

```

# Function reworked by C. Scherber based on an original version of anova.lmer

# Last edits made on 24th June 2010.

# The function requires two arguments:
# (1) a null model
# (2) a full model

# Based on the null model, terms are added sequentially (first to last),
# until the full model (with k terms) is reached

# The resulting k+1 models (including the intercept-only model) are then subjected to a
# QAICc-based model selection.

# The best model of this set (based on QAICc) is then subjected to subsequent likelihood ratio tests
# These tests again start with the null model until j+1 terms have been added
# (where j is the number of terms in the QAICc-based best model.

anova.lmer=function(nullmodel=quote(nullmodel),maxmodel=quote(maxmodel),...){
require(lme4)

# sub-function to calculate QAICc
# According to Bolker et al. 2009:

QAICc <- function(mod, scale, QAICc = TRUE) {
LL <- logLik(mod)
ll <- as.numeric(LL)
df <- attr(LL, "df")
n <- length(mod@y)
if (QAICc)
qaic = as.numeric(-2 * ll/scale + 2 * df + 2 * df * (df +
1)/(n - df - 1))
else qaic = as.numeric(-2 * ll/scale + 2 * df)
qaic
}

# Extract overdispersion parameter from the model

phi = lme4:::sigma(maxmodel)^2

#the ^2 is from a recent suggestion by Ben Bolker
# on R-Sig-ME from 10.02.2010, 22:33.

# Extract the model terms
modelterms=attr(attr(maxmodel@frame,"terms"),"term.labels")

modelA<-nullmodel

#####
## First section: Do the QAICc model selection:
model=c()

```

```

Currentmodel=c()
Currentmodel.parsed=c()
Currentmodel.eval=c()
modelcall=c()
modelQAIC=c()
df=c()

modelcall[length(modelterms)+1]=deparse(summary(nullmodel)@call,width.cutoff=500L)
modelQAIC[length(modelterms)+1]<-QAICc(nullmodel,scale = phi)

for(i in 1:length(modelterms)){
  lett=LETTERS[i]
  model[i]<-paste("update(model",lett,"~.",modelterms[i],")",sep="")
  upmodel<-paste("model",LETTERS[i+1],"<- ",sep="")

  Currentmodel<-paste(upmodel,model[i],sep="")
  Currentmodel.parsed=parse(text=Currentmodel)
  Currentmodel.eval=eval(Currentmodel.parsed)

  modelcall[i]=deparse(Currentmodel.eval@call,width.cutoff=500L)
  modelQAIC[i]<-QAICc(Currentmodel.eval,scale = phi)
}

QAICc.table=data.frame(Model=modelcall,QAICc=modelQAIC)

#####

## Second section: Select the QAICc-based minimal adequate model
## and return a sequential ANOVA table for it:

QAICc.table.ordered<-QAICc.table[order(QAICc.table[,2]),]
bestmodel<-QAICc.table.ordered[1,1]

eval.bestmodel<-eval(parse(text=as.character(bestmodel)))
bestmodelterms=attr(attr(eval.bestmodel@frame,"terms"),"term.labels")

anovamodelA<-nullmodel

anovamodel=c()

for(i in 1:length(bestmodelterms)){
  lett=LETTERS[i]
  anovamodel[i+1]<-paste("update(anovamodel",lett,"~.",bestmodelterms[i],")",sep="")
  upanovamodel<-paste("anovamodel",LETTERS[i+1],"<- ",sep="")
  current.anovamodel<-paste(upanovamodel,anovamodel[i+1],sep="")
  anovamodel.parsed=parse(text=current.anovamodel)
  eval(anovamodel.parsed)
}

anovacall<-
paste("anova(nullmodel,",paste("anovamodel",LETTERS[1:length(bestmodelterms)+1],sep="",collapse=","),")",sep="")

anovaexpression<-parse(text=anovacall)

```

```
structure(list(QAICc.table.ordered,eval(anovaexpression)), heading = c("QAICc table\n",paste("Model:",modelcall)),  
class = c("list","character"))  
}
```