CoTiMA User's Guide:
A package for R to perform Continuous Time Meta-Analysis

Christian Dormann & Markus Homberg

07/02/21

Chair of Business Education & Managemen, Johannes Gutenberg University Mainz, CoTiMa@uni-mainz.de

Contents

1 General description

Continuous time meta-analysis (CoTiMA) performs meta-analyses of correlation matrices of repeatedly measured variables. Since variables are measured at discrete time points (e.g., today at 4pm, next week on Monday etc.) this imposes a problem for meta-analysis of studies that repeatedly measured the variables because the time intervals between measurement could vary across studies. However, so-called continuous time math can be used to extrapolate or interpolate the results from all studies to any desired time interval. By this, effects obtained in studies that used different time intervals can be meta-analyzed[1].

A critical assumption is the validity of the underlying causal model that describes the investigated process. CoTiMA is based on a rather general model, which can be restricted on demand. For instance, for a causal system that describes how a single variable that is measured repeatedly (e.g., x_1, x_2, x_3, etc.) develops over time, the default CoTiMA model assumes that x_1 affects x_2, x_2 affects x_3 and so forth. This is called a first order autoregressive structure. In a two-variable model of x and y, the underlying CoTiMA model is a cross-lagged model with autoregressive effects for x and y and, in addition, a cross-lagged effect of x_t on y_{t+1} and of y_t on x_{t+1}. More complex models (e.g., including x_t on y_{t+1} and x_t on y_{t+2}) can also be meta-analyzed, but they require user-specific adaptations. Restricted versions of the default CoTiMA model (e.g., x_t on y_{t+1} but not y_t on x_{t+1}) are easier to implement and several specific models (e.g., x_t on y_{t+1} exactly of the same size as y_t on x_{t+1}) could be optionally requested. Correlations of primary studies serve as an input for CoTiMA and synthesized (i.e., meta-analytically aggregated) effect sizes represent the output of CoTiMA.

```
library(devtools)
install_github('CoTiMA/CoTiMA')
library(CoTiMA)
```

Figure 1: Installing CoTiMA from GitHub

CoTiMA is a package for R (R Core Team, 2020). It can be downloaded from CRAN (https://cran.r-project.org) using (`install.packages('CoTiMA')`). After the devtools R package is installed (`install.packages('devtools')`), the latest version of the R package CoTiMA can be installed from our GitHub repository (Dormann & Homberg, 2020) using the code shown in Figure 1. All code and examples shown in the User's Guide were performed and tested with R version 4.0.3 and run using RStudio (RStudio Team, 2020).

The next six pages show how to conduct a CoTiMA. This involves several

---

[1]In a nutshell, CoTiMA fits models to empirical data using the structural equation model (SEM) package ctsem. The effects specified in a SEM are related (contrained) to parameters that are not directly included in the model (i.e., *continuous time parameters*; together, they represent the *continuous time structural equation model*, CTSEM) which is done in a fashion similar to other SEM programs (e.g., like $a = b \times c$ to test for mediation in MPLUS) using matrix algebra functions (e.g., matrix exponentiation, which is not available in MPLUS), and statistical model comparisons and significance tests are performed on the continuous time parameter estimates. For details see Dormann, Guthier, and Cortina (2020).

steps starting with entering primary study information (correlations etc.), initial fitting of a ctsem model to each primary study, fitting the CoTiMA, and plotting the results[2].

2 A CoTiMA Example

To prepare a CoTiMA, users have to supply information about the primary studies to be meta-analyzed. Primary study information has to be stored into 'objects' (everything in R is an object). Some objects have pre-defined names and are either always mandatory (`delta_t`i), mostly mandatory (`sampleSize`i, `empcov`i), or optional (`pairwiseN`i, `studyNumber`i, `moderator`i, etc., with i indicating the study number). User-defined object names could be added (e.g., `criticalRemark`i).

```
empcov2 <- matrix(c(1.00, 0.45, 0.57, 0.18,
                    0.45, 1.00, 0.31, 0.66,
                    0.57, 0.31, 1.00, 0.40,
                    0.18, 0.66, 0.40, 1.00), nrow=4, ncol=4)
delta_t2 <- 12
sampleSize2 <- 148

empcov3 <- matrix(c(1.00, 0.43, 0.71, 0.37,
                    0.43, 1.00, 0.34, 0.69,
                    0.71, 0.34, 1.00, 0.50,
                    0.37, 0.69, 0.50, 1.00), nrow=4, ncol=4)
delta_t3 <- 12
sampleSize3 <- 88

empcov313 <- matrix(c(1.00, 0.38, 0.54, 0.34, 0.60, 0.28,
                      0.38, 1.00, 0.34, 0.68, 0.28, 0.68,
                      0.54, 0.34, 1.00, 0.47, 0.66, 0.39,
                      0.34, 0.68, 0.47, 1.00, 0.38, 0.72,
                      0.60, 0.28, 0.66, 0.38, 1.00, 0.38,
                      0.28, 0.68, 0.39, 0.72, 0.38, 1.00), nrow=6, ncol=6)
delta_t313 <- c(1.5, 1.5)
sampleSize313 <- 335
```

Figure 2: Entering information of three primary studies

The present example of a small but nevertheless full CoTiMA is based on two variables (Variable 1 = V1, Variable 2 = V2). They were measured in three primary studies. The cross-lagged effects of earlier V1 on later V2 *(V1toV2)*

---

[2]When it is desired, all R objects created in the following examples (e.g., `empocv2`, `delta_t2`, etc. in Figure 2 or `CoTiMAstudyList_3` in Figure 3) can be created in the user's R environment in two ways. First, the code could be copied directly from this User Guide and then run. Second, the objects are 'invisible' but actually available in the package:CoTiMA environment. For example, `empcov2 <- empcov2` copies `empcov2` into the global environment. Afterwards, `rm(empcov2)` removes `empcov2` from the global environment, but it still available in the package:CoTiMA environment. Objects that are available in the package:CoTiMA environment only but nut in the global environment are not used when the user performs any CoTiMA analyses.

and vice versa *(V1toV2)* are meant to be meta-analyzed. The first two studies, which are numbered '2' and '3' in our database, both comprise two variables measured at two measurement occasions, which results in a correlation matrix with four rows (`nrow=4`) and four columns (`ncol=4`; i.e., a 4×4 correlation matrix; see Figure 2). The correlations reported in primary studies are stored in the objects `empcov2` and `empcov3`, respectively. The third Study 313 has three waves of measurements and the empirical correlation matrix of Study 313 has, therefore, 6×6 entries. The order of the variables has to be V1 at Time 0, V2 at Time 0, V1 at Time 1, and V2 at Time 1 etc. Note that in the continuous time literature it is common to number time points starting with 0. In the automatically generated output files, these two variables are labeled "V1" and "V2". The matrices have to be symmetric. Lack of symmetry is automatically detected by CoTiMA, a warning is issued, and processing is interrupted.

In addition to correlation matrices, a CoTiMA requires further information. Researchers need to provide time intervals (`delta_t`i) and sample sizes (`sampleSize`i). Primary Study 2 had a time lag of 12 months, which is stored in the object `delta_t2` (see Figure 2). One could also use 1.0 to indicate a 1.0 year lag. Any time scale is possible, but it has to be used consistently across primary studies. It is recommended using a time scale that allows assigning a value of 6 or less to the longest of all time intervals. For example, if the longest time lag was 10 years, it is recommended to use the number of 5-year intervals as the time scale, and to assign the value 2 to `delta_t`i if Study i had a 10-year interval. Since Study 313 had three waves of observations, the corresponding two time intervals have to be provided as vector (`delta_t313 <- c(1.5, 1.5)`).

Primary Study 2 further had a sample size of 148, which is stored in the object `sampleSize2` (not `sampleSize0`2). In cases in which the correlation matrix includes correlations based on pairwise deletion of missing values, sample sizes vary between correlations, too. This could be specified as explained later.

```
CoTiMAstudyList_3 <- ctmaPrep(selectedStudies = c(2, 3, 313))
```
Figure 3: Compiling a list of primary studies (`ctmaPrep`)

After all primary study information was entered, the next step is to compile them into a list[3] and store this list as an R object. This is done with the `ctmaPrep` function included in CoTiMA. The created list object (e.g., `CoTiMA-studyList_3` in Figure 3) could be inspected as we shall see later. For the moment, it is sufficient to just have it available. Note that all functions provided by the CoTiMA R package start with 'ctma' such as `ctmaPrep`. In general, we label the objects to store the results delivered by the ctma-functions starting with "CoTiMA", such as `CoTiMAstudyList_3`.

After a list of primary study information has been complied with `ctmaPrep`, the next step is to fit a ctsem model to each primary study in a series of separate

---

[3]A list is a particular R object that is useful to collect a variety of information such as values, vectors, matrices, names etc.)

```
activeDirectory <- "/Users/cdormann/CoTiMA Users Guide/"
coresToUse = 1      # SET THIS TO 1 INSTEAD OF -1 IF YOU ARE USING WINDOWS

CoTiMAInitFit_3 <- ctmaInit(primaryStudies = CoTiMAstudyList_3,
                            n.latent = 2,
                            activeDirectory = activeDirectory,
                            coresToUse = coresToUse)

summary(CoTiMAInitFit_3)
saveRDS(CoTiMAInitFit_3, paste0(activeDirectory, "CoTiMAInitFit_3.rds"))
```

Figure 4: Fitting a ctsem model to each primary study (`ctmaInit`).

models. This step is mandatory for subsequent CoTiMA for several reasons. One of the most important reason is that at this stage one could check the results and identify possible problems with the data entered.

The use of `ctmaInit` is shown in Figure 4. Before using `ctmaInit`, specify the `activeDirectory` (where to save results) and the number of computer cores to be used (-1 indicates all cores except 1). This can then be copied into all subsequent function calls. `ctmaInit` requires the number of latent variables per measurement occasion (in most cases probably identical to the overall number of variables) to be provided by the user as well as an `activeDirectory`, which is where `ctmaInit` will save the fitted models. In this case, the fitted models in stored in an Init-Fit object named `CoTiMAInitFit_3`, which can be saved to disk with `saveRDS`. Using `summary(CoTiMAInitFit_3)` displays the results, of which we selected the most intersting results in Figure 5[4], [5].

```
##                                    V1toV1     SE        V2toV1    SE
## Study No 2   "Reference not provided" "-0.0537" "0.0111" "0.0117" "0.0105"
## Study No 3   "Reference not provided" "-0.0323" "0.0134" "0.0049" "0.0106"
## Study No 313 "Reference not provided" "-0.4126" "0.0452" "0.1453" "0.0376"
##              V1toV2     SE       V2toV2     SE
## Study No 2   "-0.0202" "0.009"  "-0.0254" "0.0078"
## Study No 3   "0.0102"  "0.0097" "-0.0358" "0.0097"
## Study No 313 "0.0939"  "0.0377" "-0.2823" "0.0359"
```

Figure 5: ctsem results (`summary(CoTiMAInitFit_3)`).

The output in Figure 5 displays the so-called drift effects. The two auto effects (*V1toV1* & *V2toV2*) are negative as one would expect in continuous time

---

[4]Note that if computations are reproduced, the results are unlikely to be exactly the same because parameters and their standard errors are drawn from 1.000 parameter samples by default for final results computation. Using the argument `finishsamples = 10000` or larger numbers would make results reproducible with any desired precision.

[5]Note that CoTiMA could handle a range of time intervals by default; we have experienced few problems with time intervals from .01 to 16. However depending on the time scale used, one has to use the argument `scaleTime = 1/t`, with t being a reasonable number. For instance, suppose one's time scale is one week. An average year has $(365.25)/4 = 52.1786$ weeks. If the longest time interval among the primary studies was 16years, this would correspond to `delta_t`$i = 834.8576$. Using `scaleTime = 1/52.1786` would probably ensure getting proper results. Contrary, without adapting the time scale, one would probably see several warning or error messages during the fitting process, and the results should not be trusted.

modelling - we explain this later in Section 6. The two cross effects are mostly positive. Note that an effect is regarded as significant if its size is more than 1.96 times its standard error *(SE)*. However, confidence intervals (or credible intervals in case Bayesian estimation is chosen) are easily available, too, and should be preferred. All of this – and a bit more – is displayed after entering `summary(CoTiMAInitFit_3)`. We do not show the output here due to space reasons. The study numbers are repeated as row names. "Reference not provided" just indicates that, yes, we did not provide a reference for each study, which would improve readability of the table. We explain later how to provide references for labeling the output.

The fit object `CoTiMAInitFit_3` resulting from applying the `ctmaInit` function to the list of compiled studies `CoTiMAstudyList_3` can then be used for aggregating (i.e., meta-analyzing) drift effects, performing moderator analyses, estimating publication bias, calculation of expected power and required samples sizes for different time intervals, plotting, and much more. In virtually all cases, the CoTiMA functions to perform these tasks take `CoTiMAInitFit_3` as the first (and only required) argument.

A full CoTiMA, with 'full' indicating that all drift parameters are simultaneously aggregated, is conducted by the code in Figure 6. Note that the `activeDirectory` is copied from `CoTiMAInitFit` if not provided as an argument to `ctmaFit`.

```
CoTiMAFullFit_3 <- ctmaFit(ctmaInitFit = CoTiMAInitFit_3,
                           coresToUse = coresToUse)
summary(CoTiMAFullFit_3)
```

Figure 6: Conducting a full CoTiMA (`ctmaFit`)

The summary function displays a couple of results that we present here in reduced format and in two subsequent steps. Results not shown here are explained later in Section 7.

We reduced the `$estimates` section in Figure 7 compared to the actual output displayed on screen. Reason is that among the whole lot of estimates presented, only the four drift effects are of major interest. These are the meta-analytically aggregated effects as indicated by the additional label 'invariant'. Invariant means that this effect does not vary among primary studies and only one overall effect is estimated. This is similar to fixed effect analysis, where it is also assumed that a single overall (true) effect exists. This is what one usually wants from CoTiMA. We are done. All drift effects are significant by means of the T-values as well as by virtue of their confidence intervals.

```
##                         row col    Mean     sd     2.5%     50%    97.5% Tvalues
## DRIFT V1toV1 (invariant)  1   1 -0.3867 0.0464 -0.4841 -0.3841 -0.3024 -8.3280
## DRIFT V2toV1 (invariant)  1   2  0.2291 0.0327  0.1657  0.2289  0.2942  7.0092
## DRIFT V1toV2 (invariant)  2   1  0.1158 0.0350  0.0468  0.1156  0.1841  3.3087
## DRIFT V2toV2 (invariant)  2   2 -0.1378 0.0341 -0.2142 -0.1341 -0.0824 -4.0385
```

Figure 7: First part of summary(`CoTiMAFullFit_3`)

```
## $minus2ll
## [1] 7325.971
##
## $n.parameters
## [1] 22
##
## $opt.lag
##      [,1] [,2]
## [1,]   NA    6
## [2,]    6   NA
##
## $max.effects
##          [,1]    [,2]
## [1,]       NA 0.3656
## [2,] 0.1848      NA
```

Figure 8: Second part of summary(`CoTiMAFullFit_3`)

The second part of the output generated by `summary(CoTiMAFullFit_3)` is shown in Figure 8. It displays the minus 2 loglikelihood (-2ll) value and the number of estimated parameters (both become important when researchers want to compare nested models), and the optimal lag in terms of Dormann and Griffin (2015) across which the effects become largest. The previous output in Figure 7 informed us that the effect of *V1toV2* is located in Row 2 and Column 1 and, conversely, the effect of *V2toV1* is located in Row 1 and Column 2. In this case, the optimal lag is six months for both effects, where the effects (see `$max.effects`) become .3669 for *V2toV1* and .1850 for *V1toV2*. The latter seems to be much smaller than the former, and we explain later how to test this statistically.

Effects in continuous time are difficult to interpret. Therefore, they are usually back-translated into discrete time. More specifically, they are usually back-translated into the cross-lagged regression coefficients that can be expected across a range of different time intervals. This is achieved when plotting a CoTiMA fit object (or several of the CoTiMA fit objects in the same plot).

```
plot(ctmaFitList(CoTiMAInitFit_3, CoTiMAFullFit_3),
     timeUnit="Months",
     timeRange=c(1, 144, 1) )
```

Figure 9: Plotting a Full CoTiMA (`plot`)

Figure 9 shows how to plot both the effects of the three separately fitted primary studies and the aggregated effect into single figures. Actually, since there are four effects (auto effect *V1toV1*, auto effect *V2toV2*, cross effect *V1toV2*, and cross effect *V2toV1*), four figures will be created. To tell the plot function that these are multiple CoTiMA fit objects, they have to be combined using the CoTiMA function `ctmaFitList`. For labeling of the x-axis the time unit is specified by `timeUnit="Months"` ranging from 1 to 144 in 1-month steps (the smaller the steps, the smoother the plot). For the effect *V2toV1*, the resulting plot is shown in Figure 10.
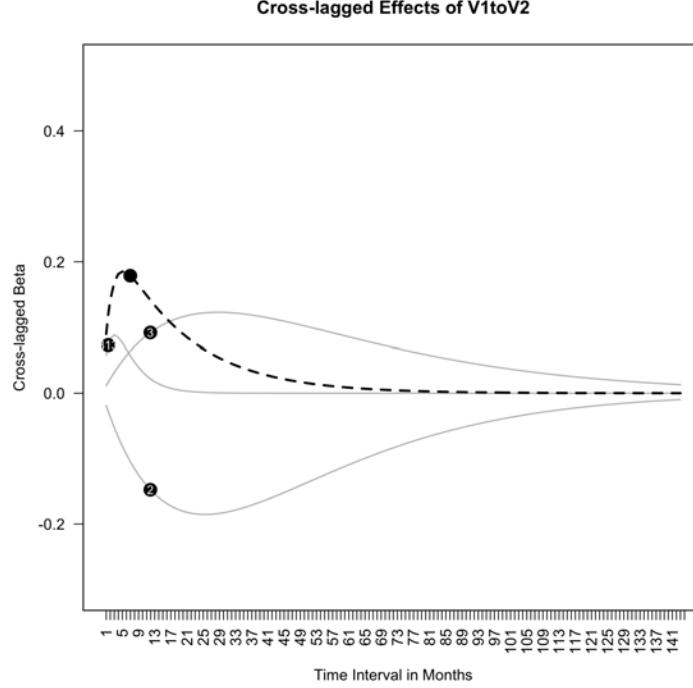
Figure 10: The cross-lagged effect *V1toV2* across 1 to 144 months.

As can be seen, the dashed black line that represents the aggregated effect reaches its maximum across time intervals of 6 month, where it can be expected to be .3669 (see Figure 8), and then becomes smaller eventually approaching zero. It is noteworthy, albeit not occurring very often and probably limited to CoTiMAs with very few primary studies, that the aggregated effect does not always have to be somewhere in between the smallest and largest effects observed among the primary studies. CoTiMA does not aggregate by taking a (weighted) average of single effects. Rather, it optimizes estimates of all effects simultaneously by minimizing the loglikelihood value of the fit function, and the single set of the two auto effects and the two cross effects best explains the observed correlations across the three primary studies.

