ALFAM2 confidence interval based on parameter uncertainty

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

```
## [1] '3.68'
```

```
#remove.packages('ALFAM2') ; devtools::install_github('sashahafner/ALFAM2', ref = 'dev')
```

```
dat <- data.frame(ctime = 0:84*2, TAN.app = 100, man.dm = 8,
                  air.temp = 7 + 7*sin(0:84*2 * 2*pi/24) + rnorm(85, 0, 2),
                  wind = 10^(0.5 + 0.4*sin(0:84*2 * 2*pi/24) +
                             rnorm(85, 0, 0.12)),
                  app.mthd = "bc")
```

```
#plot(air.temp ~ ctime, data = dat, type = 'o', col = 'gray45')
#plot(wind ~ ctime, data = dat, type = 'o', col = 'blue')
```

Normal call without confidence intervals (CI).

```
pred1 <- alfam2(dat, app.name = 'TAN.app', time.name = 'ctime', warn = FALSE)
head(pred1)
```

```
##   app.mthd.ts app.mthd.bc app.mthd.os app.mthd.cs ctime dt          f        s
## 1           0           1           0           0     0  0 54.8263762 45.17362
## 2           0           1           0           0     2  2 19.3223744 46.05789
## 3           0           1           0           0     4  2 10.2710822 46.31374
## 4           0           1           0           0     6  2  3.4406593 46.31254
## 5           0           1           0           0     8  2  1.2643074 46.18237
## 6           0           1           0           0    10  2  0.6557423 46.01327
##          e      e.int          j        er        f0        r1         r2
## 1  0.00000  0.0000000        NaN 0.0000000 0.5482638 0.1000374 0.01587869
## 2 34.61973 34.6197346 17.3098673 0.3461973 0.5482638 0.5055752 0.01587869
## 3 43.41518  8.7954415  4.3977207 0.4341518 0.5482638 0.3000870 0.01587869
## 4 50.24680  6.8316237  3.4158119 0.5024680 0.5482638 0.5309559 0.01587869
## 5 52.55332  2.3065193  1.1532596 0.5255332 0.5482638 0.4846907 0.01587869
## 6 53.33099  0.7776689  0.3888345 0.5333099 0.5482638 0.3123772 0.01587869
##            r3 f4 r5
## 1 0.002153413  1  0
## 2 0.002153413  1  0
## 3 0.002153413  1  0
## 4 0.002153413  1  0
## 5 0.002153413  1  0
## 6 0.002153413  1  0
```

CI with defaults

Add CI.

