ALFAM2 model fit with parameter set 3

Sasha D. Hafner

17 December, 2024 13:47

1. Summary

Determination of ALFAM2 model performance using parameter set 3, for the 700+ plots used for parameter estimation.

2. Setup

Some functions.

```r
source('functions/interpm.R')
source('functions/dfsumm.R')
source('functions/model_stats.R')
```

Packages.

```r
library(data.table)
library(ggplot2)
```

ALFAM2, check version.

```r
library(ALFAM2)
packageVersion('ALFAM2')
```

```
## [1] '4.2.1'
```

```r
alfam2pars03
```

```
##            int.f0    app.mthd.os.f0    app.mthd.cs.f0 man.source.pig.f0 
##        0.45305451       -2.89718049       -7.09642528       -0.95213804 
##         man.dm.f0            int.r1    app.mthd.bc.r1    app.mthd.ts.r1 
##        0.49956176       -1.45119862        0.73714111       -0.07393662 
##         man.dm.r1         man.ph.r1       air.temp.r1      wind.sqrt.r1 
##       -0.03300931        0.42121280        0.03321186        0.46104870 
##            int.r2      rain.rate.r2            int.r3    app.mthd.cs.r3 
##       -1.16953266        0.60163865       -2.68829766       -0.38439637 
##    incorp.deep.r3         man.ph.r3 incorp.shallow.f4    incorp.deep.f4 
##       -5.35112099        0.11776977       -1.41820869       -2.94966810 
##            int.r5      rain.rate.r5 
##       -1.80000000        0.48425409
```

And it is good practice to check hash, in case someone (me) did a poor job with version numbers.

```
devtools::package_info('ALFAM2')
```

```
##  package * version  date (UTC) lib source
##  ALFAM2  * 4.2.1    2024-12-14 [1] Github (AU-BCE-EE/ALFAM2@3b39aff)
##  Rcpp      1.0.13-1 2024-11-02 [1] CRAN (R 4.4.2)
##
##  [1] /home/sasha/R/x86_64-pc-linux-gnu-library/4.4
##  [2] /usr/local/lib/R/site-library
##  [3] /usr/lib/R/site-library
##  [4] /usr/lib/R/library
```

3. Measurement data

See `data-emission/data/data_version.txt` for version. This was copied directly from the 20204 paper (https://github.com/aU-BCE-EE/hafner-2024-ALFAM2-dev) so is already outdated. Change `rtag` in `data-emission/scripts/main.R` to select a different version.

```
idat <- fread('data-emission/data/ALFAM2_interval.csv')
pdat <- fread('data-emission/data/ALFAM2_plot.csv')
```

The `pmid` keys from the 2024 paper (https://github.com/aU-BCE-EE/hafner-2024-ALFAM2-dev).

```
pmid1 <- fread('pmid/pmid_sub1.txt')[[1]]
```

Subset to subset 1 from 2024 paper.

```
dim(pdat)
```

```
## [1] 2837  220
```

```
pdat <- pdat[pmid %in% pmid1, ]
dim(pdat)
```

```
## [1] 722 220
```

Merge into idat

```
dim(idat)
```

```
## [1] 77398    47
```

```
idat1 <- merge(idat, pdat, by = c('pid', 'pmid'))
dim(idat1)
```

```
## [1] 19230   265
```

New names.

```r
idat1[, wind.sqrt := sqrt(wind.2m)]
idat1[, app.mthd := app.method]
```

