

Package ‘ALFAM2’

December 6, 2025

Type Package

Title Dynamic Model of Ammonia Emission from Field-Applied Manure

Version 4.2.14

Date 2025-12-05

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. More information is available via citation(“ALFAM2”).

License GPL-3

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

VignetteBuilder knitr

LazyData true

LinkingTo Rcpp

NeedsCompilation yes

Author Sasha D. Hafner [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-0955-0327>>),
Christoph Haeni [aut] (ORCID: <<https://orcid.org/0000-0003-1458-1849>>),
Roland Fuss [aut] (ORCID: <<https://orcid.org/0000-0002-0274-0809>>),
Frederik Dalby [aut] (ORCID: <<https://orcid.org/0000-0003-2437-064X>>),
Johanna Pedersen [ctb] (ORCID: <<https://orcid.org/0000-0002-9026-5031>>),
Valdemar Petersen [ctb]

Repository CRAN
Date/Publication 2025-12-06 13:30:12 UTC

Contents

alfam2	2
alfam2pars	8
alfami	9

Index	12
--------------	-----------

alfam2	<i>Predict Ammonia Emission from Field-Applied Manure</i>
--------	---

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

<code>dat</code>	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 ammonia 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.
<code>pars</code>	A named numeric vector (or a list of vectors) with model parameters (secondary parameters). Three parameter sets are provided with the package: <code>alfam2pars01</code> , <code>alfam2pars02</code> and <code>alfam2pars03</code> , with <code>alfam2pars03</code> recommended. See alfam2pars for details. Note that the function could be called with <code>pars = alfam2pars03</code> (omitting the <code>ALFAM2:</code> 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.
<code>add.pars</code>	additional parameter values that will extend or overwrite the <code>pars</code> . Named numeric vector.
<code>app.name</code>	name of column in <code>dat</code> that contains total ammonia nitrogen (TAN) application rate (usually kg/ha). If omitted, function will return relative emission (fraction of applied TAN) and a warning (if <code>warn = TRUE</code>).
<code>time.name</code>	name of column in <code>dat</code> that contains cumulative time since manure was applied (h)
<code>time.incorp</code>	either name of column in <code>dat</code> 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.
<code>group</code>	name of column in <code>dat</code> that contains a grouping variable for identifying individual plots or locations. Optional.
<code>center</code>	numeric vector with means for centering. Generally should not be changed by users, because parameter values depend on particular centering values, and non-sense predictions can result from changes to even a single centering value. Set to <code>NULL</code> 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., <code>center = c(x1 = 10)</code> , which will not affect any of the default values. In contrast, <code>center = c(wind.2m = 5)</code> would change the value for the variable <code>wind.2m</code> . Default parameters are based on centered values.
<code>pass.col</code>	character vector with name(s) of column(s) in <code>dat</code> that should be passed through to the returned data frame.
<code>incorp.names</code>	character vector with name(s) of column(s) in <code>dat</code> that contain binary incorporation variables.

<code>prep.dum</code>	if TRUE (default), function will automatically prepare dummy variables from input data. If FALSE, any necessary dummy variables must already be present in <code>dat</code> . See vignette. Length one logical vector.
<code>prep.incorp</code>	if TRUE (default), function will automatically prepare incorporation inputs from input data. See vignette. Length one logical vector.
<code>add.incorp.rows</code>	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.
<code>check</code>	should the function check inputs, including for NA values in calculation of primary parameters? Default of TRUE is recommended.
<code>warn</code>	set to FALSE to suppress some warnings and messages. Doing so is useful for reducing console or report clutter, but do so with caution, because problems with the input data or call could be missed. Even with <code>warn = FALSE</code> the <code>alfam2</code> function may make some substitutions in inputs (see vignette).
<code>value</code>	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 <code>alfam2</code> with <code>prep.incorp = FALSE</code> to speed up evaluation. Otherwise must be "emis" for "emission". See vignette. Length one character.
<code>conf.int</code>	confidence interval setting. Default (NULL) does not return a confidence interval. Use numeric values for confidence interval, e.g., <code>conf.int = 0.90</code> for 90% confidence interval. This value will be used with the quantile function. Note that with default <code>pars.ci</code> like <code>alfam2pars03var</code> the returned confidence intervals are an estimate of uncertainty in the <i>average</i> 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 <code>conf.int</code> 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.
<code>pars.ci</code>	matrix or data frame of parameter sets for confidence interval calculations. See <code>alfam2pars03var</code> for an example.
<code>n.ci</code>	number of parameter sets to use. Defaults to total number available.
<code>var.ci</code>	calculate confidence intervals for these variables. Calculation is done separately by variable and time interval.
<code>...</code>	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, degrees 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 three categorical variables (this can be turned off with `prep.dum = FALSE`):