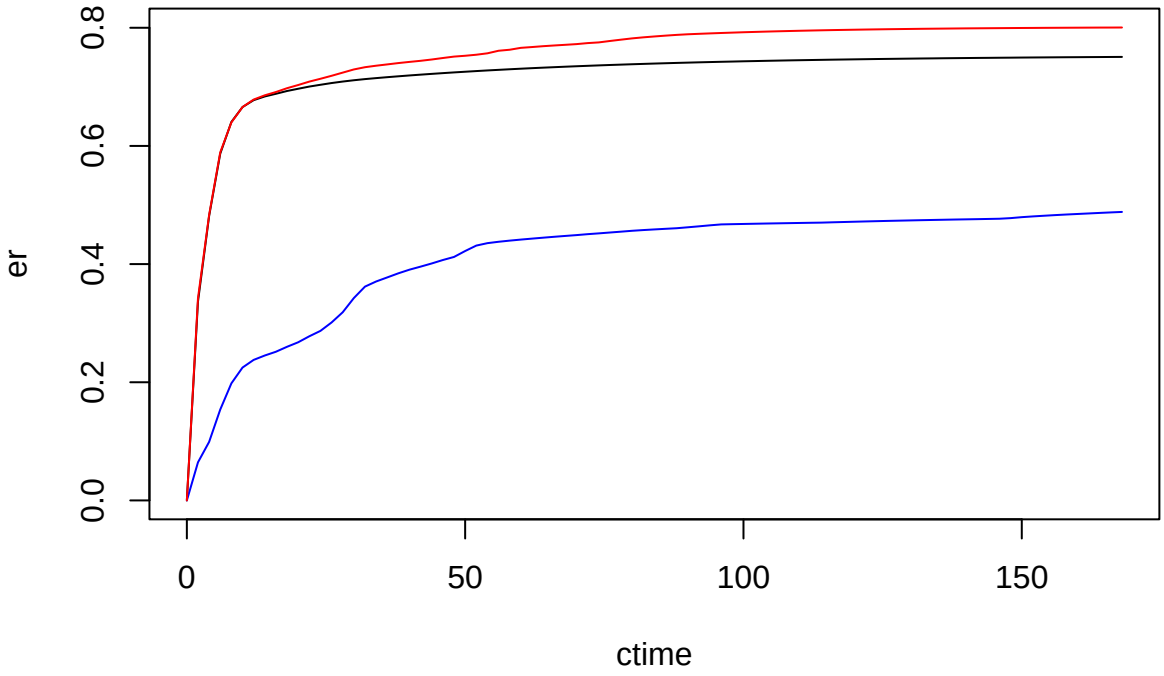
```
predci <- alfam2(dat, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', warn = FALSE, conf.int = 0.90,
                 pars.ci = alfam2pars03var_alpha)
head(predci)
```

```
##    ctime app.mthd.ts app.mthd.bc app.mthd.os app.mthd.cs dt         f        s
## 1      0           0           1           0           0  0 89.331296 10.66870
```

```
## 42     2           0           1           0           0  2 46.446095 19.41055
## 53     4           0           1           0           0  2 27.030394 23.64878
## 64     6           0           1           0           0  2 13.884898 25.45816
## 75     8           0           1           0           0  2  7.293177 25.86690
## 2     10           0           1           0           0  2  4.211828 25.61248
##                e         e.int         j            er       f0        r1
## 1  -8.075431e-16 -8.075431e-16      -Inf -8.075431e-18 0.893313 0.1178079
## 42  3.364960e+01  3.364960e+01 16.824802  3.364960e-01 0.893313 0.2554942
## 53  4.813686e+01  1.448726e+01  7.243630  4.813686e-01 0.893313 0.1991297
## 64  5.868956e+01  1.055270e+01  5.276349  5.868956e-01 0.893313 0.2615445
## 75  6.405658e+01  5.367022e+00  2.683511  6.405658e-01 0.893313 0.2503957
## 2   6.657533e+01  2.518745e+00  1.259373  6.657533e-01 0.893313 0.2029857
##            r2          r3 f4         r5        er.lwr       er.upr
## 1  0.07153553 0.004662259  1 0.01584893 -1.799140e-17 2.151580e-17
## 42 0.07153553 0.004662259  1 0.01584893  6.442025e-02 3.420993e-01
## 53 0.07153553 0.004662259  1 0.01584893  9.908436e-02 4.832279e-01
## 64 0.07153553 0.004662259  1 0.01584893  1.537739e-01 5.890970e-01
## 75 0.07153553 0.004662259  1 0.01584893  1.981282e-01 6.398058e-01
## 2  0.07153553 0.004662259  1 0.01584893  2.248480e-01 6.657544e-01
```

By default only returned for variable `er` = relative cumulative emission.

```
plot(er ~ ctime, data = predci, type = 'l', ylim = c(0, max(predci$er.upr)))
lines(er.lwr ~ ctime, data = predci, type = 'l', col = 'blue')
lines(er.upr ~ ctime, data = predci, type = 'l', col = 'red')
```



Add variables

Can add any variables for CI calculation, but `quantile` is applied by variable.

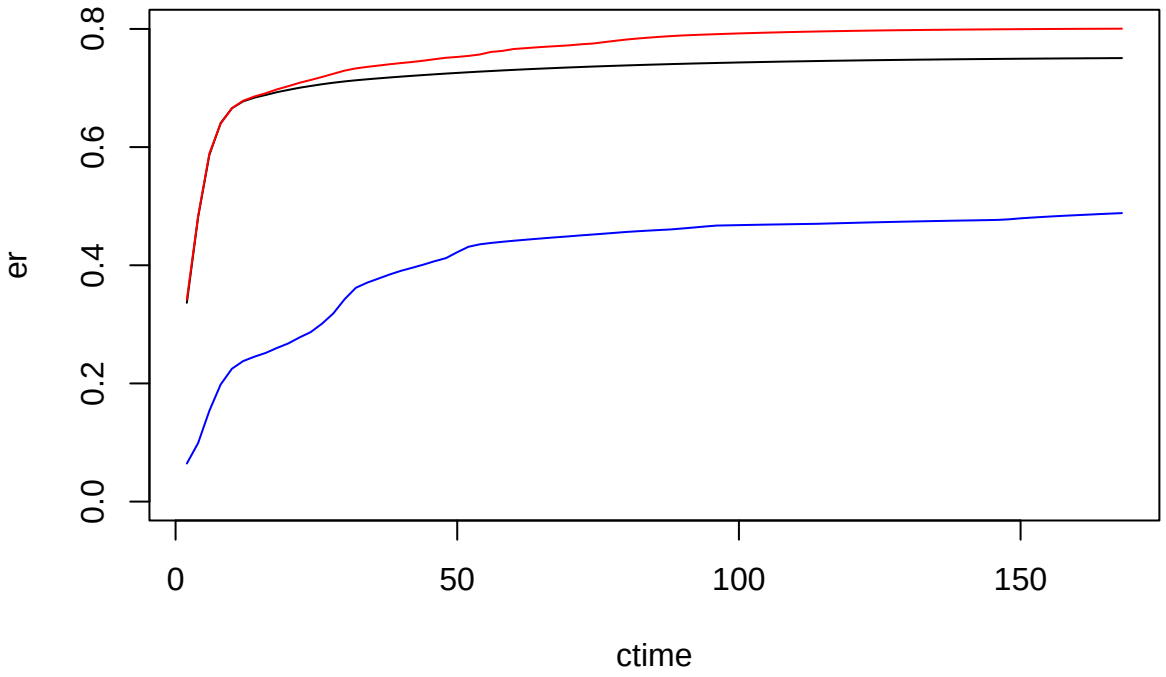
```
predci <- alfam2(dat, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', warn = FALSE, conf.int = 0.90,
                 pars.ci = alfam2pars03var_alpha, var.ci = c('er', 'j', 'r1'))
```

```
head(predci)
```