Check for missing values.

```r
dfsumm(idat1[, .(app.mthd, man.dm, man.source, man.ph, tan.app)])
```

```
##
##  19230 rows and 5 columns
##  593 unique rows
##                     app.mthd  man.dm man.source  man.ph tan.app
## Class              character numeric  character numeric numeric
## Minimum                   bc   0.772        cat     4.3    10.9
## Maximum                   ts    13.8        pig     8.9     235
## Mean                    <NA>    5.71       <NA>    7.52    69.9
## Unique (excld. NA)         5     276          4      93     571
## Missing values             0       0          0     680       0
## Sorted                 FALSE   FALSE      FALSE   FALSE   FALSE
##
```

```r
dfsumm(idat1[, .(ct, cta, air.temp, wind.sqrt, rain.rate, rain.cum)])
```

```
##
##  19230 rows and 6 columns
##  15041 unique rows
##                         ct     cta air.temp wind.sqrt rain.rate rain.cum
## Class              numeric numeric  numeric   numeric   numeric  numeric
## Minimum               0.15     -65    -4.83         0         0        0
## Maximum                650    1420     35.4      5.33       8.4      113
## Mean                   112     114     12.6      1.49    0.0394     4.08
## Unique (excld. NA)    3114    5834     3184      8382       555      266
## Missing values           0     554        2        13        87       87
## Sorted               FALSE   FALSE    FALSE     FALSE     FALSE    FALSE
##
```

Plenty, as in paper.

Fill in missing pH with institute means from full database

```r
mnph <- pdat[, .(man.ph.mean = mean(na.omit(man.ph)), man.ph.n = sum(is.na(man.ph))), by = inst]
mnph
```

```
##      inst man.ph.mean man.ph.n
##     <int>       <num>    <int>
##   1:   104    7.377500        0
##   2:   202    7.378258        0
##   3:   203    7.500000        0
##   4:   204    7.560000        0
##   5:   205    7.123514        0
##   6:   206    7.796667        0
##   7:   207    7.251111        0
##  8:   208    7.636987        1
##  9:   209    7.526667        0
## 10:   210    5.812500        0
## 11:   212    8.191837        0
## 12:   304    8.360000        0
## 13:   305    7.105000        0
## 14:   214    7.274167       44
```

```
idat1 <- merge(idat1, mnph, by = 'inst')
idat1[is.na(man.ph), man.ph.missing := TRUE]
idat1[is.na(man.ph), man.ph := man.ph.mean]
```

Interpolate missing wind and air temperature values. Set missing rain to 0. Set `cta` to `ct` where missing. And drop obs with `cta < 0`.

```
idat1[, `:=` (interp.wind = is.na(wind.2m), interp.air.temp = is.na(air.temp)), ]
idat1 <- interpm(idat1, 'ct', c('wind.sqrt', 'air.temp'), by = 'pmid', rule = 2)

idat1[is.na(rain.rate), rain.missing := TRUE]
```

```
## Warning in ‘[.data.table‘(idat1, is.na(rain.rate), ‘:=‘(rain.missing, TRUE)):
## Invalid .internal.selfref detected and fixed by taking a (shallow) copy of the
## data.table so that := can add this new column by reference. At an earlier
## point, this data.table has been copied by R (or was created manually using
## structure() or similar). Avoid names<- and attr<- which in R currently (and
## oddly) may copy the whole data.table. Use set* syntax instead to avoid copying:
## ?set, ?setnames and ?setattr. If this message doesn’t help, please report your
## use case to the data.table issue tracker so the root cause can be fixed or this
## message improved.
```

```
idat1[is.na(rain.rate), rain.rate := 0]
idat1[is.na(rain.cum), rain.cum := 0]

idat1[is.na(cta), cta := ct]

idat1 <- idat1[cta > 0, ]
```

Check for other missing values.

```
dfsumm(idat1[, .(app.mthd, man.dm, man.source, man.ph, tan.app)])
```

```
##
##  18732 rows and 5 columns
##  593 unique rows
##                      app.mthd  man.dm man.source  man.ph tan.app
## Class               character numeric  character numeric numeric
## Minimum                    bc   0.772        cat     4.3    10.9
## Maximum                    ts    13.8        pig     8.9     235
## Mean                     <NA>    5.72       <NA>    7.51    69.6
## Unique (excld. NA)          5     276          4      95     571
## Missing values              0       0          0       0       0
## Sorted                  FALSE   FALSE      FALSE   FALSE   FALSE
##
```

```r
dfsumm(idat1[, .(ct, cta, air.temp, wind.sqrt, rain.rate, rain.cum)])
```

```
##
##  18732 rows and 6 columns
##  14618 unique rows
##                           ct     cta air.temp wind.sqrt rain.rate rain.cum
## Class                numeric numeric  numeric   numeric   numeric  numeric
## Minimum                 0.15  0.0167    -4.83         0         0        0
## Maximum                  650    1420     35.4      5.33       8.4      113
## Mean                     114     115     12.6       1.5    0.0392     4.15
## Unique (excld. NA)      3114    5644     3127      8043       554      266
## Missing values             0       0        0         0         0        0
## Sorted                 FALSE   FALSE    FALSE     FALSE     FALSE    FALSE
##
```

Everything is OK now.