CoTiMA could be used for much more than demonstrated up to this point. Capabilities include traditional fixed and random effects analyses, analyses of publication biases, assessing heterogeneity, comparing effect sizes within models, moderator analysis, and analysis of statistical power. However, for CoTiMA like for any kind of meta-analysis, the most time consuming work is data collection and data management. Therefore, the two next sections deal with this topic. We make several recommendations of how to proceed and we introduce further functions and capabilities of CoTiMA which could make the life of a meta-analyst more convenient. Subsequent sections then address additional types of

analyses that could be conducted after a full CoTiMA.

3 EPIC-BiG-Power: A Recommended CoTiMA Workflow

Our recommended CoTiMA workflow can be summarized with the acronym EPIC-BiG-Power, which stands for **E**xtract, **P**repare, **I**nitFit, **C**oTiMAs, **Bi**as & **G**eneralizability, and statistical **Power**.

1. **E**PIC: **E**xtract correlations from the literature and save them to disk. There are no particular `ctmaXxxx` function available supporting this step. It is hard work! We make some suggestions in Section 4.

2. E**P**IC: In a **P**reparatory step, modify correlations, add further study information, and compile lists of primary studies to be used for subsequent analysis using `ctmaPrep` (and `ctmaEmpCov` if useful). This is elaborated in Section 5.

3. EP**I**C: Perform a series of Initial fits, in which each primary study out of the lists of primary studies is used to fit a ctsem model using `ctmaInit`. This is demonstrated in Section 6.

4. EPI**C**: The fit object delivered in Step 3 is typically used to perform a CoTiMA using `ctmaFit`. This is the core of CoTiMA!

   (a) We show how to perform a full CoTiMA in Subsection 7.1, in which an entire drift matrix is aggregated.

   (b) In Subsection 7.2 a partial CoTiMA is demonstrated, in which subsets of drift coefficients are aggregated.

   (c) To address the question whether two (or more) drift effects (e.g., the 2 cross effects) estimated in Step 4(b) are identical, or if one effect is significantly larger than the other one, use the CoTiMA fit object delivered in Step 4(b) and `ctmaEqual` to test this, and see Subsection 7.3 for details.

   (d) To address the question whether one (or more) drift effects are moderated by certain characteristics of the primary studies (e.g., the year when they were published), use the CoTiMA fit object delivered in Step 3 and `ctmaFit` to test this. See Subsection 7.4.

5. **BiG**: Analysis of publication **Bi**as including possible corrections and can also be performed. Further, various measure of heterogeneity, which allow answering the question if effects could be **G**eneralized, are reported. This also involves z-curve analysis. Classical fixed and random effects of each single drift effect (not as a set) are estimated, too. Use the CoTiMA fit object delivered in Step 3 and `ctmaBiG` to test this. This is demonstrated in Section 8.

6. **Power**: Calculation of the statistical (post hoc) **P**ower of each single primary study (using the CoTiMA results as true effect estimates) as well as required sample sizes for future studies using a range of different time intervals could be performed using by the CoTiMA fit object delivered in Step 3 and `ctmaPower`. This is demonstrated in Section 9.

7. Results of the different analyses could be plotted with `plot(CoTiMaFit-Objects)`. Funnel and forest plots will be created if `CoTiMaFitObjects` is a CoTiMA fit object delivered by `ctmaBiG`. Plots of required sample sizes are delivered if `CoTiMaFitObjects` is a CoTiMA fit object delivered by `ctmaPower`. Discrete time cross-lagged and autoregressive effect size plots will be created if `CoTiMaFitObjects` is a CoTiMA fit object delivered by `ctmaInit` or `ctmaFit`. This is demonstrated throughout Section 6 – Section 9.

4   Extraction of Correlations from the Literature

In the previous example, we used only the mandatory objects (`delta_t`i) and objects that are probably required in most instances (`sampleSize`i, `empcov`i). We show later how data management can be improved by using further objects. This section starts, however, with some recommendations and helpful functions that can make data entry easier and offer new possibilities.

One of the most laborious steps is entering the correlation matrices of primary studies. Although it would be less laborious to enter only lower triangular correlation matrices, the requirement to have full correlation matrices serves to double check if correlations are entered correctly. Small typographical errors could have large consequences such as time-consuming and poor convergence in fitting the model to the data. Although it is preferred to analyze correlation matrices in meta-analyses rather than covariances, the option to analyze covariances is available; CoTiMA automatically switches to the analysis of covariances if vectors of variances (`empVar`i) are provided. This is, however, not recommended because different variances imply that effect sizes between studies are on different scales, making aggregated effects impossible to interpret. Similarly, empirical mean values for all variables (`empMeans`i) could be provided, but we do address these possibilities here.

Figure 11 shows an example of how to enter and save correlation matrices. We recommend entering them as they are published and not change any signs or skip variables. This could be easily done later. Although it is no formal requirement, we also recommend labeling the variables (i.e., the row names and column names of the matrices) as they are labelled by the authors of the primary studies. The correlation matrices including the labels are then saved. For demonstrational purposes, we change the original matrix by deleting one variable from the matrix shown in Figure 11. In the original study (Childs & Stoeber, 2012, Study 1), the variable 'role stress_2' was available, but sometimes researchers do not measure all variable at all time points. When conducting a

```
empcov128 <- matrix(c(
        1.00,  0.48,  0.50,        0.50,  0.43,  0.40,  0.39, -0.51, -0.45,
        0.48,  1.00,  0.17,        0.23,  0.22,  0.00,  0.01, -0.10, -0.08,
        0.50,  0.17,  1.00,        0.63,  0.42,  0.45,  0.44, -0.52, -0.41,

        0.50,  0.23,  0.63,        1.00,  0.65,  0.59,  0.50, -0.50, -0.37,
        0.43,  0.22,  0.42,        0.65,  1.00,  0.49,  0.64, -0.41, -0.41,
        0.40,  0.00,  0.45,        0.59,  0.49,  1.00,  0.75, -0.54, -0.46,
        0.39,  0.01,  0.44,        0.50,  0.64,  0.75,  1.00, -0.48, -0.57,
       -0.51, -0.10, -0.52,       -0.50, -0.41, -0.54, -0.48,  1.00,  0.70,
       -0.45, -0.08, -0.41,       -0.37, -0.41, -0.46, -0.57,  0.70,  1.00), 9, 9)
pairwiseN128 <- matrix(c(
        100,  99,  88,       77,  66,  55,  44,  33,  22,
         99,  99,  99,       88,  77,  66,  55,  44,  33,
         88,  99,  88,       99,  88,  77,  66,  55,  44,

         77,  88,  99,       77,  99,  88,  77,  66,  55,
         66,  77,  88,       99,  66,  99,  88,  77,  66,
         55,  66,  77,       88,  99,  55,  99,  88,  77,
         44,  55,  66,       77,  88,  99,  44,  99,  88,
         33,  44,  55,       66,  77,  88,  99,  33,  99,
         22,  33,  44,       55,  66,  77,  88,  99,  22), 9, 9)
variableNames128 <- c("SPP_1", "SOP_1",
                      "role stress_1",
                      "exhaustion_1", "exhaustion_2",
                      "cynicism_1", "cynicism_2", "efficacy_1", "efficacy_2")
dimnames(empcov128) <- list(variableNames128, variableNames128)
saveRDS(empcov128, file=paste0(activeDirectory, "empcov128.rds"))
saveRDS(pairwiseN128, file=paste0(activeDirectory, "pairwiseN128.rds"))
```

Figure 11: Entering correlation matrices

CoTiMA one has to deal with missing data (correlations) then. To demonstrate how this could be achieved, we deleted 'role stress_2' from the original matrix, which we visualize in Figure 11 by blanks after the column and row containing the correlations for 'role stress_1'.

A further possible challenge for CoTiMA are correlation matrices reported in primary studies that are based on pairwise deletion of missing values. One possible problem is that such matrices might not be suited at all for analysis if they or not positive definite. This cannot happen with listwise deletion. A not positive definite matrix is given, for example, if the correlation between A and B is $r = .90$, between A and C $r = .80$, and between B and C $r = .10$ - given the two large correlations, such a small correlation is impossible if all correlations are based on identical samples. If a matrix is not positive definite, we recommend contacting the authors of the primary study and ask for a correlation matrix based on listwise deletion, or for raw data. Another option is to drop one or more variables from the correlation matrix. One could check if the matrix is positive definite after dropping variables; the code `eigen(empcov128)$values` should deliver only positive eigenvalues then.

A second challenge resulting from pairwise deletion of missing values in primary studies is the sample size to be used for CoTiMA. Sometimes, authors re-

port the range of pairwise N (e.g., pairwise $N = 22$ to 100) in a table note. We recommend using the smallest value then (e.g., `sampleSize128 = 22`). Sometimes, however, authors report pairwise N for each correlation. Thus, we also have a matrix of pairwise N, which we illustrate in Figure 11. Recall that we also have to deal with the entirely missing variable 'role_stress2', which we again visualize by inserted blanks and an empty row in the matrix of pairwise N. Using a matrix of pairwise N rather than just the smallest of all N increases the statistical power of a CoTiMA. We recommend saving the matrix to disk. In case there is a matrix of pairwise N this could also be saved.

5 Preparatory Step (`ctmaEmpCov`, `ctmaCorRel`, `ctmaPrep`)

CoTiMA uses correlation matrices to generate 'pseudo raw data' using the MASS R package (Veneables & Ripley, 2002). Pseudo raw data exactly (!) reproduce the correlation matrices and offers a couple of interesting options. In the present section we show how data can be processed in terms of recoding variables, combining two or more variables into composite scores, and dealing with missing correlations. (The possibility to combine two or more correlations into a single correlation will be added soon.)

We turn now to processing the correlations shown in Figure 11. Our aim is to analyze the reciprocal effects between job demands and burnout. In particular, we (1) want to correct the correlations for unreliability (aka correction for attenuation, disattenuation). Further, we (2) want to drop the variables 'SPP_1' and 'SOP_1' because these variables do not exist in other primary studies and because they are not of particular interest. We also (3) want to recode 'efficacy_1' and 'efficacy_2' so that they represent lack of efficacy rather than efficacy. Lack of efficacy, cynicism, and exhaustion are the three burnout symptoms, and we (4) want to combine them into a single variable[6]. Whereas a measure of demands is available for the first measurement occasion ('role stress 1'), such a measure is missing at the second measurement occasion. Thus, we (5) also have to deal with missing correlations.

First, to achieve our aims, we start with reading the previously saved correlation matrix and the matrix of pairwise N (see Figure 12) and assign them to R objects `empcov128` and `pairwiseN128`. With `colnames(empcov128)` (not shown in Figure 12) we could recall the variable names, which are 'SPP_1', 'SOP_1', 'role stress_1', 'exhaustion_1', 'exhaustion_2', 'cynicism_1', 'cynicism_2', 'efficacy_1', and 'efficacy_2'. First, we do the corrections for unreliability. This has to be done first because, for example, reliabilities would be no longer available after two or more variables are combined. To correct for

---

[6]CoTiMA could also be used with measurement models, for example, with lack of efficacy, cynicism, and exhaustion as manifest indicators of a latent factor. However, in meta-analysis the most common case is that burnout would be measured using different (numbers of) variables. Therefore, combining the available variables for each primary study and then using a single manifest indicator in subsequent CoTiMA is frequently the only viable way.

```
empcov128 <- readRDS(file=paste0(activeDirectory, "empcov128.rds"))
pairwiseN128 <- readRDS(file=paste0(activeDirectory, "pairwiseN128.rds"))
delta_t128 <- c(6)
alphas128 <- c(.87, .88, .80,      .94, .91, .88, .95, .81, .88)
empcov128 <- ctmaCorRel(empcov128, alphas128)
targetVariables128 <- c("role stress_1",
                        "exhaustion_1", "cynicism_1", "efficacy_1",
                        "exhaustion_2", "cynicism_2", "efficacy_2")
recodeVariables128 <- c("efficacy_1", "efficacy_2")
sampleSize128 <- NULL
combineVariables128 <- list("role stress_1",
                            c("exhaustion_1", "cynicism_1", "efficacy_1"),
                            c("exhaustion_2", "cynicism_2", "efficacy_2"))
combineVariablesNames128 <- c("Demands1", "Burnout1", "Burnout2")
missingVariables128 <- c(3)
results128 <- ctmaEmpCov(targetVariables=targetVariables128,
                        recodeVariables=recodeVariables128,
                        combineVariables=combineVariables128,
                        combineVariablesNames=combineVariablesNames128,
                        missingVariables=missingVariables128,
                        nlatents=2,
                        pairwiseN=pairwiseN128,
                        Tpoints=2,
                        empcov=empcov128)
empcov128 <- results128$r
pairwiseN128 <- results128$pairwiseNNew
```

Figure 12: Processing correlation matrices (`ctmaCorRel`, `ctmaEmpCov`)

unreliability, a vector of reliabilities (`alpha128`) has to be provided, and then the `ctmaCorRel` is used to replace `empcov128` by its disattenuated counterpart[7].

Second, we reduce the number of variables. All variables except the two we want to drop are assigned to `targetVariables128`. Note that a formal requirement of CoTiMA is that the variables are ordered in Time (Time 0 variables, Time 1 variables, etc.). This is also achieved by ordering the variables accordingly when creating `targetVariables128`.

Third, the two variables we want to recode are assigned to the object `recode-Variables128`. If an `empcov`i does not include variable names (no dimnames), one could use the variables' positions (i.e., `recodeVariables128 <- c(4, 7)`). Note that if numbers are used, they should correspond to the positions in the `targetVariables`i object rather than the ros/columns in the `empcov`i object (i.e., recoding is done after `targetVariables`i were selected from `empcov`i).

Although it is not explicitly necessary to assign NULL to `sampleSize128`,

---

[7]Correlations are disattenuated using the well-known formula developed by Spearman (1904). This formula is based on several assumption. One of these assumptions is that the assumption underlying Cronbach's alpha (or any other estimate of reliability), which is usually used to measure reliability, are correct. While violations of the assumptions do usually not cause visible consequences when dealing with a single cross-sectional correlation coefficient, in the case of correlation matrices of longitudinal studies it might cause problems. One problem is that disattenuated test-retest correlations could become larger than 1.0, which is automatically corrected by `ctmaCorRel` (i.e., they are set to 1.0). Another problem is that the disattenuated matrices might not positive definite and could not be analyzed then.

we do it for completeness here. This prevents that a single sample size for Study 128 is reported in some of the generated outputs, which might be misleading.

Fourth, we use a list (!) of variable names or vectors of variable names to specify the variables that should or should not be combined. This list is stored in the object `combineVariables128`. We keep the variable 'role stress_1' as it is, whereas for the first and second measurement occasion the three burnout variables are combined into a single scale, respectively. The three final variables are then labeled as specified in `combineVariablesNames128`.

Fifth, since there is no variable for demands at the second time point, we declare it as missing. This is done by stating which variable is missing in the imagined set of 'Demands1, Burnout1, Demands2, Burnout2', which is the 3rd element. Thus, `missingVariables128 <- 3`.

```
## [1] empcov128
##           [,1]      [,2] [,3]      [,4]
## [1,] 1.0000000 0.7361878   NA 0.5809288
## [2,] 0.7361878 1.0000000   NA 0.8118634
## [3,]        NA        NA   NA        NA
## [4,] 0.5809288 0.8118634   NA 1.0000000
## [1] pairwiseN128
##      [,1] [,2] [,3] [,4]
## [1,]   88   77    0   44
## [2,]   77   55    0   44
## [3,]    0    0    0    0
## [4,]   44   44    0   22
```

Figure 13: Results of applying `ctmaEmpCov` to the specifications of Study 128

The CoTiMA package comes with the function `ctmaEmpCov` which performs the desired operations (recoding, combining etc.) and yields the final correlation matrix that we want to use for our subsequent CoTiMA. Since we have a matrix of pairwise N this will be processed by `ctmaEmpCov`, too. Among others, `ctmaEmpCov` returns a new correlation matrix which is then used to replace the `empcov128` from which we started. Further, `ctmaEmpCov` returns a new matrix of pairwise N, which is then used to replace the `pairwiseN128`. Figure 13 shows the new correlation matrix and matrix of pairwise N.

```
rawData128 <- list(fileName=paste0(activeDirectory, "rawdata128.txt"),
                    studyNumbers=128, missingValues=c(-99),
                    standardize=TRUE, header=FALSE, dec=".", sep=" ")
delta_t128 <- c(NA)
```

Figure 14: Using raw data

Instead of correlation matrices, raw data can be used as well, and the arguments required to read raw data from disc have to be stored in a `rawData`i object. In R, a `list` is a list (sic!) that has elements, which have their own labels (like in a shopping list, in which you summarize the planned purchases in subitems like "vegetables", "cheese" etc.). Unlike a vector, the elements of a list could be of different types, for example, characters, numbers, symbols, ma-

trices etc. The list object created in Figure 14 has seven elements: `fileName`, `studyNumbers`, `missingValues`, `standardize`, `header`, `dec`, and `sep`.

```
  V1_T0    V2_T0    V1_T1    V2_T1   dT1
  0.835    2.328   -0.778    2.969    11
  1.555    2.634    1.977    1.807    12
  3.209    1.849    2.291    2.795    12
  0.416    2.351    0.127    1.705    13
-99.000  -99.000    0.476  -99.000    13
-99.000  -99.000    0.854  -99.000    11
-99.000  -99.000  -99.000    2.987    12
-99.000  -99.000  -99.000    2.087    12
-99.000  -99.000  -99.000    0.927    13
```

Figure 15: Raw data file structure

The raw data have to be included in an ordinary text file, and the name of the file should be stored in the list element `fileName`. Possibly missing values should be specified, and only a single value is possible (-99 is assumed by default) and stored in the list element `missingValues`. Whether or not the raw data should be standardized, which implies the analysis of correlations, or not, which implies the analysis of covariance, could be specified by setting the list element `standardize` to either `TRUE` (default and recommended) or to `FALSE`. Whether or not the raw data files include a header with variable names (as for the example data below) could be specified by setting the element `header` to either `TRUE` (default) or to `FALSE`. Finally, a decimal delimiter (default = '.') and the characters separating the values (default = ' ') could be defined using the list elements `dec` and `sep`, respectively. Note that in meta-analysis, moderators are usually study characteristics (e.g., the average age of a sample) rather than characteristics of individual study participants. Therefore, moderator values are not taken from a raw data file, but they are defined directly for a primary study that does provide raw data by assigning values to the moderator object; this is explained later.