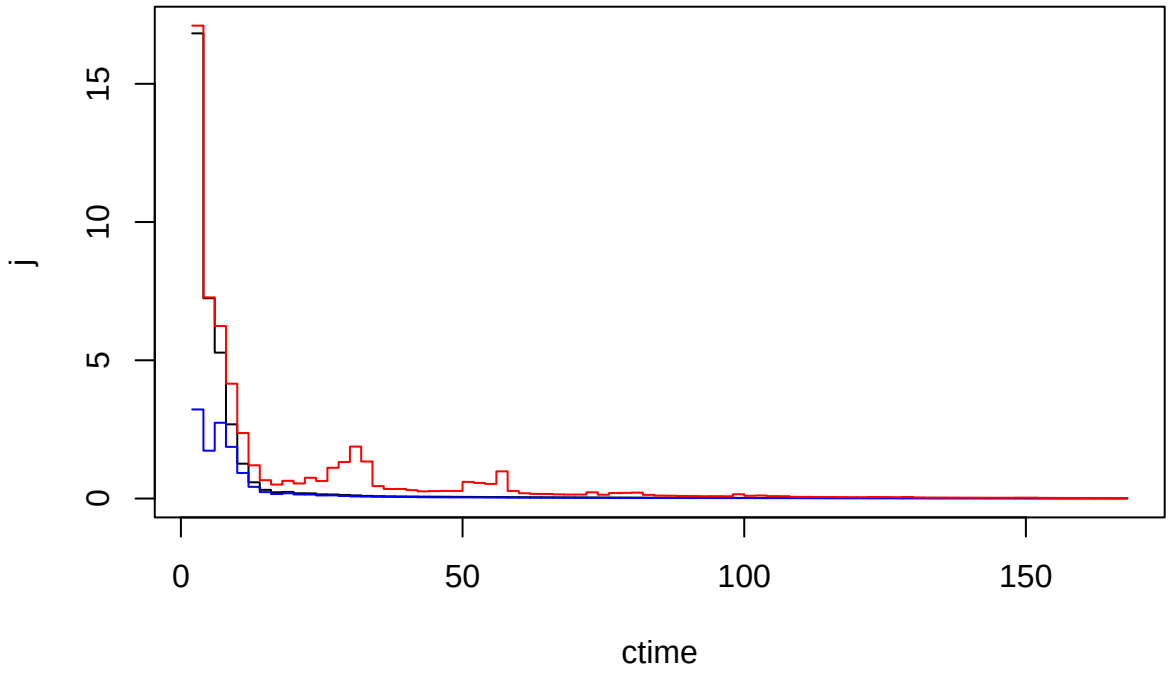
```
##    ctime app.mthd.ts app.mthd.bc app.mthd.os app.mthd.cs dt         f        s
## 41     2           0           1           0           0  2 46.446095 19.41055
## 52     4           0           1           0           0  2 27.030394 23.64878
## 63     6           0           1           0           0  2 13.884898 25.45816
## 74     8           0           1           0           0  2  7.293177 25.86690
## 1     10           0           1           0           0  2  4.211828 25.61248
## 12    12           0           1           0           0  2  2.782222 25.06566
##           e     e.int         j        er       f0        r1         r2
## 41 33.64960 33.649603 16.824802 0.3364960 0.893313 0.2554942 0.07153553
## 52 48.13686 14.487260  7.243630 0.4813686 0.893313 0.1991297 0.07153553
## 63 58.68956 10.552698  5.276349 0.5868956 0.893313 0.2615445 0.07153553
## 74 64.05658  5.367022  2.683511 0.6405658 0.893313 0.2503957 0.07153553
## 1  66.57533  2.518745  1.259373 0.6657533 0.893313 0.2029857 0.07153553
## 12 67.74808  1.172746  0.586373 0.6774808 0.893313 0.1357879 0.07153553
##             r3 f4         r5     er.lwr     j.lwr     r1.lwr    er.upr
## 41 0.004662259  1 0.01584893 0.06442025 3.2210127 0.03967091 0.3420993
## 52 0.004662259  1 0.01584893 0.09908436 1.7289820 0.02487925 0.4832279
## 63 0.004662259  1 0.01584893 0.15377392 2.7396257 0.04181163 0.5890970
## 74 0.004662259  1 0.01584893 0.19812821 1.8683027 0.03791594 0.6398058
## 1  0.004662259  1 0.01584893 0.22484800 0.9240107 0.02553395 0.6657544
## 12 0.004662259  1 0.01584893 0.23783068 0.4222397 0.01044036 0.6785059
##        j.upr    r1.upr
## 41 17.104964 0.2822901
## 52  7.275783 0.2186697
## 63  6.241112 0.2885673
## 74  4.153142 0.2769944
## 1   2.368303 0.2235291
## 12  1.200779 0.1573230
```