4. ALFAM2 predictions

```r
args(alfam2)
```

```
## function (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, 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", ...)
## NULL
```

```r
pred <- alfam2(idat1, pars = alfam2pars03, app.name = 'tan.app', time.name = 'cta',
          group = 'pmid',
          time.incorp = 'time.incorp')
```

```
## Default parameters (Set 3) are being used.
```

```
## Incorporation applied for groups: 1500, 1501, 1506, 1507, 1515, 1516, 1517, 1518, 1522, 1523, 1524,
```

```
## Warning in calcPParms(pars[which5], dat, warn = warn, upr = 100): Some
## calculated primary parameters are at the limit. Check input parameters.
```

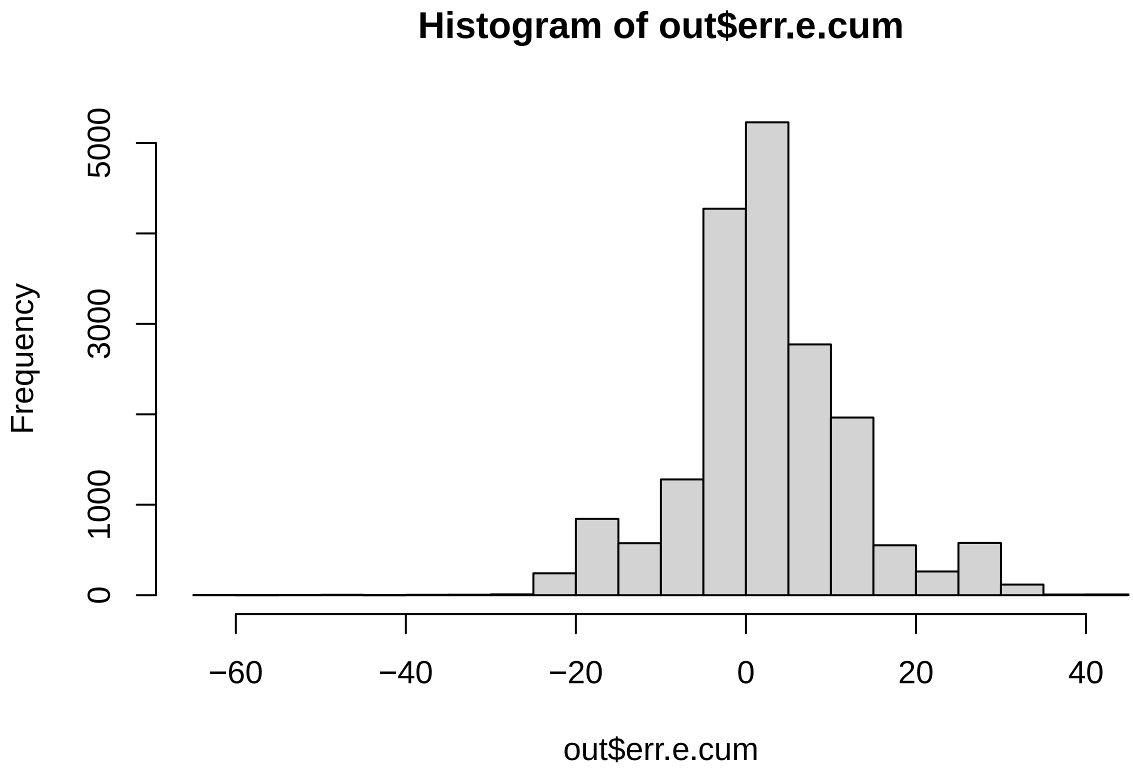
Combine with measurements.

```r
out <- merge(idat1, pred, by = c('pmid', 'cta'), suffixes = c('', '.pred'))
```

