as.asrtests(max.asrt$asreml.obj, NULL, NULL,
                           label = "Maximal model", ICLikelihood = "full")
```

```
Warning in asreml(fixed = yield ~ WithinColPairs + Variety, random = ~Rep/(Row + : Log-likelihood
not converged
```

```
current.asrt <- iterate(current.asrt)
current.asrt <- changeModelOnIC(current.asrt, dropFixed = "WithinColPairs",
                                label = "withinColPairs",
                                ICLikelihood = "full", which.IC = "AIC",
                                allow.unconverged = FALSE)
```

```
Warning in asreml(fixed = yield ~ Variety, random = ~Rep + units + Rep:Row + : Some components
changed by more than 1% on the last iteration
```

Warning in asreml(fixed = yield ~ Variety, random = ~Rep + units + Rep:Row + : Some components changed by more than 1% on the last iteration

```
print(current.asrt, which = "testsummary", omit.columns = "p")
```

#### Sequence of model investigations

(If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)

	terms	DF	denDF	AIC	BIC	action
1	Maximal model	26	7	1647.191446	1746.542411	Starting model
2	withinColPairs	-1	0	-1.866103	-4.876738	Swapped

Given the warning about a lack of convergence, we use `iterate.asrtests` to perform additional iterations of the fitting process. It seems that it was successful.

It can be seen from the `test.summary` that the term has been swapped out and this has the effect of reducing the number of fixed parameters by one and makes no change to the variance parameters.

## Check the nugget term

```
current.asrt <- changeModelOnIC(current.asrt, dropRandom = "units",  
                                label = "units", IClkelihood = "full",  
                                allow.unconverged = FALSE)
```

Warning in asreml(fixed = yield ~ Variety, random = ~Rep + Rep:Row + Rep:Column, : Some components changed by more than 1% on the last iteration

Warning in asreml(fixed = yield ~ Variety, random = ~Rep + Rep:Row + Rep:Column, : Some components changed by more than 1% on the last iteration

## Check Row autocorrelation

```
current.asrt <- changeModelOnIC(current.asrt, newResidual = "Row:ar1(Column)",  
                                label="Row autocorrelation", IClkelihood = "full",  
                                allow.unconverged = FALSE)
```

Warning in asreml(fixed = yield ~ Variety, random = ~Rep + units + Rep:Row + : Log-likelihood not converged

Warning in asreml(fixed = yield ~ Variety, random = ~Rep + units + Rep:Row + : Some components changed by more than 1% on the last iteration

Warning in asreml(fixed = yield ~ Variety, random = ~Rep + units + Rep:Row + : Log-likelihood not converged

Warning in asreml(fixed = yield ~ Variety, random = ~Rep + units + Rep:Row + : Some components changed by more than 1% on the last iteration

Warning in newfit.asreml(asreml.obj, fixed. = fix.form, random. = ran.form, :

## Check Column autocorrelation (depends on whether Row autocorrelation retained)

```
{
  last.action <- current.asrt$test.summary$action[current.asrt$test.summary$terms ==
                                                    "Row autocorrelation"]
  if (grepl("Unswapped", last.action, fixed = TRUE) |
      grepl("Unchanged", last.action, fixed = TRUE))
    current.asrt <- changeModelOnIC(current.asrt, newResidual = "ar1(Row):Column",
                                   label="Col autocorrelation", ICLikelihood = "full",
                                   allow.unconverged = FALSE)
  else
    current.asrt <- testresidual(current.asrt, newResidual = "Row:Column",
                                 label="Col autocorrelation", ICLikelihood = "full",
                                 allow.unconverged = FALSE)
}
```

Warning in infoCriteria.asreml(asreml.obj, ICLikelihood = ic.lik, bound.exclusions = bound.exclusions):  
Row:Column!Row!cor

Warning in rmboundary.asrtests(as.asrtests(asreml.obj, wald.tab, test.summary, : In analysing yield, es  
Row:Column!Row!cor

Warning in infoCriteria.asreml(new.asrtests.obj\$asreml.obj, ICLikelihood = ic.lik, : The following bound  
Row:Column!Row!cor

## Output the results

```
print(current.asrt, which = "test", omit.columns = "p")
```

#### Sequence of model investigations

(If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)

	terms	DF	denDF	AIC	BIC	action
1	Maximal model	26	7	1.647191e+03	1.746542e+03	Starting model
2	withinColPairs	-1	0	-1.866103e+00	-4.876738e+00	Swapped
3	units	0	-1	4.801053e+00	1.790418e+00	Unswapped
4	Row autocorrelation	0	0	-7.342295e-03	-7.342295e-03	Unchanged - new unconverged
5	Col autocorrelation	0	-2	1.947985e+01	1.345858e+01	Unswapped

```
summary(current.asrt$asreml.obj)$varcomp
```

	component	std.error	z.ratio	bound	%ch
Rep	-2391.8946799	1.194671e+03	-2.002136	U	0.4
Rep:Row	5035.4828349	3.406065e+03	1.478387	U	0.3



Rep:Column	761.9005140	1.612048e+03	0.472629	U 1.2
units	5933.1408473	1.610819e+03	3.683306	P 0.1
Row:Column!R	45970.2439168	2.635029e+04	1.744582	P 0.0
Row:Column!Row!cor	0.8101593	9.995689e-02	8.105087	U 0.1
Row:Column!Column!cor	0.8846965	7.503099e-02	11.791081	U 0.0

The `test.summary` shows us that the model without the autocorrelation failed to converge and so no change was made to the model. It, and the messages from checking the Column autocorrelation, also show us that the omission of the Column autocorrelation resulted in the Row autocorrelation becoming bound. That is, dropping the Column autocorrelation resulted in the dropping of two variance parameters

The function `printFormulae.asreml` is used to display the fitted model.

```
printFormulae(current.asrt$asreml.obj)
```

```
#### Formulae from asreml object
```

```
fixed: yield ~ Variety
random: ~ Rep + units + Rep:Row + Rep:Column
residual: ~ ar1(Row):ar1(Column)
```

## References

- Brien, C. J. (2024) `asremlPlus`: *Augments ASReml-R in fitting mixed models and packages generally in exploring prediction differences*. Version 4.4.27. <https://cran.r-project.org/package=asremlPlus/> or <http://chris.brien.name/rpackages/>.
- Butler, D. G., Cullis, B. R., Gilmour, A. R., Gogel, B. J. and Thompson, R. (2023). *ASReml-R Reference Manual Version 4.2*. VSN International Ltd, <https://https://asreml.kb.vsnr.com.au/>.
- Gilmour, A. R., Thompson, R., & Cullis, B. R. (1995). Average Information REML: An Efficient Algorithm for Variance Parameter Estimation in Linear Mixed Models. *Biometrics*, **51**, 1440–1450.
- Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., & Schabenberger, O. (2006). *SAS for Mixed Models* (2nd ed.). Cary, N.C.: SAS Press.
- R Core Team (2024) *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.r-project.org/>.
- Verbyla, A. P. (2019). A note on model selection using information criteria for general linear models estimated using REML. *Australian & New Zealand Journal of Statistics*, **61**, 39-50. <https://doi.org/10.1111/anzs.12254/>.