Raw data of a primary study has to be provided as a text (ascii) file. Data has to be in wide format (i.e., one row per individual). Assuming there are t measurement occasions, the order of the variables should be V1_T0, V2_T0, ..., V1_Tt, V2_Tt, dT1, dT2, ... dT(t-1), where dTt are the variables representing the time intervals (deltas) between measurements (see Figure 15; here with header). Note that if t measurement occasions exist, there are t-1 time intervals. Compared to correlation matrices as input, raw data allow the time intervals to vary between the individuals within a study (average time intervals are automatically reported in CoTiMA fit objects). However, for studies that supply raw data, it is mandatory to specify the `delta_ti` object! It has to have as many NA as the largest number of possible time intervals in the respective study is, for example, in the case of not more than three intervals, `delta_ti <- c(NA, NA)`. In the example in Figure 15 there are only two time points and, thus, one interval `dT1`. Thus, `delta_ti` is indeed the only mandatory object

```
ageM2 <- 39.3
ageSD2 <- 8.7
malePercent2 <- .60
occupation2 <- c("Bank employees")
country2 <- c("Netherlands")
demands2 <- c("Workload")
burnout2 <- c("Emotional Exhaustion")
targetVariables2 <- c("Demands1", "Burnout1", "Demands1", "Burnout2")
source2 <- c("Houkes, I,", "Janssen, P, P, M,", "de Jonge, J", "& Bakker, A, B",
             "Study1", "2003")
moderator2 <- c(1, 0.72)

ageM3 <- 47.4
ageSD3 <- 5.8
malePercent3 <- .70
occupation3 <- c("Teachers for adults")
country3 <- c("Netherlands")
demands3 <- c("Workload")
burnout3 <- c("Emotional exhaustion")
targetVariables3 <- c("Demands1", "Burnout1", "Demands1", "Burnout2")
source3 <- c("Houkes, I,", "Janssen, P, P, M,", "de Jonge, J", "& Bakker, A, B",
             "Study2", "2003")
moderator3 <- c(1, 0.72)

ageM313 <- 30
ageSD313 <- 6
malePercent313 <- 0.30
occupation313 <- c("Wmployment agency employees")
country313 <- c("Netherlands")
demands313 <- c("Work pressure")
burnout313 <- c("Exhaustion")
targetVariables313 <- c("Demands1", "Burnout1", "Demands1", "Burnout2",
                        "Demands3", "Burnout3")
source313 <- c("Demerouti", "Bakker", "& Bulters", "2004")
moderator313 <- c(2, 0.72)

ageM128 <- 41
ageSD128 <- 11.4
malePercent128 <- 0.203
occupation128 <- c("Managerial employees in NHS trusts")
country128 <- c("UK")
demands128 <- c("Role Stress")
burnout128 <- c("Exhaustion", "Cynicism")
source128 <- c("Childs, J. H.", "& Stoeber, J.", "Study1", "2012")
moderator128 <- c(2, 0.66)
```

Figure 16: Additional information for study data entered before

because `rawData`*i* could substitute `empcov`*i* and `pairwiseN`*i* could substitute `sampleSize`*i*.

So far, we introduced the objects `delta_t`*i*, `sampleSize`*i*, `empcov`*i*, `targetVariables`*i*, `alphas`*i*, `pairwiseN`*i*, and `rawData`*i*). Further pre-defined object names are:

- `moderator`*i*. A vector of numerical values either representing categorical or continuous variables, e.g., `moderator6 <- c(1, 2, 2, 0.76, 2.56,`

2001)

- `startValues`i. A vector of start values, which was used in previous Co-TiMA versions. Currently the use of start values is disabled, but this might change in the future.

- `empMeans`i. Mean values of variables (default = 0). It is not recommended to change the default, but it is possible, e.g., `empMeans7 <- c(1, -2.5, 1.1, -2.4)`

- `empVars`i. Variances of variables; (default = 1). It is not recommended to change the default, but it is possible, e.g., `empVars6 <- c(1, 2, 1.1, 1.9)`

- `studyNumber`i. A special number used for labeling in the outputs of subsequently fitted CoTiMA models, e.g., `studyNumber6 <- 66`

- `source`i. Useful to label the table displaying the estimated parameters for each primary study, rather than using the numbers used for the primary study objects (e.g., 128 from `empcov128`), e.g., `source6 <- c("De Jonge", "Dormann", "Janssen", "Dollard", "Landeweerd", "& Nijhuis", "2001")`

- `ageM`i. A value indicating the mean age of participants in a primary study, e.g., `ageM6 <- 31.78`

- `malePercent`i. A value indicating the percentage of male participants in a primary study, e.g., `malePercent6 <- 0.11`

- `occupation`i. A vector of character strings representing the occupations of participants in a primary study. Of course, this has not to be taken literally. For example it could be also used to represent the program in which student participants are enrolled and similar classifications, e.g., `occupation6 <- c("Health care workers")`

- `country`i. A single character string representing the country in which a primary study was conducted, e.g., `country6 <- c("Netherlands")`

In addition to these pre-defined object names, user-defined object names could be added (e.g., `demands`i and `burnout`i, to add information about the type of measurement scale used in primary studies). The difference between pre-defined and user-defined objects is twofold. First, pre-defined objects are included in the Excel workbook that summarizes primary study information (see. Figure 20). Second, user-defined objects have to be declared in `ctmaPrep` using the argument `addElements` (see Figure 18).

To proceed further with the example, in a first step documented in Figure 16 we add information to those four primary studies data already entered before. In a second step, we add two further primary study information as shown in Figure 17.

```
empcov18 <- matrix(c(1.00, 0.44, 0.62, 0.34,
                     0.44, 1.00, 0.41, 0.62,
                     0.62, 0.41, 1.00, 0.55,
                     0.34, 0.62, 0.55, 1.00), 4, 4)
variableNames18 <- c("Demands_1",  "Burnout_1", "Demands_2",  "Burnout_2")
dimnames(empcov18) <- list(variableNames18, variableNames18)
delta_t18 <- 12
sampleSize18 <- 174
ageM18 <- 41.33
ageSD18 <- 9.70
malePercent18 <-0.03
occupation18 <- c("Service employees")
country18 <- c("Germany")
demands18 <- c("Workload")
burnout18 <- c("Emotional exhaustion", "Depersonalization")
source18 <- c("Diestel", "& Schmidt", "Study 1", "2012")
moderator18 <- c(1, 0.7)

empcov32 <- matrix(c(1.00, 0.45, 0.70, 0.40,
                     0.45, 1.00, 0.36, 0.66,
                     0.70, 0.36, 1.00, 0.43,
                     0.40, 0.66, 0.43, 1.00), 4, 4)
variableNames32 <- c("Demands_1",  "Burnout_1", "Demands_2",  "Burnout_2")
dimnames(empcov32) <- list(variableNames32, variableNames32)
delta_t32 <- 8
sampleSize32 <- 433
ageM32 <- 41.5
ageSD32 <- 10.2
malePercent32 <- 0.199
occupation32 <- c("Teachers")
country32 <- c("Canada")
demands32 <- c("classroom overload")
burnout32 <- c("Emotional exhaustion", "Depersonalization")
source32 <- c("Fernet", "Guay", "Senecal", "& Austin", "2012")
moderator32 <- c(1, NA)
```

Figure 17: Information for two further primary studies

The six studies are now compiled into a list as shown in Figure 18. We add the two user-defined object names **demands**i and **burnout**i. We also provide a vector with the labels of the two moderators, and we provide a list of vectors to label the moderator values.

To get a convenient overview of the information stored in this list, one could use the openxlsx R package (see Figure 19). An example of what is displayed when opening the excel workbook with its several sheets with openXL is shown in Figure 20. The workbook could also be saved to disk using the **saveWorkbook** function of openxlsx.

```
CoTiMAstudyList_6 <- ctmaPrep(selectedStudies=c(2, 3, 313, 128, 18, 32),
addElements = c("demands", "burnout"),
moderatorLabels=c("Burnout Measure", "Control at Work"),
moderatorValues=list(c("1 = Emotional Exhaustion", "2 = Exhaustion"),
                     "continuous"))
```

Figure 18: Compiling a list of primary studies with extended information (`ctmaPrep`)

```
openXL(CoTiMAstudyList_6$excelSheets)
saveWorkbook(CoTiMAstudyList_6$excelSheets, overwrite=TRUE,
             file=paste0(activeDirectory, "myExcelSheet.xlsx") )
```

Figure 19: Open an Excel sheet with summary information included in a compiled list of primary studies (requires package openxlsx)

Figure 20: Excel sheet with summary information included in a compiled list of primary studies

6 Initial Fitting (`ctmaInit`)

Now the first two steps (**E**xtract & **P**repare) in the recommended EPIC-BiG-Power workflow are done and we can move forward to the 'Init' step, for which the previously compiled `CoTiMAstudyList_6` is required. Initial fitting is done with the code in Figure 21 (analogous to Figure 4), and the result is then displayed on the console (see Figure 22).

```
CoTiMAInitFit_6 <- ctmaInit(primaryStudies=CoTiMAstudyList_6,
                            n.latent = 2,
                            activeDirectory = activeDirectory,
                            coresToUse = 1)
```

Figure 21: Fitting a ctsem model for each primary study (ctmaInit)

```
##                                                    V1toV1     SE        V2toV1
## Study No 2   "Houkes et al., Study1, 2003"         "-0.0486" "0.012"   "0.0071"
## Study No 3   "Houkes et al., Study2, 2003"         "-0.0323" "0.0094"  "0.0054"
## Study No 313 "Demerouti et al., 2004"              "-0.4131" "0.0453"  "0.1468"
## Study No 128 "Childs, & Stoeber, Study1, 2012"     "-10.9606" "0.6433" "0.0625"
## Study No 18  "Diestel, & Schmidt, Study 1, 2012"   "-0.0541" "0.0102"  "0.026"
## Study No 32  "Fernet et al., 2012"                 "-0.051"  "0.0071"  "0.0111"
##               SE        V1toV2     SE        V2toV2     SE
## Study No 2   "0.009"   "-0.0189"  "0.0111"  "-0.027"   "0.0087"
## Study No 3   "0.0106"  "0.0111"   "0.0097"  "-0.0367"  "0.0115"
## Study No 313 "0.0373"  "0.0861"   "0.0346"  "-0.2777"  "0.0315"
## Study No 128 "2.4258"  "-0.4631"  "0.0249"  "-0.6992"  "2.1141"
## Study No 18  "0.0105"  "0.0114"   "0.0099"  "-0.0481"  "0.0112"
## Study No 32  "0.0075"  "0.0258"   "0.008"   "-0.0663"  "0.0096"
```

Figure 22: Some results for the primary studies (ctmaInit)

For Study 128, which we used to demonstrate how to deal with missing variables, some unusual estimates emerged that are not unexpected. In Study 128, the variable V1_T1 was missing (demands T1, i.e., 'role stress_2'). Obviously, the absolute values of *V1toV1* and its *SE* and the *SE* of *V2toV1* are much larger than their counterparts of the other studies. We will use this case to review some of the general principles of continuous time structural equation modeling (CTSEM).

First, in CTSEM any pair of subsequent measurement occasions is regarded as equivalent except the length of the time interval, which may vary. Therefore, continuous time coefficients do not describe, for example, the relations between demands at Time 0 and burnout at Time 1. Rather, *earlier* demands affect *later* burnout. Thus, in CoTiMA, the effect *V1toV1* means the auto effect of earlier V1 to later V1. Similarly, the effect V1toV2 means the cross effect of earlier V1 to later V2. In continuous time, the terms *auto* effect and *cross* effects are used, whereas in discrete time, the terms *auto-regressive* effect and *cross-lagged* effects are used. In a similar vein, the terms diffusion (variance) in continuous time substitutes the term error (variance) in discrete time, and the term continuous time intercepts substitutes the term intercept (for more details

see Voelkle et al., 2012; Driver et al., 2017).

Between these pairs of continuous time and discrete time coefficients, well-defined mathematical relations exist. The only reason why continuous time coefficients are used is that the math is known to describe how coefficients change across time. To translate auto and cross effects into auto-regressive and cross-lagged effects, put the former into a matrix, multiply the matrix by length of time interval, and then apply the matrix (!) exponential function. This matrix contains the auto effect in the diagonal and the cross effects off the diagonal.

Figure 23 shows how the continuous time drift effects obtained for Study 313 (see Figure 22) relate to 1-month auto-regressive and cross-lagged effects in discrete time. Demands are slightly less stable ($V1toV1$) than Burnout ($V2toV2$). The negative auto effects in continuous time thus translate into positive auto-regressive effects in discrete time. Thus, in continuous time, the more negative an auto effect is, the less stable a variable is. Further, the effect of earlier demands on later burnout is smaller ($V1toV2$) than the effect of earlier burnout on later stressors ($V2toV1$). Note that multiplying the matrix with, for example, 2 (i.e., `expm(A313 * 2)`) yields the effects across a 2-month lag. This is the way how discrete time effect sizes are computed and plotted (see Figure 10).

```
## [1] library(expm)
## [1] A313 <- matrix(c(-0.4131, 0.1468, 0.0861, -0.2777), 2, 2, byrow=TRUE)
## [1] A313
##         [,1]     [,2]
## [1,] -0.4131  0.1468
## [2,]  0.0861 -0.2777
## [1] expm(A313 * 1)
##            [,1]       [,2]
## [1,] 0.6659772 0.1042237
## [2,] 0.0611285 0.7621072
```

Figure 23: Relation between continuous time drift coefficients of Study 313 and its discrete time effects

The result of applying the same transformation to the suspicious drift effects of Study 128 is shown in Figure 24. The critical auto effect $V1toV1$ corresponds to an almost zero auto-regressive effect across 1 month. A person's level of demands at work does virtually not predict the person's demands one month later, which one would usually regard as not very plausible. In fact, this out-of-range estimate is a consequence that in Study 128 later demands was a missing variable. Thus, we cannot expect meaningful results from fitting a ctsem model to Study 128. However, we could use the empirical information Study 128 provides in a CoTiMA, in which we assume drift effects to be invariant across primary studies[8].

---

[8]Assuming invariant drift effect means that we aggregate $V1toV1$, $V1toV2$, $V2toV1$, and $V2toV2$ across primary studies. Technically speaking, this is done by estimating the set of four values that lead to the smallest difference between the model-implied and empirical covariance matrices. Study 128 could not directly provide information to estimate $V1toV1$ and $V2toV1$, but it does provide information useful for estimating $V1toV2$ and $V2toV2$ (and the variances

```
## [1] library(expm)
## [1] A128 <- matrix(c(-10.9606, 0.0625, -0.4631, -0.6992), 2, 2, byrow=TRUE)
## [1] A128
##          [,1]    [,2]
## [1,] -10.9606  0.0625
## [2,]  -0.4631 -0.6992
## [1] expm(A128 * 1)
##                [,1]         [,2]
## [1,] -0.0001189109 0.003020042
## [2,] -0.0223773037 0.495718843
```

Figure 24: Relation between continuous time drift coefficients of Study 128 and its discrete time effects

Again, model results could also be opened as excel workbook with `open-XL(CoTiMAInitFit_6$excelSheets)`. For example, effects, their standard errors and lower limit (LL) and upper limit (UL) credible intervals are shown in Figure 24. From the workbook, coefficients could be easily copied into a word processing app to build proper results tables.

	C	D	E	F	G	H	I	J	K	L	M	N
	V1toV1	SE	V2toV1	SE	V1toV2	SE	V2toV2	SE	V1toV1LL	V1toV1UL	V2toV1LL	V2toV1UL
A, B, Study1, 2003	-0.0514	0.011	0.0089	0.0099	-0.0192	0.0091	-0.0271	0.0091	-0.0758	-0.0332	-0.0108	0.0283
A, B, Study2, 2003	-0.0328	0.011	0.0052	0.0104	0.0112	0.0107	-0.0377	0.0113	-0.0589	-0.0165	-0.0152	0.0254
	-0.4162	0.043	0.1428	0.0395	0.0921	0.0347	-0.2829	0.032	-0.5064	-0.3372	0.0658	0.2184
	-19.3478	52.2561	1.5927	5.6399	-0.0063	0.0185	-0.0001	0.0023	-110.3568	0	-9.4899	12.588
	-0.0536	0.0102	0.0257	0.0099	0.0125	0.0101	-0.0477	0.0102	-0.0761	-0.0362	0.0065	0.0449
	-0.0509	0.0075	0.0113	0.0076	0.0255	0.0081	-0.0652	0.0085	-0.0672	-0.0376	-0.0038	0.0262

| confidenceIntervals | randomEffects | stats | + |

Figure 25: Excel sheet with summary of model results

Doing the initial fitting of ctsem models to all primary studies allows setting several options (arguments) such as, for example, constraining some drift effects to be 0.0, or using different estimators such as Bayesian instead of maximum likelihood estimation. The arguments to select estimators are introduced as follows, and the entire list of possible arguments of the different CoTiMA functions are listed in the Appendix. Note that the `optimize` argument should be used and not be confused with `optimise`, which is used by ctsem.

One particular option is to use Bayesian estimation. Bayesian estimates are drawn from posterior probability distributions, for which different samplers could be used. The most robust sampler is the Hamiltonian Monte Carlo (HMC) sampler (Stan Developer Team, 2020a). However, it is the slowest one. The Stan Math library (Carpenter, Hoffman, Brubaker, Lee, Li & Betancourt, 2015), which is used by ctsem and CoTiMA for estimation, offers a No U-Turn Sampler (NUTS), which is a variant of the HMC sampler and usually works well and is faster. However, both samplers are much (!) slower than maximum likelihood estimation, which is the default estimator, or maximum a posteriori estimation. In fact, most desktop computers in 2021 probably would need a couple of weeks

of V1 and V2, their covariance, and the residual variance of V2).

for a full CoTiMA with Bayesian estimation if 20 or more primary studies are analyzed. Table 1 gives an overview of how the different estimators can be requested by setting the optimize and the nopriors argument of all CoTiMA fitting functions (`ctmaInit`, `ctmaFit`, `ctmaEqual`, & `ctmaPower`).

Table 1: Estimators available for CoTiMA

		optimize	
		FALSE	TRUE
	FALSE	Bayesian estimation via Stan's NUTS sampler	Maximum a posteriori estimation
nopriors			
	TRUE	Bayesian estimation via HMC sampling (`nopriors` will be changed to `FALSE`)	maximum likelihood estimation (default)

Note: HMC = Hamiltonian Monte Carlos Sampler; NUTS = No U-Turn Sampler

Weakly informative priors for Bayesian estimation with the NUTS sampler and for maximum a posteriori estimation are provided by ctsem. They work well under most circumstances, however, sometimes they might not work well because the priors provided by ctsem have been optimized for time measured in years. For example, one could use the argument `scaleTime = 1/365.25` if time was measured in days and previous fitting attempts did not yield meaningful results.

