Package: ALFAM2 (via r-universe)

September 12, 2024

Type Package

Title Dynamic Model of Ammonia Emission from Field-Applied Manure

Version 4.1.9

Date 2024-08-13

Maintainer Sasha D. Hafner <sasha.hafner@bce.au.dk>

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.0.8), stats

Suggests tinytest, knitr, rmarkdown

Description An implementation of the ALFAM2 dynamic emission model for ammonia volatilization from field-applied animal slurry (manure with dry matter below about 15%). The model can be used to predict cumulative emission and emission rate of ammonia following field application of slurry. Predictions may be useful for emission inventory calculations, fertilizer management, assessment of mitigation strategies, or research aimed at understanding ammonia emission. Default parameter sets include effects of application method, slurry composition, and weather. The model structure is based on a simplified representation of the physical-chemical slurry-soil-atmosphere system. See Hafner et al. (2018) <doi:10.1016/j.atmosenv.2018.11.034> for information on the model and Hafner et al. (2019) <doi:10.1016/j.agrformet.2017.11.027> for more on the measurement data used for parameter development.

License GPL-3

URL https://github.com/AU-BCE-EE/ALFAM2/, https://projects.au.dk/alfam/

VignetteBuilder knitr

LazyData true

LinkingTo Rcpp

Repository https://au-bce-ee.r-universe.dev

RemoteUrl https://github.com/au-bce-ee/alfam2

RemoteRef HEAD

RemoteSha c376bc0c4f5bdfcc0b5963071b3f87c311b04315

Contents

`alfam2` *Predict Ammonia Emission from Field-Applied Manure*

Description

An implementation of the ALFAM2 model for predicting ammonia emission from field-applied manure. The model is described in Hafner et al. (2019).

Usage

```
alfam2(
  dat,
  pars = ALFAM2::alfam2pars03,
  add.pars = NULL,
  app.name = 'TAN.app',
  time.name = 'ct',
  time.incorp = NULL,
  group = NULL,
  center = c(app.rate  = 40,
             man.dm    =  6.0,
             man.tan   =  1.2,
             man.ph    =  7.5,
             air.temp  = 13,
             wind.2m   =  2.7,
             wind.sqrt =  sqrt(2.7),
             crop.z    = 10),
  pass.col = NULL,
  incorp.names = c('incorp', 'deep', 'shallow'),
  prep.dum = TRUE,
  prep.incorp = TRUE,
  add.incorp.rows = FALSE,
  check = TRUE,
  warn = TRUE,
  value = 'emis',
  conf.int = NULL,
  pars.ci = ALFAM2::alfam2pars03var,
  n.ci = NULL,
```

```
  var.ci = 'er',
  ...)
```

Arguments

dat	data frame containing predictor variable values. The data frame must include at least the elapsed (cumulative) time since manure was applied in hours, and the application rate of total ammoniaical nitrogen (TAN) in kg/h. (Other units could be used but should match parameter units and will affect the units of output.) Typically other predictor variables are included. See the details section below and the vignette for more information.
pars	A numeric vector (or a list of vectors) with model parameters (secondary parameters). Three parameter sets are provided with the package: `alfam2pars01`, `alfam2pars02` and `alfam2pars03`, with `alfam2pars03` recommended. The latest set is described in a forthcoming paper. Set 2 is described in Hafner et al. (2021, 2024). The earlier set is described in Hafner et al. (2019). Note that the function could be called with `pars = alfam2pars03` (omitting the `ALFAM2::` bit) but for clarity and safety, and to avoid a package check problem, the package name is included by default. See details for more information.
add.pars	additional parameter values that will extend or overwrite the `pars`
app.name	name of column in `dat` that contains total ammonia nitrogen (TAN) application rate (usually kg/ha)
time.name	name of column in `dat` that contains cumulative time since manure was applied (h)
time.incorp	either name of column in `dat` that contains time at which incorporation occurred (h), or length-one numeric value giving time that incorporation occurred (h). Omit if there was no incorporation or if incorporation is not a predictor variable. Optional.
group	name of column in `dat` that contains a grouping variable for identifying individual plots or locations. Optional.
center	numeric vector with means for centering. Generally should not be changed by users, because parameter values depend on particular centering values, and nonsense predictions can result from changes to even a single centering value. Set to `NULL` to turn off centering, but only do this if you know what you are doing. Internally, supplied values are used to either replace or extend centering values by variable, so it is possible to add a new centering value with e.g., `center = c(x1 = 10)`, which will not affect any of the default values. In contrast, `center = c(wind.2m = 5)` would change the value for the variable `wind.2m`. Default parameters are based on centered values.
pass.col	character vector with name(s) of column(s) in `dat` that should be passed through to the returned data frame.
incorp.names	character vector with name(s) of column(s) in `dat` that contain binary incorporation variables.
prep.dum	if `TRUE` (default), function will automatically prepare dummy variables from input data. If `FALSE`, any necessary dummy variables must already be present in `dat`. See vignette. Length one logical vector.