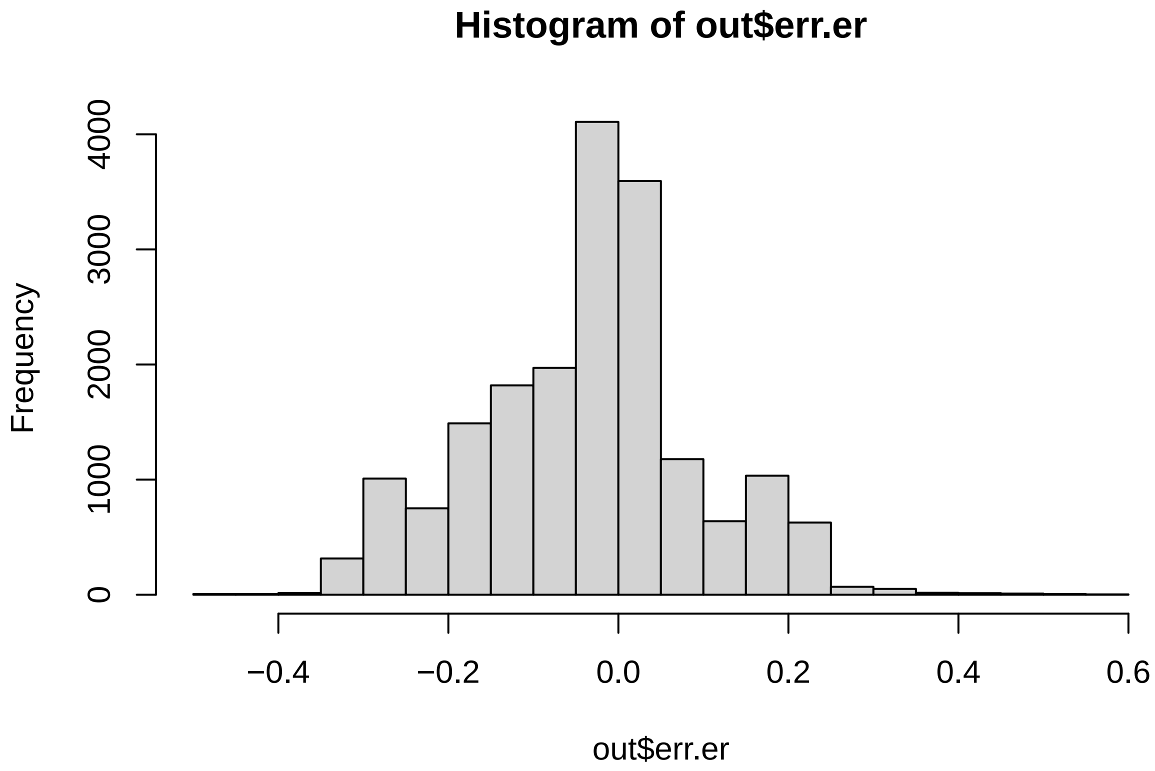
Error.

- in cumulative emission, kg N / ha,
- in relative emission, fraction of applied TAN,
- and relative to measured cumulative emission, as a percentage

```r
out[, err.e.cum := e - e.cum]
out[, err.er := e.rel - er]
out[, err.rel := 100 * err.e.cum / e.cum]
hist(out$err.e.cum)
```



```r
hist(out$err.er)
```