`app.mthd` application method, `bc` for broadcast, `ts` for trailing shoe, `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 [alfam2pars01](#), [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`. All masses are intended to be as kg ammonia N, or TAN (for total ammoniacal nitrogen), but reflect the application rate variable used as input. 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, Pedersen J, Fuss R, Kamp J, Dalby F, Amon B, Pacholski A, Adamsen A, Sommer S., 2025. Improved tools for estimation of ammonia emission from field-applied animal slurry: refinement of the ALFAM2 model and database. *Atmospheric Environment*. doi:10.1016/j.atmosenv.2024.120910
- 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.
- 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/>

See Also

[alfam2pars](#)

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")
```

alfam2pars

*Default Parameter Sets for the ALFAM2 Model***Description**

These default parameter sets can be used to generate average predictions for the `alfam2` function. They are described in the papers cited below.

Usage

```
alfam2pars01
alfam2pars02
alfam2pars03
```

Details

In general, the latest default parameter set (highest version number) should be used. Set 1 was presented in Hafner et al. (2019), 2 in Hafner et al. (2021, 2024), and 3 in Hafner et al. (2025). See vignette for more details.

Author(s)

Sasha D. Hafner, Christoph Haeni, Roland Fuss

References

Hafner S, Pedersen J, Fuss R, Kamp J, Dalby F, Amon B, Pacholski A, Adamsen A, Sommer S., 2025. Improved tools for estimation of ammonia emission from field-applied animal slurry: refinement of the ALFAM2 model and database. *Atmospheric Environment*. doi:10.1016/j.atmosenv.2024.120910

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.

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

```
# To view parameter values
alfam2pars03

# One possible way to facilitate comparison of different sets
```

```

nms <- unique(c(names(alfam2pars01), names(alfam2pars02), names(alfam2pars03)))
pars <- matrix(rep(NA, length(nms) * 3),
              ncol = 3,
              dimnames = list(nms, c('alfam2pars01', 'alfam2pars02', 'alfam2pars03')))
pars[names(alfam2pars01), 1] <- alfam2pars01
pars[names(alfam2pars02), 2] <- alfam2pars02
pars[names(alfam2pars03), 3] <- alfam2pars03
pars

# See vignette for more details on parameters and predictor variables
# vignette("ALFAM2-start")

```

alfami	<i>Emissions Inventory Calculations of Ammonia Emission from Field-Applied Manure</i>
--------	---

Description

A wrapper and higher-level analysis for the [alfam2](#) function for emissions inventory applications. This function is under development and not currently exported! Warning: The API will likely change in the coming months.

Usage