Figure 26 shows how Bayesian estimates using the NUTS sampler could be obtained. Since estimation requires long time (expect several hours), it is recommended to save the model fits for each primary study using the `saveSingle-StudyModelFit` argument. If further studies are added later, re-estimating these models could be avoided by the corresponding `readSingleStudyModelFit` argument. In the example in Figure 26, we used `chains = 2` and `coresToUse = 2`. Three chains and three cores are recommended before publishing results. Note that on Windows machines using more than one core may not work; use `coresToUse = 1` then (which doubles the time needed to fit the models). Since Bayesian estimation takes a long time, we want to take care that we get precise results in our first fitting attempt; we set `finishsamples = 10000` for this purpose. This means parameter estimates and the credible intervals will be sampled 10000 times from the estimated parameter distribution, rather than only 1000 sample, which is the default for `finishsamples`.

Part of the results obtained from the code in Figure 26 printed to the console with `summary(CoTiMAInitFit_6_NUTS)` is shown in Figure 26[9]. A comparison with the maximum likelihood effects and their standard errors in Figure 22

---

[9]In addition, several warning messages are issued. They are all related to Study 128, for which we introduced missing data. This does not happen if doing the analysis again without Study 128.

```
CoTiMAInitFit_6_NUTS <- ctmaInit(primaryStudies=CoTiMAstudyList_6,
                                 n.latent = 2,
                                 activeDirectory=activeDirectory,
                                 saveSingleStudyModelFit =
                                   c("InitFit_6_NUTS", 2, 3, 313, 128, 18, 32),
                                 optimize=FALSE,
                                 nopriors=FALSE,
                                 chains = 2,
                                 coresToUse = 2,
                                 finishsamples = 10000)
saveRDS(CoTiMAInitFit_6_NUTS, file=paste0(activeDirectory,
                                 "CoTiMAInitFit_6_NUTS.rds"))
summary(CoTiMAInitFit_6_NUTS)
```

Figure 26: Using Bayesian estimation via Stan's NUTS sampler (ctmaInit)

reveals no substantial differences except for Study 128, for which results are not trustworthy anyway. We should note, further, that Bayesian estimation is sensitive to priors, and default priors are only appropriate if the time scale is appropriately chosen, too. This could require using an appropriately chosen `scaleTime` argument (see Footnote 5) and setting `customPar = FALSE` (see Appendix).

```
##                                                  V1toV1    SE        V2toV1
## Study No 2    "Houkes et al., Study1, 2003"      "-0.0537" "0.0117" "0.0086"
## Study No 3    "Houkes et al., Study2, 2003"      "-0.0346" "0.0113" "0.0061"
## Study No 313 "Demerouti et al., 2004"            "-0.4204" "0.0447" "0.1459"
## Study No 128 "Childs, & Stoeber, Study1, 2012"   "-4.257"  "2.4053" "-0.0052"
## Study No 18   "Diestel, & Schmidt, Study 1, 2012" "-0.0562" "0.0108" "0.0276"
## Study No 32   "Fernet et al., 2012"              "-0.0517" "0.0079" "0.0119"
##              SE       V1toV2    SE        V2toV2    SE
## Study No 2    "0.0103" "-0.0193" "0.0094" "-0.0284" "0.0086"
## Study No 3    "0.0111" "0.0119"  "0.0109" "-0.0396" "0.0116"
## Study No 313 "0.0404" "0.0944"  "0.034"  "-0.284"  "0.0308"
## Study No 128 "0.929"  "0.03"    "0.3631" "-0.0744" "0.0754"
## Study No 18   "0.0104" "0.014"   "0.0111" "-0.0506" "0.0111"
## Study No 32   "0.0079" "0.026"   "0.0084" "-0.0661" "0.0088"
```

Figure 27: Estimates for the primary studies using Bayesian estimation (`ctmaInit`)

7   CoTiMA (`ctmaFit`)

Now the first three steps (**E**xtract, **P**repare, & **I**nitFit) in the recommended EPIC-BiG-Power workflow are done, and we can move forward to do CoTiMAs, for which the now available `CoTiMAInitFit_6` (or `CoTiMAInitFit_6_NUTS`) object is required. In the first subsection, we demonstrate how a *full CoTiMA* with all drift effects could be fitted. In the second subsection, we show how a *partial CoTiMA* could be fitted, and we use this subsection to introduce the possibilities to analyze (a) subsets of studies, specific (b) invariance constraints,

and (c) hierarchical CoTiMA using clusters of studies. In the third subsection, we show how to statistically *test the equality* of drift effects, that is, a CoTiMA with equality constraints. In the fourth subsection, we show how a *moderated CoTiMA* can be performed.

7.1 Full CoTiMA (`ctmaFit`)

```
CoTiMAFullFit_6 <- ctmaFit(ctmaInitFit = CoTiMAInitFit_6_NUTS,
                           coresToUse = coresToUse)
saveRDS(CoTiMAFullFit_6, file=paste0(activeDirectory, "CoTiMAFullFit_6.rds"))
summary(CoTiMAFullFit_6)
```

Figure 28: Full CoTiMA with six studies (ctmaFit)

Fitting a full CoTiMA is done with the code in Figure 28 and with `summary(-CoTiMAFullFit_6)` the results are displayed[10]. The term 'full CoTiMA' is used to refer to a model in which all possible auto effects and all possible cross effects are simultaneously aggregated. Later, we show how some effects could be excluded from the model (i.e., fixed to 0.0), and how some effects could be exempted from being invariant across primary studies. It is noteworthy that the estimator used for initial fitting, which was NUTS, does not affect which estimator is used in a CoTiMA; it is maximum likelihood in the present example, which is the default estimator. Other estimators would have to be specified as shown in Table 1.

The results in Figure 29 show the names of all parameters of the full CoTiMA model, their respective row and column numbers in the matrices in which they are used, their estimated mean population values, their standard *errors* (labelled *sd*), their 2.5% lower credible interval, mean, and 97.5% upper credible interval, and the T-values.

The four rows starting with `DRIFT` show the estimates for the continuous time drift coefficients, and their discrete time counterparts, that is, the autoregressive and cross-lagged effects, across one month are again shown closer to the bottom (`dtDRIFT`). As explained earlier, only the four rows containing the drift coefficients are usually important for reporting CoTiMA results. Nevertheless, we briefly explain what the other parameters stand for. For a more detailed description see Driver, Oud, and Voelkle (2017) and exact mathematical definitions can be found in Driver and Voelkle (2018).

`TOMEANS` at the top of Figure 29 represent the initial (T0) means of the latent variables. Closer to the bottom in Figure 29, `T0cov` shows correlation of the latent factors at T0, which is identical to their covariance because we deal with standardized variables here.

`LAMBDA` is a matrix with the factor loadings of the manifest variables on the latent factors. In the present example, this is a diagonal matrix in which the

---

[10]Fitting will issue a warning that an 'approximate' Hessian was used and standard errors are not trustworthy. This is caused by the missing variables in Study 128. Still, the credible intervals are not based on standard errors.

```
##                                 row col    Mean     sd     2.5%     50%    97.5%   Tvalues
## TOMEANS_1_1                        1   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## TOMEANS_2_1                        2   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## LAMBDA_1_1                         1   1  1.0000 0.0000   1.0000  1.0000   1.0000       Inf
## LAMBDA_1_2                         1   2  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## LAMBDA_2_1                         2   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## LAMBDA_2_2                         2   2  1.0000 0.0000   1.0000  1.0000   1.0000       Inf
## DRIFT V1toV1 (invariant)           1   1 -0.0563 0.0055  -0.0679 -0.0556  -0.0468  -10.1584
## DRIFT V2toV1 (invariant)           1   2  0.0165 0.0042   0.0072  0.0168   0.0255   3.8885
## DRIFT V1toV2 (invariant)           2   1  0.0129 0.0044   0.0052  0.0128   0.0226   2.9270
## DRIFT V2toV2 (invariant)           2   2 -0.0555 0.0051  -0.0674 -0.0551  -0.0457  -10.8589
## MANIFESTMEANS_1_1                  1   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## MANIFESTMEANS_2_1                  2   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## CINT_1_1                           1   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## CINT_2_1                           2   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## asymCINT_1_1                       1   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## asymCINT_2_1                       2   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## asymDIFFUSIONcov_1_1               1   1  0.9566 0.0949   0.7809  0.9498   1.1208   10.0789
## asymDIFFUSIONcov_1_2               1   2  0.3851 0.0714   0.2705  0.3802   0.5210   5.3909
## asymDIFFUSIONcov_2_1               2   1  0.3851 0.0714   0.2705  0.3802   0.5210   5.3909
## asymDIFFUSIONcov_2_2               2   2  1.0314 0.0951   0.8656  1.0263   1.1997   10.8420
## DIFFUSIONcov_1_1                   1   1  0.0941 0.0063   0.0824  0.0937   0.1066   14.8899
## DIFFUSIONcov_1_2                   1   2  0.0133 0.0048   0.0045  0.0137   0.0213   2.7736
## DIFFUSIONcov_2_1                   2   1  0.0133 0.0048   0.0045  0.0137   0.0213   2.7736
## DIFFUSIONcov_2_2                   2   2  0.1037 0.0076   0.0907  0.1027   0.1218   13.6653
## MANIFESTcov_1_1                    1   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## MANIFESTcov_1_2                    1   2  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## MANIFESTcov_2_1                    2   1  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## MANIFESTcov_2_2                    2   2  0.0000 0.0000   0.0000  0.0000   0.0000       NaN
## T0cov_1_1                          1   1  1.0033 0.0488   0.9189  1.0046   1.0822   20.5573
## T0cov_1_2                          1   2  0.4511 0.0440   0.3698  0.4527   0.5453   10.2453
## T0cov_2_1                          2   1  0.4511 0.0440   0.3698  0.4527   0.5453   10.2453
## T0cov_2_2                          2   2  1.0030 0.0517   0.9071  1.0062   1.0985   19.4119
## dtDRIFT_1_1                        1   1  0.9454 0.0052   0.9344  0.9461   0.9543  180.7894
## dtDRIFT_1_2                        1   2  0.0156 0.0040   0.0068  0.0159   0.0241   3.8932
## dtDRIFT_2_1                        2   1  0.0122 0.0042   0.0049  0.0120   0.0214   2.9330
## dtDRIFT_2_2                        2   2  0.9461 0.0048   0.9350  0.9464   0.9554  196.2402
```

Figure 29: Results (Part 1) of a Full CoTiMA with six studies (`ctmaFit`)

diagonal was fixed to 1.0. By this, each manifest variable loads on a single latent factor. Conversely, each latent factor is identified by a single manifest variable.

`MANIFESTMEANS` is a matrix (with a single column only) containing the means of the intercepts of the manifest variables. Again, all values were fixed to 0.0 because we deal with standardized variables here.

`CINT` are the continuous time intercepts, which in case of standardized variables are usually zero. `asymCint` are the asymptotic continuous time intercepts. They reflect the intercept values to which the process converges after infinite time. These values should also be 0.0 in the case of CoTiMA, where we use standardized variables (correlations).

Similarly, `DIFFUSIONcov` are the continuous time error variances (usually referred to as innovations in the literature), and `asymDIFFUSIONcov` reflect asymptotic diffusion (error) variances and covariances. One might speculate that the asymptotic diffusion (error) variances should be 1.0 since one cannot explain any variance across infinite time. However these estimates are based on internal transformations. Asymptotic matrices are internally useful to reduce the time to fit the model and have no inherent meaning.

`MANIFESTcov` is a matrix of variances and covariances among the manifest

variables at each measurement occasion. All values were fixed to 0.0 because we had only a single manifest indicator per latent factor.

```
## $randomEffects
## NULL
##
## $minus2ll
## [1] 14007.6
##
## $n.parameters
## [1] 40
##
## $opt.lag
##      [,1] [,2]
## [1,]   NA   19
## [2,]   19   NA
##
## $max.effects
##         [,1]   [,2]
## [1,]      NA 0.1062
## [2,] 0.0856     NA
##
## $clus.effects
## NULL
##
## $message
## [1] "Mean time interval was 7.57142857142857. It is recommended to fit"
## [2] "the model again using the arguments scaleTime=1/12 and "
## [3] "customPar=FALSE.  If the model fit (-2ll) is better (lower),"
## [4] "continue using, e.g., scaleTime=1/12 in all subsequent models."
```

Figure 30: Results (Part 2) of a Full CoTiMA with six studies (`ctmaFit`)

Part 2 of the results generated by the code in Figure 28 is shown in Figure 30. A random effects model was not requested because in our experience it hardly yields reasonable results[11]. Random effects are not reported because they were not requested. The -2 loglikelihood values and number of estimated parameters are reported next. Then the optimal time interval according to Dormann and Griffin (2015) and the sizes of effects across the optimal interval are reported. Finally, CoTiMA allows to account for *hierarchically structured* (also called *nested* or *clustered*) primary studies. We will discuss this in the next example. Since no clusters were specified, cluster effects (`clus.effects`) do not exist. Finally, a warning message is issued that the average time intervals might have been too long to ensure proper fitting, and it is recommended to fit the model again using the `scaleTime=1/12` (and `customPar=FALSE`) argument.

---

[11]A ctsem model of a single primary study could be used to estimate random effects if the study comprises three or more waves. We have not yet investigated how large the portion of primary studies with three or more waves has to be to yield reasonable CoTiMA estimates.

7.2 Partial CoTiMA (`ctmaFit`)

Figure 31 demonstrates some further possibilities for conducting a CoTiMA; additional capabilities are explained in Appendix A. The CoTiMA model specified in Figure 31 fixes the effect of *V2toV1* to 0.0 (which we do not generally recommend - let the evidence decide rather theoretical expectations), by labeling the according drift 0 or "0". Further, only the effect *V1toV2* is invariant across primary studies as specified in the `invariantDrift` argument (which could be reasonable – and which could be decided based upon a statistical test). Finally, the first three primary studies were from The Netherlands, whereas studies 3, 4 and 5 (study numbers 128, 18 & 32), were from UK, Germany, and Canada, respectively. This hierarchical structure is specified in the `cluster` argument. In the `cluster` argument, the studies that belong to one cluster receive the same number. Finally, `coresToUse` is specified here in negative notation, and -1 implies that all cores except 1 are used. As noted earlier, on Windows machines one should use 1 (not more than 1).

```
CoTiMAPart134Inv3Fit_6 <- ctmaFit(ctmaInitFit = CoTiMAInitFit_6_NUTS,
                                   drift=c("V1toV1", 0,
                                           "V1toV2", "V2toV2"),
                                   invariantDrift=c("V1toV2"),
                                   cluster=c(1,1,1,2,3,4),
                                   coresToUse = -1)
saveRDS(CoTiMAPart134Inv3Fit_6, file=paste0(activeDirectory,
                                            "CoTiMAPart134Inv3Fit_6.rds"))
summary(CoTiMAPart134Inv3Fit_6)
```

Figure 31: A partial CoTiMA with a subset of primary studies, with one cross effect fixed to 0.0, with only a single effect invariant across primary studies, and with countries used as cluster variable (`ctmaFit`)

Parts of the summary are shown in Figure 32. The section `$estimates` shows the aggregated effect across all studies with country specific effects partialled out. Here, *V1toV2* is -.0093.

The section `$clus.effects` shows how the studies from The Netherlands (cluster 1) differ from the average effects, of which the drift effects are again of major importance. Note that cluster effects only exist for clusters comprising more than one single study; for all other studies it would be impossible to disentangle study-specific and cluster-specific effects. Each primary study (except the last one) is internally represented as a dummy variable, which affects the T0 variances and covariances, the diffusion parameters, and – not the drift parameters because they should be invariant (i.e., aggregated) across studies. Cluster dummies affect all of them. If a cluster would contain a single study only, its cluster dummy as well as its single study dummy would be perfectly correlated in affecting T0 variances and covariances and diffusion parameters. Thus, clusters have to contain more than a single study.

Additionally, the section `$clus.effects` show that the auto effect of demands (*V1toV1*) and the cross effect of demands on burnout (*V1toV2*) are significantly smaller in The Netherlands. Although it is not too important at

```
## $estimates
##                          Mean     sd    2.5%      50%   97.5%  Tvalues
## DRIFT V1toV1            -0.0932 0.0180 -0.1326 -0.0632 -0.0632  -5.1664
## DRIFT V2toV1             0.0000 0.0000  0.0000  0.0000  0.0000     NaN
## DRIFT V1toV2 (invariant) -0.0093 0.0092 -0.0272 -0.0094  0.0085  -1.0146
## DRIFT V2toV2            -0.0592 0.0056 -0.0708 -0.0589 -0.0491 -10.6445
##
## $clus.effects
## $clus.effects$effects
##                              mean      sd    2.5%     50%   97.5% Tvalues
## Cluster_1_on__V1toV1        -0.0842  0.0347 -0.1649 -0.0798 -0.0303 -2.4266
## Cluster_1_on__V1toV2        -0.0359  0.0106 -0.0566 -0.0360 -0.0147 -3.3811
## Cluster_1_on__V2toV2         0.0027  0.0058 -0.0089  0.0029  0.0139  0.4696
## Cluster_1_on__diff_eta1      0.2545  0.2938 -0.1061  0.1851  1.0055  0.8663
## Cluster_1_on__diff_eta2_eta1 0.1950  0.8070 -1.3956  0.1895  1.7545  0.2416
## Cluster_1_on__diff_eta2      0.1686  0.3349 -0.1442  0.0580  1.0853  0.5035
## Cluster_1_on__T0var_eta1     4.9612  9.1557 -0.4371 -0.0035 32.1056  0.5419
## Cluster_1_on__T0var_eta2_eta1 -0.0335 2.6489 -5.2217 -0.0616  5.2375 -0.0126
## Cluster_1_on__T0var_eta2     8.7315 14.5774 -0.4350  0.0456 49.3086  0.5990
##
## $clus.effects$weights
##        non Members Cluster Member
## 1_on__     -0.8591          1.1631
##
## $clus.effects$sizes
##   non Members Cluster Member
## N        773            571
##
## $clus.effects$cluster.specific.effect
##               DRIFT V1toV1 DRIFT V2toV1 DRIFT V1toV2 (invariant) DRIFT V2toV2
## Cluster No. 1      -0.1911           NA                  -0.0511      -0.0561
##
## $clus.effects$note
## [1] "The weights represent standardized cluster dummies. "
## [2] "They are used to multiply a cluster's TI effect and"
## [3] "this product is then added to the average effect shown in"
## [4] "$estimates, which overall yields the effects within a"
## [5] "cluster as shown in $cluster.specific.effect."
##
##
## $mod.effects
## NULL
```

Figure 32: Results of the partial CoTiMA specified in Figure 31 (`ctmaFit`)

this stage to know the country-specific effects (the main issue is that they are partialled out), they are shown in the section `cluster$specific$effect` in Figure 32. For instance, consider *V1to V2*. Since the cluster effect (-.0359) was significant because the credible interval excluded zero (LL $= -$ .0566; UL $= -$ .0147), one would conclude that there is no general effect of demands on burnout ($-$ .0093, LL $= -$ .0272, UL $=$ .0085), but in The Netherlands the continuous time drift effect is significantly smaller and $-$ .0511.