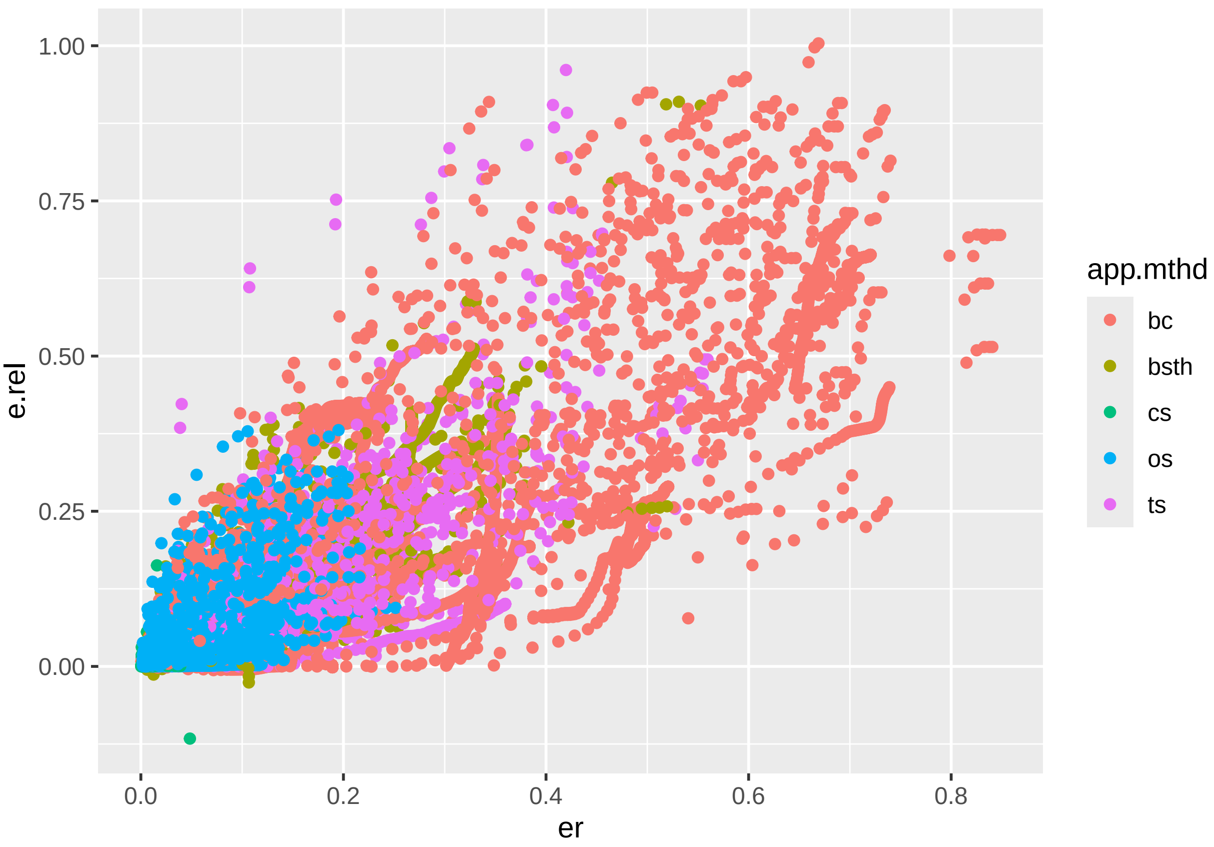


Get latest time.