Note that times with any `NaN` etc. in one of `var.ci` columns will be dropped. So here flux `j` is undefined for first time interval (0 - 0 h).

```
plot(er ~ ctime, data = predci, type = 'l', ylim = c(0, max(predci$er.upr)))
lines(er.lwr ~ ctime, data = predci, type = 'l', col = 'blue')
lines(er.upr ~ ctime, data = predci, type = 'l', col = 'red')
```




```
plot(j ~ ctime, data = predci, type = 's', ylim = c(0, max(predci$j.upr)))
lines(j.lwr ~ ctime, data = predci, type = 's', col = 'blue')
lines(j.upr ~ ctime, data = predci, type = 's', col = 'red')
```



Test on multiple groups

Here is a test where each group has a different incorporation time

```
datm <- data.frame(scenario = 1:6, ctime = 168, TAN.app = 50,
                   man.dm = 8, air.temp = 20, wind.2m = 4,
                   app.mthd = 'bc',
```

```
                  incorp = 'deep',
                  t.incorp = c(0.1, 1, 6, 24, 168, NA))
```

```
predci <- alfam2(datm, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', time.incorp = 't.incorp', group = 'scenario',
                 conf.int = 0.90,
                 pars.ci = alfam2pars03var_alpha, var.ci = c('er'))
```

```
## User-supplied parameters are being used.
```

```
## Incorporation skipped where it occurred after all intervals, for groups: 5.
```

```
## Incorporation applied for groups: 1, 2, 3, 4.
```