29

7.3 CoTiMA with equality constraints (`ctmaFit`, `ctmaEqual`, `ctmaCompFit`)

To statistically test if two or more effects are equal is a bit complex and requires three steps: (1) ensure correct coding (polarity), (2) fit a partially invariant CoTiMA using `ctmaFit`, and (3) test equality using `ctmaEqual`. First, (1) one has to take care that the effects to be compared have equal signs. For example, consider a model with three latent variables such as demands, resources, and burnout. Work-related resources, such as supervisor support, can be supposed to reduce burnout whereas demands increase burnout. To compare the effect sizes, one would need to go back to square one and re-start the EPIC part of the workflow. When preparing the correlations with `ctmaEmpCov`, one would need to use the `recode` argument to recode supervisor support so that it becomes lack of supervisor support. Then, one has to use `ctmaInit` again for initial fitting.

In the second step (2), one could start testing the equality of the effect sizes of supervisor support and of demands on burnout. This requires two CoTiMAs to be performed. The first CoTiMA has to specify those two or more effects as invariant across studies that should be tested for equality in the subsequent step. This is done with `ctmaFit`. We call this the *invariance model*.

```
CoTiMAFullInv23Fit_6 <- ctmaFit(ctmaInitFit = CoTiMAInitFit_6_NUTS,
                                invariantDrift = c("V2toV1", "V1toV2"),
                                coresToUse = -1)
saveRDS(CoTiMAFullInv23Fit_6, file=paste0(activeDirectory,
                                "CoTiMAFullInv23Fit_6.rds"))
summary(CoTiMAFullInv23Fit_6)

CoTiMAFullInvEq23Fit_6 <- ctmaEqual(CoTiMAFullInv23Fit_6,
                                    coresToUse = -1)
saveRDS(CoTiMAFullInvEq23Fit_6, file=paste0(activeDirectory,
                                "CoTiMAFullInvEq23Fit_6.rds"))
summary(CoTiMAFullInvEq23Fit_6)
```

Figure 33: Two-step procedure for testing the equality of two cross effects (`ctmaFit`, `ctmaEqual`)

Third (3), the CoTiMA fit object returned then serves as an argument for `ctmaEqual`. The code for Step 2 and 3 is shown in Figure 33.

We skip displaying the output returned from `summary(CoTiMAFullInv23-Fit_6)` here because it is sufficient to note that V1toV2 = .0131, V2toV1 = .0160, -2ll = 14007.6, and the number of estimated parameters = 40. *V1toV2* and *V2toV1* were the only parameters that were aggregated, that is, invariant across primary studies. This is recognized by `ctmaEqual`, which, in addition to their invariance, constrains *V1toV2* and *V2toV1* to be equal. We call this the *equality model*. Again, we skip displaying the output returned from `summary(CoTiMAFullEq23Fit_6)` here because it is sufficient to note that *V1toV2* = *V2toV1* = .0144, -2ll = 14007.74, and the number of estimated parameters = 39.

The -2ll difference test examines if the fit (-2ll value) of the equality model

```
## [1] "  ###   NEXT MODEL COMPARISON   ###"
## [2] "Model:  V2toV1 unequal but invariant across samples"
## [3] "V1toV2 unequal but invariant across samples"
## [4] "COMPARED WITH   Model: V2toV1_eq_V1toV2"
## [5] "Diff_Minus2LL:  0.144156491554895"
## [6] "Diff_df (= Diff_n.params):  1"
## [7] "prob:  0.70418336974162"
## [8] "Message1:  A prob value < .05 indicates a significant difference."
```

Figure 34: Result of the -2ll difference test comparing the fit of the invariance model with the fit of the equality to test if two cross effects are equal (`ctmaEqual`)

is statistically worse than the fit of the invariance model. If this would be the case, then the hypothesis that both effects are equal has to be rejected and the alternative hypothesis that one effect (*V2toV1* in this example) is significantly larger than the other one (*V1toV2* in this example), will be retained. The -2ll difference test is automatically performed by `ctmaEqual`, too, it is displayed at the end of the `summary(CoTiMAFullInv23Fit_6)`, and it is shown in Figure 34. In our example, the -2ll difference test was not significant. Thus, we could not reject the hypothesis that *V1toV2 = V2toV1*.

Finally, we shall mention the `ctmaCompFit` function that comes with the Co-TiMA package. The `ctmaCompFit` function is automatically used by `ctmaEqual`. It can also be applied whenever researchers want to compare two model fits with a -2ll difference test by using `ctmaCompFit(CoTiMAFit1, CoTiMAFit2)`. Note, however, that the result is only valid if the two models are nested, that is, the second model is derived from the first model by constraining parameters. Such constraints are present, for example, if parameters are eliminated from a model by constraining them to be 0.0, or by constraining other parameters to be equal. The former is achieved by setting the desired drift effect to "0", and the latter is achieved by assigning identical labels to the desired drift effects. This could be done with the `ctmaInit` and `ctmaFit` functions. For example, the argument drift=c("V1toV1", 0, 0, "V1toV1") could be used to fit a model that has no cross effects and equal auto effects. This model is nested in a full CoTiMA model because it is more constrained.

7.4   Moderated CoTiMA (ctmaFit)

CoTiMA can handle multiple continuous moderators and multiple categorical moderators, however, it is not yet possible to mix categorical and continuous ones. In general, we recommend starting with a single moderator to foster understanding how they operate before analyzing multiple moderators combined.

Recalling from Figure 18, we entered information about two moderators. The fist was the type of burnout measure applied in a primary study, which was either exhaustion or emotional exhaustion, and which was a categorical moderator. If there were two or more categorical moderators, the moderator numbers and moderator names would have to be provided as vectors (e.g., `mod.number`

```
CoTiMAMod1onFullFit_6 <- ctmaFit(ctmaInitFit = CoTiMAInitFit_6_NUTS,
                                 mod.number = 1,
                                 mod.type = "cat",
                                 mod.names = "Burnout Measure",
                                 coresToUse = -1,
                                 scaleMod = FALSE)
saveRDS(CoTiMAMod1onFullFit_6, file=paste0(activeDirectory,
                                           "CoTiMAMod1onFullFit_6.rds"))
summary(CoTiMAMod1onFullFit_6)
```

Figure 35: A full moderated CoTiMA with a single categorical moderator (`ctmaFit`)

```
## $estimates
##                 row col    Mean     sd    2.5%     50%   97.5% Tvalues
## DRIFT_V1toV1     1   1 -0.0470 0.0047 -0.0570 -0.0469 -0.0383 -9.9174
## DRIFT_V2toV1     1   2  0.0121 0.0033  0.0057  0.0121  0.0185  3.7010
## DRIFT_V1toV2     2   1  0.0094 0.0034  0.0026  0.0095  0.0163  2.7380
## DRIFT_V2toV2     2   2 -0.0471 0.0049 -0.0575 -0.0468 -0.0382 -9.6043
##
## $randomEffects
## NULL
##
## $minus2ll
## [1] 13825.68
##
## $n.parameters
## [1] 44
##
## $mod.effects
##                                                                mean     sd
## 2. smallest value (category) of Burnout Measure_on_V1toV1 -0.1079 0.0082
## 2. smallest value (category) of Burnout Measure_on_V2toV1  0.1460 0.0388
## 2. smallest value (category) of Burnout Measure_on_V1toV2  0.0925 0.0336
## 2. smallest value (category) of Burnout Measure_on_V2toV2 -0.0798 0.0077
##                                                                2.5%     50%
## 2. smallest value (category) of Burnout Measure_on_V1toV1 -0.1250 -0.1076
## 2. smallest value (category) of Burnout Measure_on_V2toV1  0.0693  0.1464
## 2. smallest value (category) of Burnout Measure_on_V1toV2  0.0270  0.0920
## 2. smallest value (category) of Burnout Measure_on_V2toV2 -0.0954 -0.0796
##                                                               97.5%  Tvalues
## 2. smallest value (category) of Burnout Measure_on_V1toV1 -0.0928 -13.1871
## 2. smallest value (category) of Burnout Measure_on_V2toV1  0.2217   3.7573
## 2. smallest value (category) of Burnout Measure_on_V1toV2  0.1589   2.7515
## 2. smallest value (category) of Burnout Measure_on_V2toV2 -0.0654 -10.3318
```

Figure 36: Part of the results moderated full CoTiMA (`ctmaFit`)

= c(1, 3), mod.names = c(''Burnout Measure'', ''Study Quality'')). However, in the present example, we use the first potential moderator variable only (mod.number = 1 in Figure 35), which was categorial (mod.type = ''cat'' in Figure 35) representing two types of burnout measures (mod.names = ''Burnout Measures'' in Figure 35). By default CoTiMA does not standardize moderators. In the present example, we made this explicit by including the argument

scaleMod = FALSE. Thus, the k - 1 dummy variables created from the k categories of the moderator variable use values 0 and 1, which facilitates interpretation of categorical moderator effects.

Part of the results are shown in Figure 36. The drift effects shown in the $estimates section are those in the reference group, which is always the group with the smallest category number. In the present example, these are the primary studies for which the moderator was 1 meaning they used an emotional exhaustion scale to measure burnout.
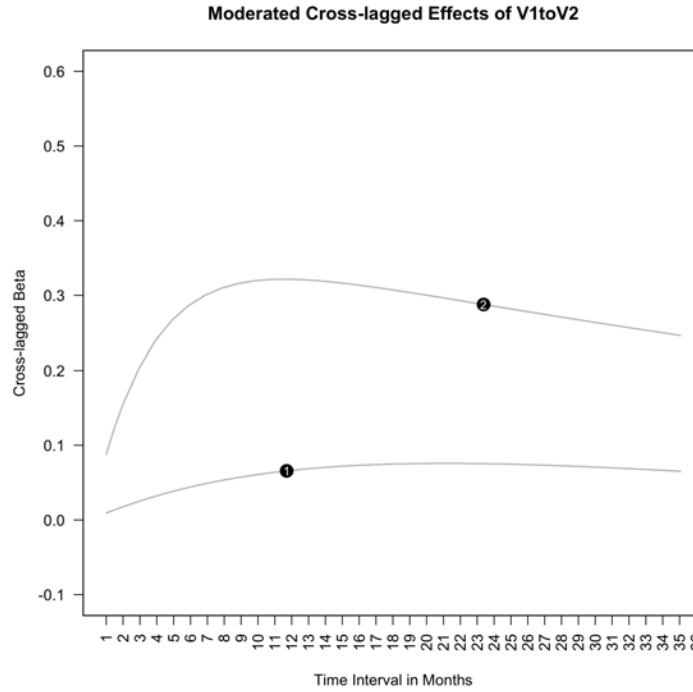

Moderated Cross-lagged Effects of V1toV2

Figure 37: The cross-lagged effect *V1to V2* moderated by type of burnout measure (1 = emotional exhaustion, 2 = exhaustion) from 1 to 36 months. The horizontal location of the category indicators 1 and 2 has no inherent meaning.

The section $mod.effects in Figure 36. show the effects belonging group with the 2nd smallest category number. In case there were more categories, one would find here four additional rows starting with 3. smallest value etc. It is important to note that this section does not show the drift effects. Rather, it shows how in this category, which used an exhaustion compared to emotional exhaustion scale to measure burnout, the drift effects change compared to the reference group. Both auto effects are significantly more negative (i.e., the variables were less stable). Both effects show that in this category the two cross effects were significantly stronger than the already significant cross effects in the reference group. We call this a *positive moderating effect* of the exhaustion

33

scale.

As always, the sizes of continuous time effects are virtually impossible to interpret. For example, the effect *V1toV2* is .0094 for emotional exhaustion and .0094 + .0925 = 0.1019 for exhaustion, the latter thus being roughly ten times as large as the former. However, how these effects unfold over time also depends on the other three effects *V1toV1*, *V2toV2*, and *V2toV1*. Therefore, we used `plot(CoTiMAMod1onFullFit_6, timeUnit = "Months", timeRange = c(1, 36, 1))` to plot the moderated discrete time effects. For *V1toV2*, the course of the moderated effect over discrete time is shown in Figure 39.

```
CoTiMAMod2on23Fit_6 <- ctmaFit(ctmaInitFit = CoTiMAInitFit_6_NUTS,
                               mod.number = 2,
                               mod.type = "cont",
                               mod.names = "Control",
                               moderatedDrift = c("V1toV2", "V2toV1"),
                               scaleMod=TRUE,
                               coresToUse = -1)
saveRDS(CoTiMAMod2on23Fit_6, file=paste0(activeDirectory,
                                         "CoTiMAMod2on23Fit_6.rds"))
summary(CoTiMAMod2on23Fit_6)
plot(CoTiMAMod2on23Fit_6, timeUnit = "Months", timeRange = c(1, 36, 1))
```

Figure 38: A partial moderated CoTiMA with a single continuous moderator (`ctmaFit`)

The code for a partial moderated CoTiMA with a single continuous moderator is shown in Figure 38. In the model, only the cross effects are moderated. It is recommended to standardize continuous moderators, which is achieved by `scaleMod = TRUE`. The summary (not shown) reveals that control significantly reduces *V2toV1* (i.e., the moderating effect) by -.0851 from the average effect, which is *V2toV1* = .2777 (i.e., the main effect).

The plot function in Figure 37 yields the plot shown in Figure 39. Across all time intervals, for people who have low levels of control at work, effects of demands on burnout are larger than for those with high levels of control. In most empirical articles that visualize moderator effects for moderator values at +2*SD* and -2*SD* are not shown. This could be achieved by using `mod.values = c(-1, 0, 1)` as additional argument for the plot function in Figure 37.

8 Bias & Generalizability (`ctmaBiG`)

After finishing the EPIC part of the EPIC-BiG-Power workflow, we can now turn to the first part of the BiG workflow, which is done by using `ctmaBiG`. It performs Egger's tests for drift coefficients (e.g., Sterne & Egger, 2001) and provides PET-PEESE corrections of fixed effect estimates (Stanley & Doucouliagos, 2014). Random effect estimates are also computed. Various measures of heterogeneity (cf. Borenstein, Hedges, Higgins, & Rothstein, 2009) as well as measures of expected replications rates (ERR) and expected discovery rates (EDR; Bartoš &
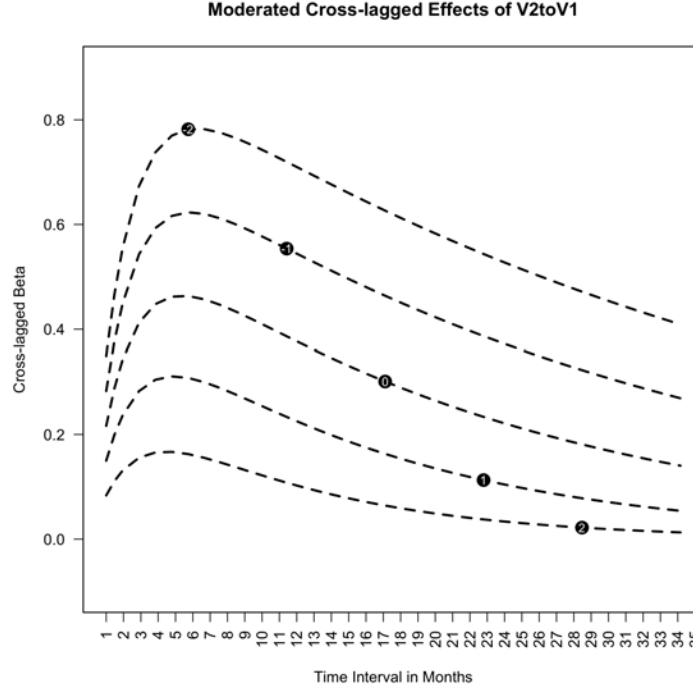
Moderated Cross-lagged Effects of V2toV1

Figure 39: The cross-lagged effect *V2toV1* moderated by control at work from 1 to 30 months. The lines show the effect of *V2toV1* for control at -2*SD* below the mean of control (-2), -1*SD* below the mean of control (-1), at the mean of control (0), +1*SD* above the mean of control (1), and +2*SD* above the mean the mean of control (2). The horizontal location of the SD values has no inherent meaning.

Schimmack, 2020; Brunner & Schimmack, 2020) are also provided by `ctmaBiG`. The return object of `ctmaBiG` can be used to plot funnel plots and forest plots.

To proceed with `ctmaBiG`, we use the init fit-file published in the online repository of Dormann, Guthier, and Voelkle (2020) that belongs to their Co-TiMA of job stressors and burnout[12]. The file containing their init fit-object can be retrieved from the website of the Open Science Foundations with the code shown in Figure 40. Note that Guthier et al. (2020) used a preliminary CoTiMA version that was based on the OpenMx R-package (Boker et al., 2011), whereas the file we suggest downloading was created with the rstan R-package (Stan Development Team, 2020b). The latter samples parameter estimates from

---

[12]The data set of 6 primary studies we have been using up to this point is not really useful for the analyses presented in this section. First, we introduced missing values for study 128 in order to show how this could be done. This could be useful in full and partial CoTiMAs, but it is mathematically impossible to estimate all four cross effects in this study given that V1 at T1 was missing. The model is said to be mathematically not identified. Excluding study 128 would leave us with five studies only, which would not produce reliable results

generated parameter distribution and results thus slightly change from analysis to analysis (unless the argument `finishsamples` is set to a large value, e.g., 10000). So one should expect minor differences compared to the results reported in Guthier et al. (2020). On the other hand, the init fit-object contains all information required to replicate all their results with minor deviations[13]. Note, however, computations would last several hours except of `ctmaBiG`. This is the major reason why we did not use their init fit-object before.

```
dl_link <- "https://osf.io/download/qhpae/"
target_file <- paste0(activeDirectory, "/CoTiMAInitFit_D_BO_stanct.rds")
download.file(dl_link, target_file)
CoTiMAInitFit_D_BO <- readRDS(target_file)
saveRDS(CoTiMAInitFit_D_BO, paste0(activeDirectory, "CoTiMAInitFit_D_BO.rds"))
```

Figure 40: Downloading the Init-Fit file of Guthier et al. (2020)

```
CoTiMABiG_D_BO <- ctmaBiG(CoTiMAInitFit_D_BO)
summary(CoTiMABiG_D_BO)
plot(CoTiMABiG_D_BO)
```

Figure 41: Analysis of bias and generalizability, summary of results, and plotting (`ctmaBiG`)

The analysis of bias and generalizability, summarizing the results, and plotting forest plots and funnel plots is achieved with the code in Figure 41. First, results of fixed effects analyses of *single* drift coefficients are displayed. Recall that in CoTiMA all drift effects (full CoTiMA) or a subset (partial CoTiMA) is aggregated simultaneously, thereby taking the entire causal system into account. Thus, CoTiMA estimates a *set* of fixed effects by a multi group SEM which constraints a set of drift effects to be invariant across groups (i.e., primary studies). Estimation is based on minimization of the discrepancy between the model implied covariance matrices and their empirical counterparts.