`prep.incorp` if TRUE (default), function will automatically prepare incorporation inputs from input data. See vignette. Length one logical vector.

`add.incorp.rows` function will add additional rows that exactly match the incorporation time(s) (no more than one per level of `group`) if they are not already present. Should these be returned or left out (default)? Length one logical vector.

`check` should the function check inputs, including for NA values in calculation of primary parameters? Default of TRUE is recommended.

`warn` set to FALSE to suppress some warnings and messages. Doing so is useful for reducing console or report clutter, but use with caution, because problems with the input data or call could be missed. Even with `warn = FALSE` the `alfam2` function may make some substitutions in inputs (see vignette).

`value` type of output. Set to `"incorp"` to return results early with incorporation (or dummy variable) pre-processing and no emission calculations. Output can be used to run `alfam2` with `prep.incorp = FALSE` to speed up evaluation. Otherwise must be `"emis"` for "emission". See vignette. Length one character.

`conf.int` confidence interval setting. Default (NULL) does not return a confidence interval. Use numeric values for confidence interval, e.g., `conf.int = 0.90` for 90% confidence interval. This value will be used with the `quantile` function. Note that with default `pars.ci` like `alfam2pars03var` the returned confidence intervals are an estimate of uncertainty in the *average* response for the particular values provided for input variables. These parameters are based on variability in measured emission among research institutions, and were developed using a bootstrap approach. Set `conf.int` to `'all'` to have the function return all predictions instead of quantile estimates of the confidence interval. This can be useful for incorporation of uncertainty in input variables. See vignette.

`pars.ci` matrix or data frame of parameter sets for confidence interval calculations. See `alfam2pars03var` for an example.

`n.ci` number of parameter sets to use. Defaults to total number available.

`var.ci` calculate confidence intervals for these variables. Calculation is done separately by variable and time interval.

`...` additional optional arguments as length-one vectors that set values of fixed predictor variables. See examples.

Details

Names and units (matching units is essential) for numerical predictors are:

`app.rate.ni` manure application rate, but not for injection (`app.mthd = "os"` or `app.mthd = "cs"`) in t/ha

`man.dm` slurry dry matter, percentage of fresh matter

`man.ph` slurry pH, pH units

`air.temp` air temperature, degress C

`wind.2m` wind speed measured (or adjusted to) 2 m height, m/s

`rain.rate` rainfall rate, mm/h

See the vignette for more details.

Categorical predictor variables can be entered as binary dummy variables or left as character or factors. The alfam2 function automatically creates the dummy variables from from three categorical variables (this can be turned off with prep.dum = FALSE):

app.mthd application method, bc for broadcast, ts for trailing hose, os for open slot injection, and cs for closed slot injection(and see examples and vignette for aliases)

man.source type (source) of manure, pig for pig, otherwise assumed to cattle or other (reference)

incorp incorporation, either shallow or deep (change in levels would require change in incorp.names as well as parameter values)

For parameter set values, see the alfam2pars02, alfam2pars02, or alfam2pars03 objects.

Users are responsible for checking that input variable values are not beyond limits of measurement data used for parameter estimation. For parameter set 3, recommended limits are: DM 1-15% DM, pH 5.5-9.0, air temperature 0-30 deg. C, wind speed 0-10 m/s, rainfall rate 0-2.5 mm/h, and duration 0-168 h.

Value

By default, a data frame with the same number of rows as dat (unless add.incorp.rows = TRUE). First column is time.name column from input data dat. Defaults for following columns are:

dt interval duration (time step)

f0 f0 parameter

r1 r1 parameter

r2 r2 parameter

r3 r3 parameter

r4 r4 parameter

r5 r5 parameter

f fast pool size at ct (kg/ha)

s slow pool size at ct (kg/ha)

j average NH3 flux in interval (kg/ha-h)

jinst instantaneous NH3 flux at given time (kg/ha-h)

ei interval emission (kg/ha)

e cumulative emission (from time = 0 to ct) (kg/ha)

er relative cumulative emission (fraction of applied TAN)

If prep.dum is used, additional dummy variable columns will also be returned. And if a grouping variable is used via group, this column will be included as well. Any columns listed in pass.cols will also be returned.