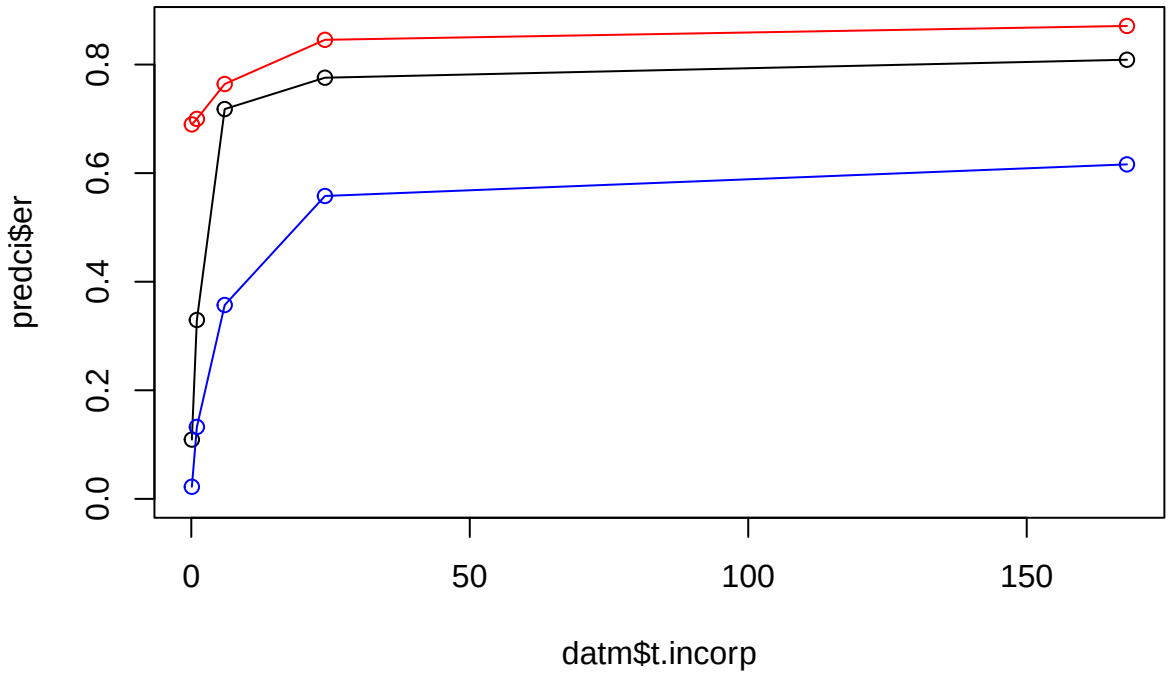
```
## Warning in alfam2(dat = dat, pars = pars, add.pars = add.pars, app.name = app.name, : Running with 18
## These secondary parameters have been dropped:
##   app.rate.ni.f0
##   man.source.pig.f0
##   man.ph.r1
##   rain.rate.r2
##   man.ph.r3
##   rain.rate.r5
##   wind.sqrt.r1
```

```
predci
```

```
##   scenario ctime app.mthd.ts app.mthd.bc app.mthd.os app.mthd.cs incorp.shallow
## 1        1   168           0           1           0           0              0
## 2        2   168           0           1           0           0              0
## 3        3   168           0           1           0           0              0
## 4        4   168           0           1           0           0              0
## 5        5   168           0           1           0           0              0
## 6        6   168           0           1           0           0              0
##   incorp.deep  dt            f         s         e     e.int          j
## 1           1 168 9.047443e-34 3.1123681  5.459711  5.459711 0.03249828
## 2           1 168 9.047443e-34 2.3691465 16.474979 16.474979 0.09806535
## 3           1 168 9.047443e-34 1.0132177 35.899590 35.899590 0.21368803
## 4           1 168 9.047443e-34 0.7839046 38.788686 38.788686 0.23088504
## 5           1 168 8.708448e-33 0.4007291 40.443414 40.443414 0.24073461
## 6           1 168 8.708448e-33 0.4007291 40.443414 40.443414 0.24073461
##          er       f0        r1         r2           r3        f4         r5
## 1 0.1091942 0.893313 0.3904896 0.07153553 2.989078e-06 0.1038927 0.01584893
## 2 0.3294996 0.893313 0.3904896 0.07153553 2.989078e-06 0.1038927 0.01584893
## 3 0.7179918 0.893313 0.3904896 0.07153553 2.989078e-06 0.1038927 0.01584893
## 4 0.7757737 0.893313 0.3904896 0.07153553 2.989078e-06 0.1038927 0.01584893
## 5 0.8088683 0.893313 0.3904896 0.07153553 4.662259e-03 1.0000000 0.01584893
## 6 0.8088683 0.893313 0.3904896 0.07153553 4.662259e-03 1.0000000 0.01584893
##       er.lwr    er.upr
## 1 0.02222003 0.6897585
## 2 0.13248098 0.6999392
## 3 0.35707203 0.7641794
## 4 0.55788859 0.8455538
## 5 0.61613940 0.8710960
## 6 0.61613940 0.8710960
```

Plot emission versus time of incorporation.

```
plot(datm$t.incorp, predci$er, type = 'o', ylim = c(0, max(predci$er.upr)))
lines(datm$t.incorp, predci$er.lwr, type = 'o', col = 'blue')
lines(datm$t.incorp, predci$er.upr, type = 'o', col = 'red')
```



Limit number of iterations

By default the model is run with all the different parameter sets provided in `pars.ci`.

```
dim(alfam2pars03var_alpha)
```

```
## [1] 100  25
```

For speed some users might want to sometimes reduce that.

```
predci <- alfam2(datm, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', time.incorp = 't.incorp', group = 'scenario',
                 conf.int = 0.90, pars.ci = alfam2pars03var_alpha,
                 n.ci = 10)
```

```
## User-supplied parameters are being used.

## Incorporation skipped where it occurred after all intervals, for groups: 5.

## Incorporation applied for groups: 1, 2, 3, 4.

## Warning in alfam2(dat = dat, pars = pars, add.pars = add.pars, app.name = app.name, : Running with 18
## These secondary parameters have been dropped:
##   app.rate.ni.f0
##   man.source.pig.f0
##   man.ph.r1
##   rain.rate.r2
##   man.ph.r3
##   rain.rate.r5
##   wind.sqrt.r1
```

```
predci
```

```
##   scenario ctime app.mthd.ts app.mthd.bc app.mthd.os app.mthd.cs incorp.shallow
## 1        1   168           0           1           0           0              0
## 2        2   168           0           1           0           0              0
## 3        3   168           0           1           0           0              0
## 4        4   168           0           1           0           0              0
## 5        5   168           0           1           0           0              0
## 6        6   168           0           1           0           0              0
##   incorp.deep  dt            f         s         e     e.int          j
## 1           1 168 9.047443e-34 3.1123681  5.459711  5.459711 0.03249828
## 2           1 168 9.047443e-34 2.3691465 16.474979 16.474979 0.09806535
## 3           1 168 9.047443e-34 1.0132177 35.899590 35.899590 0.21368803
## 4           1 168 9.047443e-34 0.7839046 38.788686 38.788686 0.23088504
## 5           1 168 8.708448e-33 0.4007291 40.443414 40.443414 0.24073461
## 6           1 168 8.708448e-33 0.4007291 40.443414 40.443414 0.24073461
##          er       f0        r1         r2           r3        f4         r5
## 1 0.1091942 0.893313 0.3904896 0.07153553 2.989078e-06 0.1038927 0.01584893
## 2 0.3294996 0.893313 0.3904896 0.07153553 2.989078e-06 0.1038927 0.01584893
## 3 0.7179918 0.893313 0.3904896 0.07153553 2.989078e-06 0.1038927 0.01584893
## 4 0.7757737 0.893313 0.3904896 0.07153553 2.989078e-06 0.1038927 0.01584893
## 5 0.8088683 0.893313 0.3904896 0.07153553 4.662259e-03 1.0000000 0.01584893
## 6 0.8088683 0.893313 0.3904896 0.07153553 4.662259e-03 1.0000000 0.01584893
##       er.lwr    er.upr
## 1 0.01851698 0.5963537
## 2 0.13707414 0.6704356
## 3 0.51303989 0.8389531
## 4 0.71950943 0.8557512
## 5 0.75368463 0.8736830
## 6 0.75368463 0.8736830
```