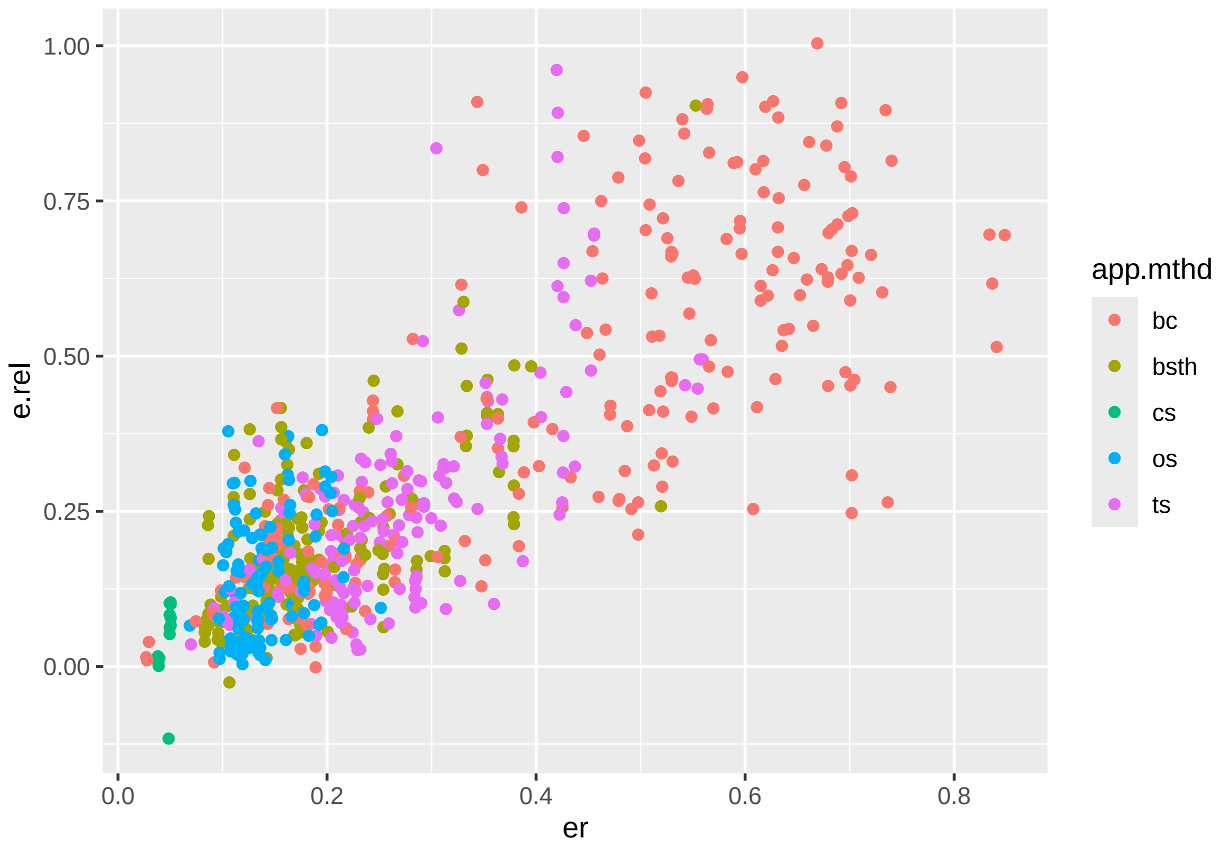
```
out[, cta.max := max(cta), by = .(pmid)]
outfinal <- out[cta == cta.max, ]
```

Take a look.

```
ggplot(out, aes(er, e.rel, colour = app.mthd)) + geom_point()
```



```r
ggplot(outfinal, aes(er, e.rel, colour = app.mthd)) + geom_point()
```



Fit statistics.

Absolute emission kg N / ha

```
rmse(m = outfinal$e.cum, p = outfinal$e)
```

```
## [1] 8.329429
```

```
mae(m = outfinal$e.cum, p = outfinal$e)
```

```
## [1] 5.625063
```

```
me(m = outfinal$e.cum, p = outfinal$e)
```

```
## [1] 0.6648617
```

Relative emission, fraction applied TAN

```
rmse(m = outfinal$e.rel, p = outfinal$er)
```

```
## [1] 0.1274875
```

```
mae(m = outfinal$e.rel, p = outfinal$er)
```

```
## [1] 0.09325539
```

```
me(m = outfinal$e.rel, p = outfinal$er)
```

```
## [1] 0.6632208
```

Mean and median relative error

```
mean(abs(outfinal$err.rel))
```

```
## [1] 94.12122
```

```
median(abs(outfinal$err.rel))
```

```
## [1] 31.63085
```

```
quantile(abs(outfinal$err.rel))
```

```
##           0%          25%          50%          75%         100% 
## 9.144258e-02 1.573304e+01 3.163085e+01 6.106379e+01 1.333879e+04
```

And by application method (low emission methods have high relative error).

```
outfinal[, .(mn = mean(abs(err.rel)), md = median(abs(err.rel))), by = .(app.mthd)]
```

```
##    app.mthd        mn       md
##      <char>     <num>    <num>
## 1:       bc 107.62399 25.55094
## 2:     bsth  51.28340 32.16638
## 3:       ts  64.13843 28.02946
## 4:       os 142.30126 46.83060
## 5:       cs 473.96022 51.16381
```

Also quite poor for broadcast `bc`!