Contrary, in terms of a traditional fixed and random effects analysis, the drift effects of all primary studies, which resulted from the initial fitting of ctsem models one by one rather than as a set, are analyzed. Estimation is based on the standard errors of the drift effects rather than on minimizing discrepancies between implied and empirical covariance matrices. The fixed effect estimates of the two cross effects reported in the section `$'Fixed Effects of Drift Coefficients'` of Figure 42 were $V1toV2 = .0024$ ($p < .001$) and $V2toV1 = .0054$ ($p < .001$).

The next section in Figure 42 is `$Heterogeneity`. Here τ^2, H^2, and I^2 are shown, of which I^2 is usually of most interest. Note that estimates of τ^2 were small so even four decimal places are not sufficient to show this. Consequently,

---

[13]In addition to the fitted ctsem models of each primary study, it is possible to extract all information from an init fit-object that were originally complied with `ctmaPrep` by, e.g., `originalStudyList <- initFitObject$primaryStudyList`. Thus, replicability of CoTiMA results is easily enabled by making one's init fit-object available for download in an repository, for example, using the Open Science Framework http://osf.io/.

```
## $`Fixed Effects of Drift Coefficients`
##                           V1toV1  V2toV1 V1toV2    V2toV2
## MeanOfDriftValues        -0.0590  0.0219 0.0112   -0.0539
## FixedEffect_Drift        -0.0219  0.0054 0.0024   -0.0133
## FixedEffect_DriftVariance  0.0000  0.0000 0.0000    0.0000
## FixedEffect_DriftSE       0.0004  0.0004 0.0003    0.0003
## FixedEffect_DriftUpperLimit -0.0211  0.0061 0.0030   -0.0128
## FixedEffect_DriftLowerLimit -0.0227  0.0047 0.0017   -0.0139
## FixedEffect_DriftZ       -54.3360 14.8412 7.4877 -46.5243
## FixedEffect_DriftProb     0.0000  0.0000 0.0000    0.0000
##
## $Heterogeneity
##                     V1toV1   V2toV1   V1toV2     V2toV2
## tau2Drift            0.0001   0.0001   0.0000     0.0001
## Q_Drift            772.8459 534.4175 217.4290 1236.0298
## H2_Drift            16.4435  11.3706   4.6261   26.2985
## H2DriftUpperLimit   18.0367  12.6087   5.2890   28.4874
## H2DriftLowerLimit   14.9911  10.2540   4.0463   24.2778
## I2_Drift            93.9186  91.2054  78.3837   96.1975
## I2DriftUpperLimit   94.9455  92.8478  83.4626   96.7594
## I2DriftLowerLimit   92.6831  89.1858  71.7451   95.5382
##
## $`Random Effects of Drift Coefficients`
##                                 V1toV1  V2toV1  V1toV2    V2toV2
## RandomEffecttot_Drift           -0.0402  0.0114  0.0061   -0.0380
## RandomEffecttot_DriftVariance    0.0000  0.0000  0.0000    0.0000
## RandomEffecttot_DriftSE          0.0021  0.0017  0.0011    0.0021
## RandomEffecttot_DriftUpperLimit -0.0360  0.0147  0.0082   -0.0339
## RandomEffecttot_DriftLowerLimit -0.0444  0.0080  0.0039   -0.0420
## RandomEffecttot_DriftZ         -18.8203  6.6986  5.5511  -18.2134
## RandomEffecttot_DriftProb        0.0000  0.0000  0.0000    0.0000
## RandomEffecttot_DriftUpperLimitPI -0.0169  0.0289  0.0153   -0.0148
## RandomEffecttot_DriftLowerLimitPI -0.0635 -0.0062 -0.0032   -0.0611
```

Figure 42: Part 1 of results of `ctmaBiG`

between study heterogeneity as indicated by I^2 was larger with the exception of the (small) effect $V1toV2$.

The third section ($`Random Effects of Drift Coefficients`) in Figure 42 displays the random effect estimates, their SE, confidence intervals (`Limit`), and the z-values with their associated probability levels. In addition, prediction intervals (`LimitPI`) also allow assessing the degree of heterogeneity. Prediction intervals describe a region in which about 95% of the true study effects are expected to be found (e.g., Guddat, Grouven, Bender, & Skipka, 2012). The effects $V1toV2 = .0061$ ($p < .001$) and $V2toV1 = .0114$ ($p < .001$) were larger than their fixed effects counterparts reported earlier. Note that the corresponding CoTiMA (fixed) effects reported by Guthier et al. (2020) were $V1toV2 = .0039$ ($p < .001$) and $V2toV1 = .0084$ ($p < .001$), and they were right in the middle between the traditional fixed and random effects estimates.

Part 2 of the results returned from `ctmaBiG` is shown in Figure 43. These results directly address possible publication bias. Egger's tests (e.g., Sterne & Egger, 2001) is a statistical test of funnel plot asymmetry. Significant results

```
## $`PET-PEESE corrections`
##                 V1toV1 V2toV1 V1toV2  V2toV2
## PET_Drift       -0.0148 0.0031 0.0010 -0.0079
## PET_SE           0.0014 0.0015 0.0008  0.0010
## PEESE_Drift     -0.0206 0.0048 0.0021 -0.0126
## PEESE_SE         0.0013 0.0012 0.0007  0.0013
## PET_PEESE_Drift -0.0206 0.0048 0.0010 -0.0126
## PET_PEESE_SE     0.0013 0.0012 0.0008  0.0013
## WLS_Drift       -0.0219 0.0054 0.0024 -0.0133
## WLS_SE           0.0016 0.0012 0.0007  0.0015
##
## $`Egger's tests`
##            V1toV1 V2toV1 V1toV2  V2toV2
## Egger's b0 -3.9484 1.4749 1.0973 -4.9827
## SE(b0)      0.5032 0.5854 0.3510  0.5146
## T          -7.8459 2.5196 3.1259 -9.6831
## p           0.0000 0.0153 0.0031  0.0000
```

Figure 43: Part 2 of results of `ctmaBiG`

indicate that small-N studies produced larger effect sizes (i.e., more positive, if the true effect is positive & more negative, if the true effect is negative), suggesting that the aggregated effects are biased. Thus, the results in the $`Egger's tests` part of Figure 43 suggest that the cross effects are biased upwards, and the two auto effects are biased downwards. The latter means that demands and burnout in small-N studies are less stable than in large-N studies. This could have many reasons. For instance, if job stress studies with small-N were based on single organizations or single occupations, variance might be restricted, implying lower test-retest correlations eventually resulting in smaller auto effects. Contrary, this reasoning would also imply smaller cross effects, which was not the case. Selective reporting might be a more plausible reason here.

Precision-effect test and precision effect estimates with standard errors (PET-PEESE; Stanley & Doucouliagos, 2014) removes small sample bias (selective reporting) from the fixed effect estimates in an 'aggressive' fashion (Stanley, Carter, & Doucouliagos, 2018, p. 1333). PET-PEESE involves a decision rule when PET or PEESE is more important. The result of this decision is the `PET_PEESE_Drift` row in the section $`PET-PEESE corrections` of Figure 43. The `WLS_Drift` estimates of the auto effects $V1toV1$ and $V2toV2$, which are identical to the fixed effect estimates in Figure 42 (but have more appropriate SE), are more negative compared to their corrected `PET_PEESE_Drift` counterparts, but the differences are not very large. This also applies to the $V2toV1$ cross effects, representing the effect of earlier burnout on later burnout. However, PET-PEESE of $V1toV2 = .0010$, which is less than $1/5$ of the fixed effect. Hence, the true effect of earlier demands on later burnout is probably much smaller than suggested by the fixed effect estimate.

Results of Z-Curve 2.0 (Bartoš & Schimmack, 2020) analyses including Expected Replication Rates (ERR) and Expected Discovery Rates (EDR) based are displayed in Figure 44. Figure 44 is limited to the auto effect $V1toV1$ and the cross effect $V1toV2$ for space reasons.

```
## $`Z-Curve 2.0 Results:`$`Z-Curve 2.0 analysis of
## V1toV1`
##  Call:
##  zcurve::zcurve(z = tmp1)
##  model: EM via EM
##      Estimate  l.CI  u.CI
##  ERR    0.979 0.905 1.000
##  EDR    0.710 0.386 1.000
##  Model converged in 24 + 69 iterations
##  Fitted using 25 z-values. 48 supplied, 47 significant
## (ODR = 0.98, 95% CI [0.88, 1.00]).
##  Q = -27.94, 95% CI[-36.08, -18.51]
##  $`Z-Curve 2.0 Results:`$`Z-Curve 2.0 analysis of
## V1toV2`
##  Call:
##  zcurve::zcurve(z = tmp1)
##  model: EM via EM
##      Estimate  l.CI  u.CI
##  ERR    0.573 0.224 0.824
##  EDR    0.507 0.050 0.795
##  Model converged in 14 + 221 iterations
##  Fitted using 20 z-values. 48 supplied, 21 significant
## (ODR = 0.44, 95% CI [0.30, 0.59]).
##  Q = -19.87, 95% CI[-28.16, -9.05]
```

Figure 44: Expected Replication Rates (ERR) and Expected Discovery Rates (EDR) based on Z-Curve 2.0 analysis of *V1toV1* and of *V1toV2* by `ctmaBiG`.

ERR is the probability of finding a significant effect in an exact replication study. ERR is also called conditional mean power, that is, the mean power of the *subset* of all conducted studies that produced significant effects. The unconditional power, that is, the mean power of *all* conducted studies, is called EDR. To put it differently, EDR is the proportion of all studies (published and unpublished) that found significant effects. EDR can be compared with the actually Observed Discovery Rate (ODR), that is, the proportion of significant effects in the subset of studies that have actually been published or otherwise identified as useful for CoTiMA. If ODR is high and EDR is low, publication bias is likely.

For the auto effect *V1toV1*, one might be tempted to believe that it should virtually always be replicated (some stability over one month should be present). However, the credible interval of *V1toV1* in the study by Jimenez and Dunkl (2017) included 0 (cf. Guthier et al., 2020, Table 1), so that only 47 out of 48 studies produced a significant effect, corresponding to the ODR of .98 in Figure 44. Still, ERR and EDR are very large and the CI always include 1.00. For *V1toV2*, the difference between ODR and EDR is not too large, either, but in this case EDR exceeds ODR. This could happen, especially if ODR is within the 95%CI of EDR, and in most instances it could occur simply due to sampling error when there is no publication bias.

Funnel plots and forest plots could be obtained with `plot(CoTiMABiG_D_BO)`. Funnel plots represent the graphical counterpart of Egger's tests, and they plot

standard error of effects (an indicator of small N; y-axis; large at the bottom & low at the top) against the effect size (x-axis). Without small-N bias, funnel plots would be symmetric. Conversely, funnel plot asymmetry indicates small-N bias. The funnel plot of *V1toV2*, for which Eggers's and PET-PEESE indicated large bias, is shown in Figure 45. Effect sizes are clearly asymmetrically distributed on the right hand side, particularly at the bottom where effect sizes of small-N studies (with large SE) are located.
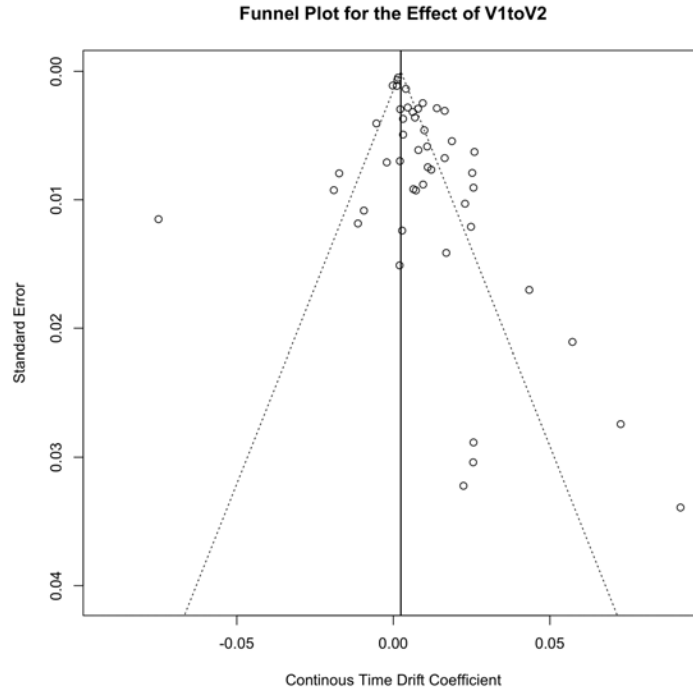

Funnel Plot for the Effect of V1toV2

Figure 45: Funnel plot of the effect *V1toV2* of the fit object returned by `ctmaBiG` (`plot`)

A better impression of the effects obtained in all primary studies is provided in forest plots. The effects for each of the primary studies is represented by a square and their confidence intervals are represented by horizontal lines through these squares. A forest plot of the *V1toV2* effect is shown in Figure 46. The squares vary in size depending on their sample sizes, and they are sometimes small because sample sizes varied considerably across primary studies. The diamond at the bottom shows the aggregated fixed effect. There is no visible horizontal line for its confidence interval because the overall SE was very small and, thus, the confidence interval is rather narrow.
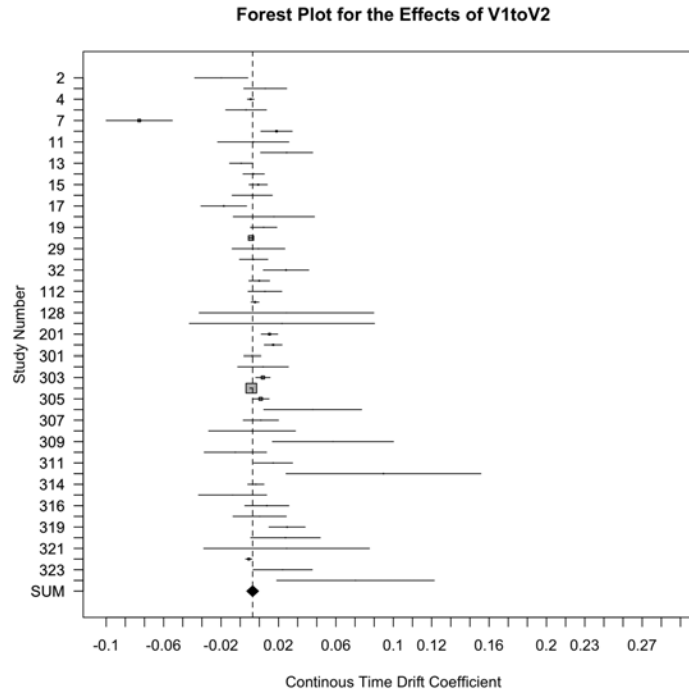
Figure 46: Forest plot of the effect *V1toV2* of the fit object returned by `ctmaBiG` (`plot`)

9 Statistical Power (`ctmaPower`)

Finally, we can turn to the Power part of the EPIC-BiG-Power workflow, which can be performed with `ctmaPower`. It conducts two types of analyses. First, it estimates required sample sizes for a range of different time intervals to achieve a desired statistical power. This is important for designing future studies. Second, it calculates the expected power for all primary studies (some-times also referred to as post hoc power or retrospective power). This is important to know if past studies might have failed to replicate effects with statistical significance because they were under-powered.

```
CoTiMAPower_D_BO <- ctmaPower(ctmaInitFit=CoTiMAInitFit_D_BO,
                              statisticalPower = c(.50, .80, .95),
                              coresToUse = -1,
                              finishsamples = 10000)
saveRDS(CoTiMAPower_D_BO, file=paste0(activeDirectory,
                                      "CoTiMAPower_D_BO.rds"))
summary(CoTiMAPower_D_BO)
```

Figure 47: Calculating expected (post hoc) power for three different probability levels (`ctmaPower`)

41

To calculate statistical power, a highly restrictive CoTiMA model has to be estimated. The most restrictive CoTiMA model estimated so far was the full CoTiMA, in which all drift effects were constrained to be invariant across primary studies. To calculate statistical power, a more restrictive model is required that, in addition, constraints the variance and covariances at T0 as well as the diffusion coefficients to be invariant. Stated differently, one has to assume all samples analyzed in the primary studies were drawn from the same population. There are several arguments that can be used with `ctmaPower`, and they are enumerated in the Appendix. In most cases, requesting the desired levels of power in addition to the init fit-object is probably sufficient. We used the code in Figure 47 for generating the subsequently discussed output and the figures.

```
##                          V1toV1   (SE)    Tvalue V2toV1   (SE) Tvalue V1toV2
## Fixed Effects Drift      -0.0525 0.0009 -58.3333 0.0164 0.0008   20.5 0.0119
## Fixed Effects Diffusion  0.0975 0.0013  75.0000 0.0096 0.0008   12.0 0.0096
## Fixed Effects T0Var      0.9982 0.0087 114.7356 0.3757 0.0065   57.8 0.3757
##                            (SE) Tvalue V2toV2   (SE)    Tvalue
## Fixed Effects Drift      0.0007   17.0 -0.0428 0.0007 -61.1429
## Fixed Effects Diffusion 0.0008   12.0  0.0818 0.0010  81.8000
## Fixed Effects T0Var     0.0065   57.8  0.9984 0.0087 114.7586
```

Figure 48: Estimates of drift parameters using a model with all variance and covariances at T0, all drift effects, and all diffusion coefficients invariant across primary studies (`ctmaPower`)

Then, `summary(CoTiMAPower_D_BO)` creates a large output on the console that we again discuss in parts. Figure 48 displays the parameter estimates of the model with *all* effects being invariant. These are the parameter estimates that are regarded as the *true* effects (mean of the distribution of true effects). In concert with the sample sizes and the time intervals of the primary studies (both are taken from `CoTiMAInitFit_D_BO` and do not need to be provided as arguments) the true effects determine the statistical power of the primary studies to achieve significance levels of $\alpha = .05$ and $\alpha = .01$. Further, across a range of time intervals (could be provided with the argument `timeRange`; otherwise it is from 1 to 1.5 times the longest interval used in primary studies), the true effects determine the required sample sizes to and achieve the requested levels of statistical power.