Get all predictions

When `ci=` some number it is used in the `quantile` function. To get all results, use `conf.int = 'all'`. This could be useful to combine uncertainty in parameter values with uncertainty in inputs. Of course then the CIs would have to be constructed outside the `alfam2` function, but it cannot do everything, so I think this is OK.

```
datvar <- data.frame(ctime = 168, TAN.app = 50, man.dm = rnorm(7, mean = 8, sd = 1), air.temp = 20, wind
```

Here is what we get with numeric `conf.int`.

```
predci <- alfam2(datvar, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', group = 'man.dm',
                 conf.int = 0.90, pars.ci = alfam2pars03var_alpha)
```

```
## User-supplied parameters are being used.
```

```
## Warning in prepDat(dat, value = "dummy", warn = warn): Argument prep.dum = TRUE but there are no vari
##   Ignoring prep.dum = TRUE.
```

```
## Warning in alfam2(dat = dat, pars = pars, add.pars = add.pars, app.name = app.name, : Running with 8
## These secondary parameters have been dropped:
##   app.mthd.os.f0
##   app.rate.ni.f0
##   man.source.pig.f0
##   app.mthd.cs.f0
```

```
##   app.mthd.bc.r1
##   app.mthd.ts.r1
##   man.ph.r1
##   rain.rate.r2
##   app.mthd.bc.r3
##   app.mthd.cs.r3
##   man.ph.r3
##   incorp.shallow.f4
##   incorp.shallow.r3
##   incorp.deep.f4
##   incorp.deep.r3
##   rain.rate.r5
##   wind.sqrt.r1
```

```
predci
```

```
##      man.dm ctime  dt            f        s        e    e.int         j
## 3 1.7922042   168 168 2.493279e-10 1.438748 26.48209 26.48209 0.1576315
## 2 1.7705438   168 168 2.411854e-10 1.438581 26.47561 26.47561 0.1575929
## 6 2.9845227   168 168 1.366801e-09 1.471672 26.40863 26.40863 0.1571942
## 7 3.0418268   168 168 1.474604e-09 1.474245 26.38714 26.38714 0.1570663
## 4 2.6853443   168 168 9.119067e-10 1.459528 26.49740 26.49740 0.1577226
## 5 2.8751675   168 168 1.180794e-09 1.466975 26.44575 26.44575 0.1574152
## 1 0.4861935   168 168 2.851687e-11 1.464812 25.44066 25.44066 0.1514325
##          er        f0         r1         r2          r3 f4         r5    er.lwr
## 3 0.5296418 0.8822404 0.08262510 0.07153553 0.002038241  1 0.01584893 0.4346500
## 2 0.5295123 0.8810306 0.08281456 0.07153553 0.002038241  1 0.01584893 0.4347227
## 6 0.5281725 0.9341330 0.07283757 0.07153553 0.002038241  1 0.01584893 0.4273565
## 7 0.5277428 0.9359952 0.07239753 0.07153553 0.002038241  1 0.01584893 0.4268586
## 4 0.5299480 0.9235687 0.07517874 0.07153553 0.002038241  1 0.01584893 0.4297615
## 5 0.5289149 0.9304389 0.07368473 0.07153553 0.002038241  1 0.01584893 0.4282750
## 1 0.5088133 0.7883163 0.09486145 0.07153553 0.002038241  1 0.01584893 0.4015666
##      er.upr
## 3 0.7564399
## 2 0.7548414
## 6 0.8035365
## 7 0.8062387
## 4 0.7886906
## 5 0.7980466
## 1 0.6841236
```

Not so useful.