```

alfami(
  dat,
  pars = ALFAM2::alfam2pars03,
  time.name,
  app.tan.name = 'app.tan',
  time.incorp = NULL,
  eventkey = NULL,
  aggkey = NULL,
  uncert = NULL,
  pars.uncert = ALFAM2::alfam2pars03var,
  uncert.settings = NULL,
  nu = 100,
  cl = 0.8,
  seed = NULL,
  quiet = FALSE,
  ...
)

```

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 at which emission should be evaluated, and the quantity of total ammoniaical nitrogen (TAN) applied in t (metric tonnes). (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 vignette for more information.
<code>pars</code>	A named numeric vector (or a list of vectors) with model parameters (secondary parameters). Three parameter sets are provided with the package: <code>alfam2pars01</code> , <code>alfam2pars02</code> and <code>alfam2pars03</code> , with <code>alfam2pars03</code> recommended. See alfam2pars for details. Note that the function could be called with <code>pars = alfam2pars03</code> (omitting the <code>ALFAM2::</code> 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.
<code>time.name</code>	name of column in <code>dat</code> that contains cumulative time since manure was applied (h) for evaluation of emission.
<code>app.tan.name</code>	name of column in <code>dat</code> with total ammoniaical nitrogen (TAN) application (t)
<code>time.incorp</code>	passed to, so as in, alfam2 : either name of column in <code>dat</code> 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.
<code>eventkey</code>	name of column containing a unique key application event key. Unlike alfam2 , each row is otherwise considered a separate application event.
<code>aggkey</code>	name of column or columns that should be used for aggregation of TAN application and emission. Character vector.
<code>uncert</code>	use to include uncertainty based on model parameters (" <code>pars</code> "), input variables (" <code>vars</code> "), both (<code>c("pars", "vars")</code>), or none (<code>NULL</code>). Character vector.
<code>pars.uncert</code>	set of parameter sets to use for model uncertainty. Unusual to specify; use default.
<code>uncert.settings</code>	data frame with uncertainty settings for uncertainty in input variables (when ' <code>vars</code> ' is included in <code>pars.uncert</code>), with these columns: <code>pvar</code> (predictor variable name), <code>rel</code> (relative, absolute, or centered uncertainty), <code>dist.type</code> (normal or uniform distribution), <code>sd</code> (distribution standard deviation), <code>min</code> (uniform distribution minimum), <code>max</code> (uniform distribution maximum).
<code>nu</code>	number of uncertainty iterations.
<code>cl</code>	confidence level.
<code>seed</code>	integer seed for uncertainty iterations. Use for reproducing results. Length-one integer vector.
<code>quiet</code>	set to <code>TRUE</code> to suppress some messages.
<code>...</code>	additional optional arguments to alfam2 .

Details

This function simplifies emission calculations for inventory purposes and can estimate uncertainty from both model input variables and parameter values and aggregate TAN application and emission results. Actual emission calculations are made using `alfam2()`. The name of the function comes from 'ALFAM' and emissions inventory.

Value

A list with six elements, each a data frame with predicted emission and calculated TAN application, but with various additional variables and aggregated in different ways. See vignette.

Author(s)

Sasha D. Hafner

References

The AIFAM2 project website. <https://projects.au.dk/alfam/>

See Also

[alfam2](#) [alfam2pars03](#) [var](#) [alfam2pars](#)

Examples

```
## Not run:
dat <- data.frame(zone = c('north', 'north', 'south', 'east', 'west'),
  sector = c(1, 2, 1, 1, 1),
  man.source = c('cattle', 'pig', 'cattle', 'cattle', 'pig'),
  man.dm = c(7, 4, 6.5, 6.5, 4),
  man.ph = c(7.0, 7.2, 6.9, 7.5, 7.4),
  app.mthd = c(rep('trailing hose', 4), 'broadcast'),
  wind.sqrt = sqrt(c(2.2, 2.1, 4.0, 3.7, 4.0)),
  air.temp = c(12, 8, 8, 15, 14),
  time.h = 168,
  TAN.app.tot = c(2000, 1500, 80, 4300, 25))

dat

ei <- alfami(dat, time.name = 'time.h', app.tan.name = 'TAN.app.tot', aggkey = 'zone')

ei$emistot
ei$emisagg

## End(Not run)
# See vignette for more examples and explanation. Run:
# vignette("ALFAM2-inventory")
```

Index

* **models**

- alfam2, [2](#)
- alfam2pars, [8](#)
- alfami, [9](#)

* **nonlinear**

- alfam2, [2](#)
- alfami, [9](#)

alfam2, [2](#), [8–11](#)

alfam2pars, [3](#), [6](#), [8](#), [10](#), [11](#)

alfam2pars01 (alfam2pars), [8](#)

alfam2pars02, [5](#)

alfam2pars02 (alfam2pars), [8](#)

alfam2pars03, [5](#)

alfam2pars03 (alfam2pars), [8](#)

alfam2pars03var, [11](#)

alfam2pars03var (alfam2pars), [8](#)

alfami, [9](#)