The next section in the generated output reports the expected power of primary studies. For the effect of *V1toV2*, this is displayed in Figure 49. Note that in Guthier et al. (2020) we reported numerical problems in estimating the expected statistical power across short time intervals – since then we solved this issue. We left out several studies (6 to 23 & 28 to 47) for space reasons here. Assuming the aggregated effects in Figure 49 are the true effects, the probability values in Figure 49 represent the statistical power each primary study had to detect the focal true *V1toV2* effect (i.e., .0119; see Figure 48) with $p < .05$ and $p < .01$. For those studies with more than two measurement occasions, the statistical power is reported for all adjacent time intervals. At the bottom,

```
##              N   Lag  Power (.05) Power (.01) Lag  Power (.05) Power (.01)
## Study_No_1  148  12   0.2068      0.0751      <NA> <NA>        <NA>
## Study_No_2  188  12   0.2513      0.0983      <NA> <NA>        <NA>
## Study_No_3  556  96   0.0527      0.0127      <NA> <NA>        <NA>
## Study_No_4  261  12   0.3306      0.145       <NA> <NA>        <NA>
## Study_No_5  1378 18   0.9461      0.839       <NA> <NA>        <NA>
## ...         ...  ...  ...         ...         ...  ...         ...
## Study_No_24 195  3    0.1343      0.0422      <NA> <NA>        <NA>
## Study_No_25 999  12   0.8474      0.6581      12   0.8474      0.6581
## Study_No_26 668  12   0.6844      0.4449      12   0.6844      0.4449
## Study_No_27 370  12   0.4419      0.222       12   0.4419      0.222
## ...         ...  ...  ...         ...         ...  ...         ...
## Study_No_48 171  3    0.1228      0.0375      <NA> <NA>        <NA>
## Mean        <NA> <NA> 0.3957      0.2407      <NA> <NA>        <NA>
## Median      <NA> <NA> 0.3142      0.1348      <NA> <NA>        <NA>
```

Figure 49: Expected (post hoc) power across primary studies (`ctmaPower`)

median and mean statistical power across all primary studies is shown. For instance, the median statistical power was .3142 to find a significant *V2toV1* effect with $p <. 05$. As in most meta-analyses, this demonstrates that many primary studies are heavily under-powered and finding a significant effect is less likely than like getting heads-up when flipping a coin.

```
##        V1toV2 Power=0.5 V1toV2 Power=0.8 V1toV2 Power=0.95
## 1      2737             5589             9251
## 1.5    1876             3830             6339
## 2      1447             2953             4888
## 3      1020             2082             3445
## 4      810              1652             2733
## ...    ...              ...              ...
## 15     416              847              1400
## 16     415              844              1396
## 17     415              846              1399
## 18     418              851              1407
## 19     421              859              1420
## 20     427              869              1437
## 21     433              882              1459
## ...    ...              ...              ...
## 142    345410           705741           1168433
## 143    368470           752857           1246439
## 144    393512           804022           1331148
## ---    ---              ---              ---
## Min N  415              844              1396
```

Figure 50: Required sample sizes to achieve requested levels of statistical power across a range of time intervals (`ctmaPower`)

The generated output further shows the required samples sizes for (future) studies to obtain significant effects across different time intervals (Figure 50). Note that in Guthier et al. (2020) we reported numerical problems in estimating the required samples across short time intervals – since then we solved this issue. For most effects and most desired levels of statistical power, required sample sizes are lowest around 16-18 month intervals. We show how to plot

required sample sizes against time interval later. Note that the output showing the required sample sizes would also display the expected (discrete time) effect sizes, which we omitted from Figure 50.

The last interesting output deals with combinations of possible time intervals and samples sizes, and it informs about the range of time intervals across which one could expect significant effects. If neither a sample size (`failSafeN`) nor a p-level (`failSaveP`) is provided as function argument, the average sample size of the primary studies is used (otherwise the values assigned to `failSafeN`) and $p < .01$ (otherwise the values assigned to `failSaveP`) are used. As the `$estimates$'Range of significant effects'` section in Figure 51 reports, with N corresponding to the average $N = 549$ across primary studies, one should select time intervals between 8-32 months to find a significant *V2toV1* effect. With the average N used in primary studies, one cannot expect finding a significant *V1toV2* effect across neither time interval.

```
## [1] The shortest interval across which the effect (V2toV1) is
## significant  with p < 0.01 assuming N = 549 ( = avg. N) is 8.  The
## longest interval across which the effect (V2toV1) is significant  with
## p < 0.01 assuming N = 549 ( = avg. N) is 32.  Note that you have not
## provided an explicit time range for analysis of statistical power. The
## time intervals used ranged from 1 to 1.5 times the longest interval
## used in the primary studies, using integer steps of 1.0. These
## intervals were then augmented by time intervals found in primary
## studies that were non-integers.
##  [2] There is no shortest interval across which the effect (V1toV2) is
## significant  with p < 0.01 assuming N = 549 ( = avg. N).  There is no
## longest interval across which the effect (V1toV2) is significant  with
## p < 0.01 assuming N = 549.  Note that you have not provided an
## explicit time range for analysis of statistical power. The time
## intervals used ranged from 1 to 1.5 times the longest interval used in
## the primary studies, using integer steps of 1.0. These intervals were
## then augmented by time intervals found in primary studies that were
## non-integers.
```

Figure 51: Expected range across which significant effects could be expected (`ctmaPower`)

Finally, required sample sizes can be plotted. We used `plot(CoTiMAPower-_D_BO, timeUnit="Months", timeRange=c(1, 84, 1) )` to generate the plot displayed in Figure 52. This figure is based on the values previously shown in parts in Figure 50
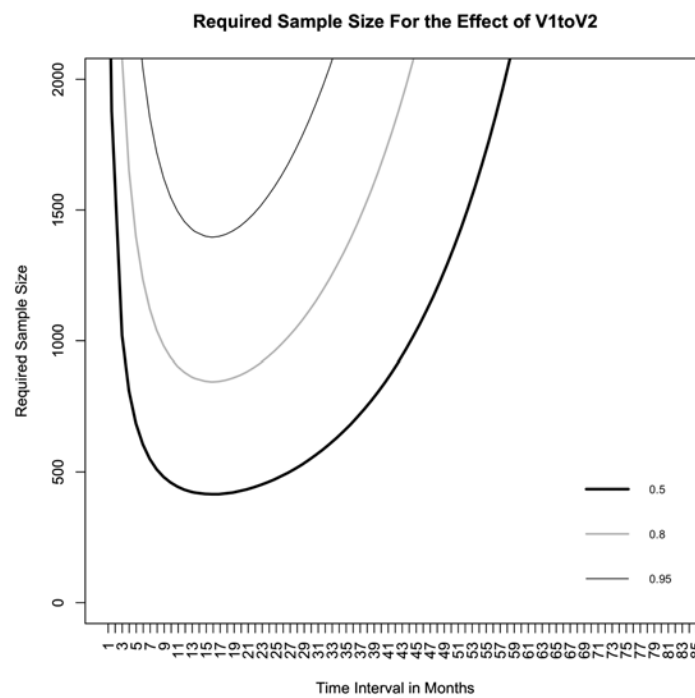
Figure 52: Required sample sizes across time to achieve a statistical power of .50, .80, and .95 (`plot`)

References

[1] Boker, S. M., Neale, M., Maes, H., Wilde, M., Spiegel, M., Brick, T., . . ., & Fox, J. (2011). OpenMx: An open source extended structural equation modeling framework. *Psychometrika, 76,* 306–317. https://doi.org/10.1007/s11336-010-9200-6

[2] Borenstein, M., Hedges, L. V., Higgins, J. P., & Rothstein, H. R. (2009). *Introduction to meta-analysis.* Hoboken, NJ: John Wiley & Sons Inc.

[3] Bartoš, F. & Schimmack, U. (2020; preprint). Z-curve 2.0: Estimating Replication Rates and Discovery Rates. *Meta-Psychology.*

[4] Brunner, J. & Schimmack, U. (2020). Estimating population mean power under conditions of heterogeneity and selection for significance. *Meta-Psychology, 4,* MP.2018.874, https://doi.org/10.15626/MP.2018.874

[5] Carpenter, B., Hoffman, M. D., Brubaker, M., Lee, D. D., Li, P., & Betancourt, M. (2015). *The Stan Math Library: Reverse-Mode Automatic Differentiation in C++.* CoRR abs/1509.07164.

[6] Dormann C., & Homberg, M. (2020). GitHub repository, https://github.com/CoTiMA/CoTiMA.

[7] Dormann, C., Guthier, C., & Voelkle, M. (2020, December 11). *CoTiMA Burnout.* Retrieved from osf.io/e92jd

[8] Driver, C. C, Oud, J. H. L., & Voelkle, M. C. (2017). Continuous Time Structural Equation Modeling with R Package ctsem. *Journal of Statistical Software, 77* (5), 1–35. https://doi.org/10.18637/jss.v077.i05

[9] Driver, C. C., & Voelkle, M. C. (2018). Hierarchical Bayesian continuous time dynamic modeling. *Psychological Methods, 23* (4), 774–799. http://dx.doi.org/10.1037/met0000168

[10] Guddat, C., Grouven, U., Bender, R. & Skipka, G., (2012). A note on the graphical presentation of prediction intervals in random-effects meta-analyses. *Systematic Reviews, 1,* 1–34. https://doi.org/10.1186/2046-405

[11] Guthier, C., Dormann, C. & Voelkle, M.C. (2020). Reciprocal Effects between Job Stressors and Burnout: A Continuous Time Meta-Analysis of Longitudinal Studies. *Psychological Bulletin, 146*(12), 1146–1173. https://doi.org/10.1037/bul0000304

[12] Jimenez, P., & Dunkl, A. (2017). The buffering effect of workplace resources on the relationship between the areas of worklife and burnout. *Frontiers in Psychology, 8,* 12. http://dx.doi.org/10.3389/fpsyg.2017.00012

[13] R Core Team (2020). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

[14] RStudio Team (2020). *RStudio: Integrated Development for R.* RStudio, PBC, Boston, MA URL http://www.rstudio.com/.

[15] Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology, 15* (1), 72–101. JSTOR 1412159

[16] Stan Development Team. (2020a). *Stan Modeling Language Users Guide and Reference Manual, 2.21.2.* https://mc-stan.org Stan Development Team (2020b).

[17] *RStan: the R interface to Stan. R package version 2.21.2,* http://mc-stan.org/.

[18] Stanley, T. D., & Doucouliagos, H. (2014). Meta-regression approximations to reduce publication selection bias. *Research Synthesis Methods, 5(1),* 60–78. http://dx.doi.org/10.1002/jrsm.1095

[19] Stanley, T. D., Carter, E. C., & Doucouliagos, H. (2018). What meta-analyses reveal about the replicability of psychological research. *Psychological Bulletin, 144(12),* 1325–1346. http://dx.doi.org/10.1037/bul0000169

[20] Sterne, J. A., & Egger, M. (2001). Funnel plots for detecting bias in meta-analysis: Guidelines on choice of axis. *Journal of Clinical Epidemiology, 54(10),* 1046–1055. http://dx.doi.org/10.1016/S0895-4356(01)00377-8

[21] Venables WN, Ripley BD (2002). *Modern Applied Statistics with S, Fourth edition.* Springer, New York. ISBN 0-387-954570, http://www.stats.ox.ac.uk/pub/MASS4/.

Appendix. Overview of CoTiMA Functions and their Arguments

ctmaBiG (EPIC-*BiG*-Power)			Performs fixed effect, random effect analysis, and analysis of publication bias (Egger's tests, PET-PEESE corrections, z.curve analysis).
Argument	Default	Possible Values	Explanation
`ctmaInitFit`	NULL	CoTiMA fit object	CoTiMA fit object created with `ctmaInit`.
`activeDirectory`	path used to create ctmaInit-Fit object	character string	Specifies the directory where required files are found and saved. Should end with "/". For example, **''/Users/GDC/CoTiMA/''**
`PETPEESEalpha`	.10	values between `0` and `1`	Probability level (condition) below which to change from PET to PEESE.
`activateRPB`	FALSE	FALSE/ TRUE	Messages (warning, finished fitting) could be send to mobile phone if TRUE.
`digits`	4	value > 0	Rounding used in output.

ctmaCompFit (EPI*C*-BiG-Power)			Compares the fit of two nested CoTiMa models via -2ll difference test.
Argument	Default	Possible Values	Explanation
`model1`	NULL	CoTiMA fit object	CoTiMA fit object created with `ctmaInit`, `ctmaFit`, or `ctmaEqual`.
`model2`	NULL	CoTiMA fit object	CoTiMA fit object created with `ctmaInit`, `ctmaFit`, or `ctmaEqual`.

ctmaCorRel (E*P*IC-BiG-Power)			Corrects (disattenuates) correlation matrix for unreliabilities.
Argument	Default	Possible Values	Explanation
`empcov`	NULL	symmetric correlation matrix	
`alphas`	NULL	vector of the same length as dimensionality of `empcov`	

ctmaEmpCov (E*P*IC–BiG–Power)			Changes a full correlation matrix by selecting target variables, recode them, combine them (add), and add rows/columns with NA if focal variables are missing.
Argument	Default	Possible Values	Explanation
targetVariables	NULL	vector of variable positions or variable names	Selects desired variables in empcov$i$ (i.e., deletes those variables that should not be analyzed). For example, `c(1, 2, 4, 5)` deletes the 3rd and 6th row and column in a 6×6 empcov$i$. Instad of positions, variable names could be used if dimnames were assigned to empcov$i$.
recodeVariables	c()	vector of variable positions or variable names	Recodes desired variables in empcov$i$ (i.e., changes the signs of the correlations). For example, `c(1, 4)` changes the signs of the correlations in the 1st and 4th row of empcov$i$. Instad of positions, variable names could be used if dimnames were assigned to empcov$i$. Note that if numbers are used, they should correspond to the positions in the targetVariables$i$ object rather than the ros/columns in the empcov$i$ object (i.e., recoding is done after targetVariables$i$ were selected from empcov$i$).
combineVariables	c()	list of (vectors of) variable positions	Creates composite variables (i.e., means of one or more variables). Variables that should be combined have to be listed in a vector. Variables that should not be combined have to be listed, too. For example, `list(1, c(2, 3), 4, c(5, 6))` combines the 2nd and 4rd as well as the 5th and 6th variables of empcov$i$. and leaves the 1st and 4th variable untouched. Instead of positions, variable names could be used if they were also used in the argument targetVariables.
combineVariables-Names	c()	vector variable names for combined variables	not yet operational
missingVariables	c()	vector of variable positions	Augments empcov$i$ and pairwiseNi by rows and columns containing NA in order to create matrices of the desired dimension. For example, if the desired matrix should contain correlations of the four variables x_0, y_0, , x_1 and y_1, but a primary study did not measure y_1, then the 4th variable is missing and the correlation matrix returned by ctmaEmpCov will be a 4×4 empcov$i$ and a 4×4 pairwiseNi with NA in the 4th row and in the 4th column, respectively.

49

ctmaEmpCov (E*P*IC-BiG-Power)			Changes a full correlation matrix by selecting target variables, recode them, combine them (add), and add rows/columns with NA if focal variables are missing.
nlatents	NULL	value > 0	The number of (latent) variables. Actually it is the number of all variables at T0. A distinction between latent and manifest variables is not made here.
Tpoints	NULL	value > 1	The number of time points.
sampleSize	NULL	value > 0	The sample size. It does not need to be specified if pairwiseN*i* is provided instead.
pairwiseN	NULL	symmetrix matrix of same dimensions as empcov containing possible pairwiseN.	A matrix with sample sizes for each correlation of empcov*i*.
empcov	NULL	symmetric correlations matrix	Correlation matrix reported in a primary study.

ctmaEqual (EPI*C*-BiG-Power)			Statistically tests if the two or more invariant drift parameters in the Co-TiMAFit object supplied are equal
Argument	Default	Possible Values	Explanation
ctmaInvariantFit	NULL	CoTiMA fit object	CoTiMA fit object thas was returned by ctmaFit. In most cases this is probably the fit of a model in which two effects were specified with the invariantDrift argument (e.g., two cross effects).
activeDirectory	path used to create ctmaInit-Fit object	path to directory	Specifies the directory where required files are found and saved. Should end with "/".
activateRPB	FALSE	FALSE/ TRUE	Messages (warning, finished fitting) could be send to mobile phone if TRUE.
digits	4	value > 0	Rounding used in output.
coresToUse	1	value > 0 or < 0	The number of cores (threads) to be used for fitting. If a negative values is specified, the value is subtracted from available cores, else the value sets the number of cores to be used. Should usually be 1 on Windows OS.