The alternative:

```
predci <- alfam2(datvar, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', group = 'man.dm',
                 conf.int = 'all', pars.ci = alfam2pars03var_alpha, n.ci = 3)
```

```
## User-supplied parameters are being used.
```

```
## Warning in prepDat(dat, value = "dummy", warn = warn): Argument prep.dum = TRUE but there are no vari
##   Ignoring prep.dum = TRUE.
```

```
## Warning in alfam2(dat = dat, pars = pars, add.pars = add.pars, app.name = app.name, : Running with 8
## These secondary parameters have been dropped:
##   app.mthd.os.f0
```

```
##   app.rate.ni.f0
##   man.source.pig.f0
##   app.mthd.cs.f0
##   app.mthd.bc.r1
##   app.mthd.ts.r1
##   man.ph.r1
##   rain.rate.r2
##   app.mthd.bc.r3
##   app.mthd.cs.r3
##   man.ph.r3
##   incorp.shallow.f4
##   incorp.shallow.r3
##   incorp.deep.f4
##   incorp.deep.r3
##   rain.rate.r5
##   wind.sqrt.r1
```

```
predci
```

```
##       man.dm ctime  dt            f         s        e    e.int         j
## 1  1.7922042   168 168 2.597760e-06 0.6964122 38.47319 38.47319 0.2290071
## 2  1.7705438   168 168 2.358333e-06 0.6958518 38.46099 38.46099 0.2289345
## 3  2.9845227   168 168 2.103210e-04 0.8253176 37.81890 37.81890 0.2251125
## 4  3.0418268   168 168 2.496007e-04 0.8361002 37.73266 37.73266 0.2245992
## 5  2.6853443   168 168 8.152998e-05 0.7756443 38.19814 38.19814 0.2273699
## 6  2.8751675   168 168 1.503387e-04 0.8058677 37.97169 37.97169 0.2260220
## 7  0.4861935   168 168 1.848229e-09 0.7921692 35.69738 35.69738 0.2124844
## 8  1.7922042   168 168 1.283148e-06 1.1619636 31.74045 31.74045 0.1889313
## 9  1.7705438   168 168 1.231451e-06 1.1606056 31.74615 31.74615 0.1889652
## 10 2.9845227   168 168 9.831997e-06 1.2732465 30.84511 30.84511 0.1836019
## 11 3.0418268   168 168 1.073135e-05 1.2800874 30.77931 30.77931 0.1832102
## 12 2.6853443   168 168 6.135964e-06 1.2394547 31.16046 31.16046 0.1854789
## 13 2.8751675   168 168 8.299230e-06 1.2605109 30.96605 30.96605 0.1843217
## 14 0.4861935   168 168 7.866464e-08 1.1400500 31.07854 31.07854 0.1849913
## 15 1.7922042   168 168 8.141222e-06 1.3875954 28.05190 28.05190 0.1669756
## 16 1.7705438   168 168 8.086366e-06 1.3886923 28.01956 28.01956 0.1667831
## 17 2.9845227   168 168 1.129034e-05 1.3633267 28.97922 28.97922 0.1724954
## 18 3.0418268   168 168 1.145105e-05 1.3634103 28.99415 28.99415 0.1725843
## 19 2.6853443   168 168 1.046836e-05 1.3642715 28.86842 28.86842 0.1718358
## 20 2.8751675   168 168 1.098668e-05 1.3633891 28.94545 28.94545 0.1722943
## 21 0.4861935   168 168 4.898355e-06 1.5262387 24.41013 24.41013 0.1452984
##           er        f0         r1         r2          r3 f4         r5
## 1  0.7694638 0.8998149 0.08115774 0.01805249 0.002442449  1 0.01584893
## 2  0.7692199 0.8983356 0.08172351 0.01805249 0.002442449  1 0.01584893
## 3  0.7563779 0.9565878 0.05536725 0.01805249 0.002442449  1 0.01584893
## 4  0.7546532 0.9583441 0.05435896 0.01805249 0.002442449  1 0.01584893
## 5  0.7639629 0.9462132 0.06094320 0.01805249 0.002442449  1 0.01584893
## 6  0.7594338 0.9530382 0.05734360 0.01805249 0.002442449  1 0.01584893
## 7  0.7139477 0.7706760 0.12337950 0.01805249 0.002442449  1 0.01584893
## 8  0.6348090 0.9242524 0.06573340 0.03783474 0.002233886  1 0.01584893
## 9  0.6349230 0.9232319 0.06597160 0.03783474 0.002233886  1 0.01584893
## 10 0.6169023 0.9643962 0.05386549 0.03783474 0.002233886  1 0.01584893
## 11 0.6155863 0.9656891 0.05335247 0.03783474 0.002233886  1 0.01584893
## 12 0.6232092 0.9568493 0.05662513 0.03783474 0.002233886  1 0.01584893
## 13 0.6193209 0.9617978 0.05485824 0.03783474 0.002233886  1 0.01584893
```

```
## 14 0.6215707 0.8359043 0.08175368 0.03783474 0.002233886  1 0.01584893
## 15 0.5610380 0.9206104 0.04973920 0.04280767 0.002553698  1 0.01584893
## 16 0.5603912 0.9189667 0.04976881 0.04280767 0.002553698  1 0.01584893
## 17 0.5795845 0.9753273 0.04813640 0.04280767 0.002553698  1 0.01584893
## 18 0.5798831 0.9767066 0.04806068 0.04280767 0.002553698  1 0.01584893
## 19 0.5773684 0.9667326 0.04853366 0.04280767 0.002553698  1 0.01584893
## 20 0.5789090 0.9724708 0.04828122 0.04280767 0.002553698  1 0.01584893
## 21 0.4882026 0.7516292 0.05155615 0.04280767 0.002553698  1 0.01584893
```