If value = 'incorp' is used, the function will return intermediate data processed for incorporation but without emission predictions. See vignette.

Author(s)

Sasha D. Hafner, Christoph Haeni, Roland Fuss

References

Hafner, S.D., Pacholski, A., Bittman, S., Carozzi, M., Chantigny, M., Genermont, S., Haeni, C., Hansen, M., Huijsmans, J., Kupper, T., Misselbrook, T., Neftel, A., Nyord, T., Sommer, S. 2019. A flexible semi-empirical model for estimating ammonia volatilization from field-applied slurry. *Atmospheric Environment* **199** 474-484. doi:10.1016/j.atmosenv.2018.11.034

Hafner, S.D., Nyord, T., Sommer, S.G., Adamsen, A.P.S. 2021. Estimation of Danish emission factors for ammonia from field-applied liquid manure for 1980 to 2019. Danish Centre for Food and Agriculture, Aarhus University, Aarhus, Denmark. Report no. 2021-0251862. https://pure.au.dk/portal/files/223538048/EFreport23092021.pdf

Hafner, S.D., Kamp, J.N., Pedersen, J., 2024. Experimental and model-based comparison of wind tunnel and inverse dispersion model measurement of ammonia emission from field-applied animal slurry. Agricultural and Forest Meteorology 344, 109790. doi:10.1016/j.agrformet.2023.109790

The AlFAM2 project website. https://projects.au.dk/alfam/

Examples

```
# Example 1
# Create predictor variable data
dat1 <- data.frame(ctime = 0:12*4, TAN.app = 100, man.dm = 8, air.temp = 15,
     app.mthd = 'trailing shoe')

# Run model, using default parameter values
pred1 <- alfam2(dat1, app.name = 'TAN.app', time.name = 'ctime')
pred1
plot(e ~ ctime, data = pred1, type = 'o')

# For fixed variables (same for all rows), they can be given as optional argument.
dat1b <- data.frame(ctime = 0:12*4)

# Run model, using default parameter values
pred1b <- alfam2(dat1b, app.name = 'TAN.app', time.name = 'ctime',
                 TAN.app = 100, man.dm = 8, air.temp = 15, app.mthd = 'trailing shoe')

all.equal(pred1, pred1b)


# Example 2
# Add incorporation (can occur at any time)
dat2 <- dat1
dat2$incorp <- 'deep'
dat2$t.incorp <- 3.5
dat2

pred2 <- alfam2(dat2, app.name = 'TAN.app', time.name = 'ctime', time.incorp = 't.incorp')
# Note change in r3
pred2
```

```
lines(e ~ ctime, data = pred2, type = 'o', col = 'red')

# Example 3
# Time step doesn't matter
dat3 <- data.frame(ctime = c(0, 48), TAN.app = 100, man.dm = 8, air.temp = 15,
     app.mthd = 'trailing shoe')
pred3 <- alfam2(dat3, app.name = 'TAN.app', time.name = 'ctime')
lines(e ~ ctime, data = pred3, type = 'o', col = 'blue')

# Example 4
# Incorporation does not need to occur at end of interval
dat4 <- dat3
dat4$incorp <- 'deep'
dat4$t.incorp <- 4
pred4 <- alfam2(dat4, app.name = 'TAN.app', time.name = 'ctime', time.incorp = 't.incorp')
lines(e ~ ctime, data = pred4, type = 'o', col = 'orange')

# Incorporation time can be numeric also (not very practical for groups)
alfam2(dat4, app.name = 'TAN.app', time.name = 'ctime', time.incorp = 4)

# To see incorporation time in output, use add.incorp.rows
alfam2(dat4, app.name = 'TAN.app', time.name = 'ctime', time.incorp = 4,
       add.incorp.rows = TRUE)

# Example 5
# Function accepts multiple groups
# Also shown here: some aliases for the different application methods
dat5 <- data.frame(field.plot = 1:5, ctime = 48, TAN.app = 100, man.dm = 5, air.temp = 15,
                   app.mthd = c('bc', 'th', 'ts', 'os', 'cs'), t.incorp = 4)
pred5 <- alfam2(dat5, app.name = 'TAN.app', time.name = 'ctime', group = 'field.plot',
                   time.incorp = 't.incorp')
pred5

# See vignette for more examples and explanation. Run:
# vignette("ALFAM2-start")
```

Index