21 rows here, small so we can look at the results.

More plausible usage would have at least 100 of each I suppose, for 10000 rows in the output.

```
datvar <- data.frame(ctime = 168, TAN.app = 50, man.dm = rnorm(100, mean = 8, sd = 1), air.temp = 20, wi
```

```
predci <- alfam2(datvar, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', group = 'man.dm',
                 conf.int = 'all', pars.ci = alfam2pars03var_alpha, n.ci = 100)
```

```
## User-supplied parameters are being used.
```

```
## Warning in prepDat(dat, value = "dummy", warn = warn): Argument prep.dum = TRUE but there are no vari
##   Ignoring prep.dum = TRUE.
```

```
## Warning in alfam2(dat = dat, pars = pars, add.pars = add.pars, app.name = app.name, : Running with 8
## These secondary parameters have been dropped:
##   app.mthd.os.f0
##   app.rate.ni.f0
##   man.source.pig.f0
##   app.mthd.cs.f0
##   app.mthd.bc.r1
##   app.mthd.ts.r1
##   man.ph.r1
##   rain.rate.r2
##   app.mthd.bc.r3
##   app.mthd.cs.r3
##   man.ph.r3
##   incorp.shallow.f4
##   incorp.shallow.r3
##   incorp.deep.f4
##   incorp.deep.r3
##   rain.rate.r5
##   wind.sqrt.r1
```

```
head(predci)
```

```
##      man.dm ctime  dt            f        s        e    e.int         j
## 1 0.9711918   168 168 8.427467e-11 1.449705 25.94337 25.94337 0.1544248
## 2 0.9152482   168 168 7.849614e-11 1.451028 25.89866 25.89866 0.1541587
## 3 0.3996563   168 168 3.998614e-11 1.468902 25.38154 25.38154 0.1510806
## 4 2.0231558   168 168 2.980813e-10 1.444355 26.42069 26.42069 0.1572660
## 5 2.4960762   168 168 5.046661e-10 1.452152 26.44469 26.44469 0.1574089
## 6 2.7086392   168 168 6.345306e-10 1.457311 26.42446 26.42446 0.1572885
##          er        f0         r1         r2          r3 f4         r5
## 1 0.5188674 0.8516291 0.08652143 0.07388547 0.002086003  1 0.01584893
## 2 0.5179732 0.8479711 0.08691861 0.07388547 0.002086003  1 0.01584893
## 3 0.5076307 0.8107109 0.09066612 0.07388547 0.002086003  1 0.01584893
## 4 0.5284137 0.9077413 0.07938168 0.07388547 0.002086003  1 0.01584893
```

```
## 5 0.5288938 0.9261255 0.07636692 0.07388547 0.002086003  1 0.01584893
## 6 0.5284893 0.9332388 0.07504944 0.07388547 0.002086003  1 0.01584893
```

And then, externally, for a 95% confidence interval that includes uncertainty in both inputs (only DM here) and parameters:

```
quantile(predci$er, c(0.05, 0.95))
```

```
##        5%       95%
## 0.4077394 0.7716059
```

Error messages

```
predci <- alfam2(datvar, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', group = 'man.dm',
                 conf.int = 'all', n.ci = 100)
```

```
## Error: Expect class "data.frame, matrix, array" for argument pars.ci but got "NULL".
```

```
predci <- alfam2(datvar, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', group = 'man.dm',
                 conf.int = 'all', pars.ci = alfam2pars03var_alpha, n.ci = 1000)
```

```
## Error: Expect values within the range "0, 100" for argument n.ci but got "1000, 1000".
```

```
predci <- alfam2(datvar, pars = alfam2pars03_alpha, app.name = 'TAN.app',
                 time.name = 'ctime', group = 'man.dm',
                 conf.int = 'all', pars.ci = alfam2pars03var_alpha, n.ci = 10,
                 var.ci = 'blahblah')
```

```
## Error: Expect one of the following values "f0, r1, r2, r3, f4, r5, f, s, j, ei, e, er" for argument
```