ctmaEx (-)			Loads R objects created in the examples (figures) of this User's Guide into the user's R environment
Argument	Default	Possible Values	Explanation
example	NULL	The figure in this User's Guide in which the desired R oibjects are created	The ' Figure' + number (e.g., 'Figure 2') which contains the R objects that should be loaded or fitted. For example, `ctmaEx('Figure 2)` loads `empocv2`, `delta_t2` in the user's R environment
do	'add'	'add' or 'remove'	Specifies wether the desired R objects should be added or removed from the user's R environment. For example, `ctmaEx("Figure 3", fit=TRUE)` creates the `CoTiMAstudyList_3` by applying the `ctmaPrep` function.
fit	FALSE	FALSE/ TRUE	Specifies of CoTiMA objects (i.e. objects created wi a `ctmaXxxx` function) should only be loaded or actually be fitted

ctmaFit (EPI$\mathcal{C}$-BiG-Power)			Fits a CoTiMA model.
Argument	Default	Possible Values	Explanation
ctmaInitFit	NULL	CoTiMA fit object	Object to which all single ctsem fits of primary studies has been assigned to (i.e., what has been returned by `ctmaInit`)
primaryStudyList	NULL	list of primary studies	A list of primary studies compiled with `ctmaPrep` that containes a subset of studies included in `ctmaInitFit`. Useful to exclude studies without the need to use `ctmaInit` again.
cluster	NULL	vector of same length as number of primary studies	Vector with cluster variables (e.g., countries), e.g., `c(1, 1, 1, 3, 3, 6, 7, 8)`. Has to be set up carfully. Will be included in `ctmaPrep` later.
activeDirectory	path used to create ctmaInit-Fit object	path to directory	Specifies the directory where required files are found and saved. Should end with "/".
activateRPB	FALSE	FALSE/ TRUE	Messages (warning, finished fitting) could be send to mobile phone if `TRUE`.
digits	4	value > 0	Rounding used in output.
drift	NULL (=all)	vector (!) of rowwise drift matrix elements	Labels for drift effects that should or should not be included. Have to be either of the type `V1toV2` or `0` for effects to be excluded, which is usually not recommended, e.g., `c("V1toV1", "V2toV1", 0, "V2toV2")`
invariantDrift	NULL (=all)	vector of drift effects	Labels for drift effects that should be aggregated. Have to be of the type `V1toV2`, e.g., `c("V2toV1")`

ctmaFit (EPI𝒞-BiG-Power)			Fits a CoTiMA model.
moderatedDrift	NULL (=all)	vector of drift effects	Labels for drift effects that should be moderated. Have to be of the type V1toV2, e.g., c("V2toV1"). Is only used if moderators are specified.
equalDrift	NULL	vector of 0 or 2 or more drift effects	Labels for drift effects that should be set equal. Have to be of the type V1toV2, e.g., c("V2toV1", "V1toV2"). Is useful, for example, to test if the sizes of cross effects are statistically different.
mod.number	NULL	vector of positions of moderators	The position(s) of the moderator(s) in the the vector of moderator values compiled with ctmaPrep, which should be used in a moderated CoTiMA, e.g., c(1, 3).
mod.type	"cont"	"cont" or "cat"	Type of moderator(s). Categorical and continuous moderators could not be used in combination, but more than one continuous or more than one categorical moderator is possible.
mod.names	NULL	(vector of) character object(s)	Names used to label moderators in the output .
indVarying	FALSE	TRUE / FALSE	Specifies a random (manifest) intercept model. Still experimental.
coresToUse	1	value > 0 or < 0	The number of cores (threads) to be used for fitting. If a negative values is specified, the value is subtracted from available cores, else the value sets the number of cores to be used. Should usually be 1 on Windows OS.
scaleTI	NULL (=FALSE)	TRUE / FALSE	With the move from the omx (OpenMX) version of ctsem to the stanct (stan) version, CoTiMA moved from fitting a multi group model to a model in which the groups are represented by dummy variables. These are internally handled as time independent (TI) predictors, and scaleTI specifies whether or not these dummy variables should be standardized. Usually not recommended and set to FALSE by taking the FALSE value from the CoTiMAStanctArgs list, which could also be used to specify more stanct fitting parameters.
scaleMod	NULL (=FALSE)	TRUE / FALSE	Whether or not moderators should be standardized. Not recommended for categorical moderators, but recommended for continuous moderators. The default value FALSE is taken from the CoTiMAStanctArgs list, which could also be used to specify more stanct fitting parameters

ctmaFit (EPI$\mathcal{C}$-BiG-Power)			Fits a CoTiMA model.
scaleClus	NULL (=TRUE)	TRUE / FALSE	These are internally handled as time independent (TI) predictors, and scaleClus specifies whether or not these dummy variables should be standardized. TRUE (default) yields avgerage drift estimates, FALSE yields drift estimates of last cluster. Not that the last cluster could represent all clusters that have a single member only. The default value TRUE is taken from the CoTiMAStanctArgs list, which could also be used to specify more stanct fitting parameters
scaleTime	NULL (=FAlSE)	value > 0	Wether or not the time scale used for delta_ti should be changed.For example, scaleTime=1/12 could change the time scale from months to years. It is usually recommended to avoid delta_ti larger than 6. The default value FALSE is taken from the CoTiMAStanctArgs list, which could also be used to specify more stanct fitting parameters.
optimize	TRUE	TRUE / FALSE	The optimize argument is passed to ctStanFit. If FALSE, Bayesian estimations is used. The chosen sampler is conditional on the nopriors argument. Note that this works differently than the optimise argument of ctSatFit.
nopriors	TRUE	TRUE / FALSE	Consequences of TRUE or FALSE are conditional on the optimize argument. optimize=TRUE & nopriors=TRUE implies maximum likelihood estimation, optimize=TRUE & nopriors=FALSE implies maximum a posteriori estimation, optimize=FALSE & nopriors=TRUE implies Bayesian estimation using HMC (Hamiltonian Monte Carlos sampler), and optimize=FALSE & nopriors=FALSE implies Bayesian estimation using NUTS (No U-Turn Sampler).
finishsamples	NULL (=1000)	values > 0	The finishsamples argument is passed to ctStanFit. It specifies the number of samples to draw for final results computation. Larger (e.g., 10.000) values make results more exactly replicable. Larger values are recommended befor manuscripts are submitted. Very large values (e.g., 100.000) might be helpful if very small effects (e.g., 0.0002) result from estimation.

ctmaFit (EPIC-BiG-Power)			Fits a CoTiMA model.
iter	NULL (=1000)		The iter argument is passed to ctStanFit. It specifies the number of iterations used for Bayesian estimation, half of which will be devoted to warmup.
chains	NULL (=2)		The chains argument is passed to ctStanFit and specifies the number of chains to be used for Bayesian estimation.
verbose	NULL	0, 1, or 2	The verbose argument is passed to ctStanFit. Higher values print more information during model fit.
allInvModel	FALSE	TRUE / FALSE	Whether or not a model should be tested in which all (!) parameters are assumed to be invariant across primiary studies. If set TRUE, other specifications (e.g., speficied with the equalDrift argument) will be ignored. An all invariant model is also used by ctmaPower.
customPar	TRUE	TRUE / FALSE	If set TRUE some starting values usually used by ctSatFit will be used by CoTiMA specific settings. Not recommended to be used in combination with Bayesian estimation. It was introduced to improve handling of large values used in delta_ti. Setting it to FALSE and use scaleTime instead could be a better alternative if estimation problems will nevertheless occur.

ctmaFitList (plotting)			Informs the plot function that more than a single CoTiMA fit object should be plotted.
Argument	Default	Possible Values	Explanation
...	nothing	CoTiMA fit objects separated by commas	For example, ctmaFitList(object1, object2).

ctmaGetPub (-)			Retrieves publication and citation indices for authors from Google Scholar, which could be further processed with ctmaPub.
Argument	Default	Possible Values	Explanation

ctmaGetPub (-)			Retrieves publication and citation indices for authors from Google Scholar, which could be further processed with ctmaPub.
authorList	NULL	List of vectors with 2 elements	Contains information about authors' names and their Google Scholar https address (or their user ID), e.g., `list( c("Wilmar B.; Schaufeli", "https://scholar.google.de/citations-?hl=en&user=w1tHcj4AAAAJ"), c("Maureen; Dollard", "user=J6oH3rgAAAAJ") ) )`. Authors' surnames are separated from given names or initials by a semicolon!
flush	FALSE	TRUE / FALSE	Argument is handed over to scholar R package. If TRUE, the cache will be cleared and the data reloaded from Google Scholar. Google Scholar will limit the retrieval of information or even suspend it for a while if the cache is flushed too frequently.
yearsToExclude	NULL	(vector of) years to exclude	Recommended to leave as NULL. Years could be excluded later when using `ctmaPub`.

ctmaInit (EP*I*C-BiG-Power)			Fits a ctsem model to a list of primary studies prepared by `ctmaPrep`.
Argument	Default	Possible Values	Explanation
primaryStudies	NULL	list	A list created with ctmaPrep that containes all information (e.g., `empcovi`, `delta_ti`, `sampleSizei` etc.) relevant for ctmaInit and subsequent analyses.
activeDirectory	NULL	character string	Specifies the directory where required files are found and saved. Should end with "/". For example, `''/Users/GDC/CoTiMA/''`
activateRPB	FALSE	FALSE/ TRUE	Messages (warning, finished fitting) could be send to mobile phone if TRUE.
checkSingleStudy-Results	TRUE	TRUE / FALSE	If yes, displays estimates from single study ctsem models and waits for user input to continue. Useful to check estimates before they are saved.
digits	4	value > 0	Rounding used in output.
n.latent	NULL	value > 0	Number of latent variables.
n.manifest	0 (= n.latent)	value $\geq$ n.latent	Number of manifest variables.
lambda	NULL (= identity matrix)	n.latent × n.manifest matrix	Matrix with pattern of fixed (=1) or free (any string) loadings.
manifestVars	NULL (= 0)	n.manifest × n.manifest matrix	Matrix with error(co-)variances of manifest indicators.

ctmaInit (EP*IC*-BiG-Power)			Fits a ctsem model to a list of primary studies prepared by `ctmaPrep`.
`drift`	NULL (=all)	vector (!) of rowwise drift matrix elements	Labels for drift effects that should or should not be included. Have to be either of the type `V1toV2` or `0` for effects to be excluded, which is usually not recommended, e.g., `c`(''`V1toV1`'', ''`V2toV1`'', `0`, ''`V2toV2`'')
`doPar`	1	integer value > 0	Fits each model `doPar` times in parallel mode and returns best single model fit. In each of the parellel fit attempts, `coresToUse` is set to 1 to avoid conflicting processes. Probably does not work in Windows machines. The number of parallel processes is limited to all available threads (cores) minus 1, and if the value assigned to `doPar` is larger, multiple parallel fits are done sequentially.
`equalDrift`	NULL	vector of 0 or 2 or more drift effects	Labels for drift effects that should be set equal. Have to be of the type `V1toV2`, e.g., `c`(''`V2toV1`'', ''`V1toV2`''). Is useful, for example, to test if the sizes of cross effects are statistically different.
`indVarying`	FALSE	TRUE / FALSE	Specifies a random (manifest) intercept model. Works only if all primary studies have 3 or more waves and no missing values (i.e., variables) exist
`saveRawData`	list()	list(saveRawData-$studyNumbers, saveRawData$file-Name, saveRawData-$row.names, saveRawData-$col.names, saveRawData$sep, saveRawData$dec)	A list with required information to save generated pseudo raw data. Mighgt be usefull for methodological research questions. For example, `list("saveRawData$studyNumbers = c(1: 20), "saveRawData$fileName" = "pseudoRaw", "saveRawData$row.names" = FALSE, "saveRawData$col.names" = TRUE, "saveRawData$sep" = " ", "saveRawData$dec" = ".")`
`coresToUse`	1	value > 0 or < 0	The number of cores (threads) to be used for fitting. If a negative values is specified, the value is subtracted from available cores, else the value sets the number of cores to be used. Should usually be 1 on Windows OS.
`silentOverwrite`	FALSE	TRUE / FALSE	Whether or not to prompt user preventing undesired overwriting of existing single study fit files (requested via `saveSingleStudyModelFit`)
`saveSingleStudy-ModelFit`	c()	vector with filename followed by the numbers of studies for which the fit is saved	Save the fit of single study ctsem models (could save a lot of time afterwards if the fit is loaded, e.g. `saveSingleStudyModelFit=c("myModel", 1, 4, 5, 6:100)`

ctmaInit (EP*IC*-BiG-Power)			Fits a ctsem model to a list of primary studies prepared by `ctmaPrep`.
loadSingleStudy-ModelFit	c()	vector with filename followed by the numbers of studies for which the fit is saved	Load the fit of single study ctsem models, e.g. `loadSingleStudyModelFit=c("myModel", 1, 4, 5, 6:100)`. This is useful, e.g, if primary studies aree added to the pool of primary studies. Only the added studies will be fitted, the previously fitted models are loaded, and all is then stored in the resulting `ctmainit fit` object.
scaleTime	NULL (=FAlSE)	value > 0	Wether or not the time scale used for `delta_ti` should be changed.For example, scaleTime=1/12 could change the time scale from months to years. It is usually recommended to avoid `delta_ti` larger than 6. The default value FALSE is taken from the `CoTiMAStanctArgs` list, which could also be used to specify more stanct fitting parameters.
optimize	TRUE	TRUE / FALSE	The `optimize` argument is passed to ctStanFit. If FALSE, Bayesian estimations is used. The chosen sampler is conditional on the `nopriors` argument. Note that this works differently than the optimise argument of ctSatFit.
nopriors	TRUE	TRUE / FALSE	Consequences of TRUE or FALSE are conditional on the optimize argument. optimize=TRUE & nopriors=TRUE implies maximum likelihood estimation, optimize=TRUE & nopriors=FALSE implies maximum a posteriori estimation, optimize=FALSE & nopriors=TRUE implies Bayesian estimation using HMC (Hamiltonian Monte Carlos sampler), and optimize=FALSE & nopriors=FALSE implies Bayesian estimation using NUTS (No U-Turn Sampler).
finishsamples	NULL (=1000)	values > 0	The `finishsamples` argument is passed to ctStanFit. It specifies the number of samples to draw for final results computation. Larger (e.g., 10.000) values make results more exactly replicable. Larger values are recommended befor manuscripts are submitted. Very large values (e.g., 100.000) might be helpful if very small effects (e.g., 0.0002) result from estimation.

ctmaInit (EP*IC*-BiG-Power)			Fits a ctsem model to a list of primary studies prepared by `ctmaPrep`.
chains	NULL (=2)	values > 0	The `chains` argument is passed to ctStanFit and specifies the number of chains to be used for Bayesian estimation.
iter	NULL (=1000)	values > 0	The `iter` argument is passed to ctStanFit. It specifies the number of iterations used for Bayesian estimation, half of which will be devoted to warmup.
verbose	NULL	0, 1, or 2	The `verbose` argument is passed to ctStanFit. Higher values print more information during model fit.
customPar	TRUE	TRUE / FALSE	If set `TRUE` some starting values usually used by ctSatFit will be used by CoTiMA specific settings. Not recommended to be used in combination with Bayesian estimation. It was introduced to improve handling of large values used in `delta_ti`. Setting it to `FALSE` and use `scaleTime` instead could be a better alternative if estimation problems will nevertheless occur.

ctmaPower (EPIC-BiG-*Power*)			Informs the plot function that more than a single CoTiMA fit object should be plotted.
Argument	Default	Possible Values	Explanation
ctmaInitFit	NULL	CoTiMA fit object	Object to which all single ctsem fits of primary studies has been assigned to (i.e., what has been returned by `ctmaInit`)
activeDirectory	path used to create ctmaInit-Fit object	path to directory	Specifies the directory where required files are found and saved. Should end with "/".
statisticalPower	c()	vector	Vector of requested statistical power values, e.g. `c(.95, .80)`
failSafeN	NULL	value > 0	A sample size used to determine across which time intervals expected effects become not significant. If not specified, the avergae sample size of primary studies will be used.
failSafeP	NULL	value between 0 and 1	A p-value used to determine across which time intervals expected effects become not significant. If not specified .01 will be used.

ctmaPower (EPIC-BiG-*Power*)			Informs the plot function that more than a single CoTiMA fit object should be plotted.
timeRange	NULL	vector with 3 values	Specifies the time range across which statistical power etc will be computed. A vector with 3 values: sarting point, end point, step witdh, e.g. c(0, 50, 1). if not specified, c(0, 1.5*maxDelta, 1) will be used, with 1.5*maxDelta indicating that the end point is 1.5 times the longest time interval among primary studies.
useMBESS	FALSE	TRUE / FALSE	If TRUE, the MBESS package is used to calculate statistical power (slower). Otherwise, use the internal CoTiMA function (faster).
coresToUse	1	value > 0 or < 0	The number of cores (threads) to be used for fitting. If a negative values is specified, the value is subtracted from available cores, else the value sets the number of cores to be used. Should usually be 1 on Windows OS.
digits	4	value > 0	Rounding used in output.
indVarying	FALSE	TRUE / FALSE	Specifies a random (manifest) intercept model. Works only if all primary studies have 3 or more waves and no missing values (i.e., variables) exist
activateRPB	FALSE	FALSE/ TRUE	Messages (warning, finished fitting) could be send to mobile phone if TRUE.
silentOverwrite	FALSE	TRUE / FALSE	Whether or not to prompt user preventing undesired overwriting of existing fit files (requested via saveAllInvFit or saveAllInvWOSingFit)
loadAllInFit	c()	character sting	Load the fit of a CoTiMA model with all effects invaraint across primary studies, e.g. loadAllInvFit=c("myAllInvariantModel")
saveAllInFit	c()	character sting	Save the fit of a CoTiMA model with all effects invaraint across primary studies, e.g. saveAllInvFit=c("myAllInvariantModel")
loadAllInvWOSingFit	c()	character sting	not yet operational
saveAllInvWOSingFit	c()	character sting	not yet operational
skipScaling	TRUE	FALSE / TRUE	If FALSE, combined raw data are standardized again. Although pseudo raw data for each primary study have variance = 1.0, this is not the case if they are combined into the single data set that is used to compute the model with all effects being invariant. This is because variance is computed with denominator N - 1. Could be corrected by setting skipScaling to FALSE, but usually has little practical consequences.

ctmaPower (EPIC-BiG-*Power*)			Informs the plot function that more than a single CoTiMA fit object should be plotted.
useSampleFraction	NULL	value between 100 and 0	Analyze only a fraction of the overall sample. Could help speeding up debugging. Provided as percent (e.g., `useSampleFraction=30` uses 30% of the overall sample size)
optimize	TRUE	TRUE / FALSE	The `optimize` argument is passed to ctStanFit. If `FALSE,` Bayesian estimations is used. The chosen sampler is conditional on the `nopriors` argument. Note that this works differently than the optimise argument of ctSatFit.
nopriors	TRUE	TRUE / FALSE	Consequences of `TRUE` or `FALSE` are conditional on the optimize argument. optimize=`TRUE` & nopriors=`TRUE` implies maximum likelihood estimation, optimize=`TRUE` & nopriors=`FALSE` implies maximum a posteriori estimation, optimize=`FALSE` & nopriors=`TRUE` implies Bayesian estimation using HMC (Hamiltonian Monte Carlos sampler), and optimize=`FALSE` & nopriors=`FALSE` implies Bayesian estimation using NUTS (No U-Turn Sampler).
finishsamples	NULL (=1000)	values > 0	The `finishsamples` argument is passed to ctStanFit. It specifies the number of samples to draw for final results computation. Larger (e.g., 10.000) values make results more exactly replicable. Larger values are recommended befor manuscripts are submitted. Very large values (e.g., 100.000) might be helpful if very small effects (e.g., 0.0002) result from estimation.
iter	NULL (=1000)	values > 0	The `iter` argument is passed to ctStanFit. It specifies the number of iterations used for Bayesian estimation, half of which will be devoted to warmup.
chains	NULL (=2)	values > 0	The `chains` argument is passed to ctStanFit and specifies the number of chains to be used for Bayesian estimation.
verbose	NULL	0, 1, or 2	The `verbose` argument is passed to ctStanFit. Higher values print more information during model fit.

ctmaPower (EPIC-BiG-*Power*)			Informs the plot function that more than a single CoTiMA fit object should be plotted.
customPar	TRUE	TRUE / FALSE	If set `TRUE` some starting values usually used by ctSatFit will be used by CoTiMA specific settings. Not recommended to be used in combination with Bayesian estimation. It was introduced to improve handling of large values used in `delta_ti`. Setting it to `FALSE` and use `scaleTime` instead could be a better alternative if estimation problems will nevertheless occur.

ctmaPub (-)			Computes publication indices for the group of authors of a study (augments `ctmaGetPub`).
Argument	Default	Possible Values	Explanation
getPubObj	NULL	Object created with `ctmaPubGet`	Publication (and citation) information of authors.
selectedStudies	NULL	Vector of study numbers	Specifies the studies, for which the groups of authors' publication information should be computed.
yearsToExclude	NULL	(vector of) years to exclude	Years to be excluded from computations. For example, the current year might be excluded because publication infromations might not be very reliably. Early years (e.g., 1900-1960) might be excluded because they would cause invalid publications (sometimes this happens in Goofle Scholar).
targetYear	NULL (=publication year)	a positive value	If left NULL, all publications before the year of the authors' publication count.
recency	5	a positive value	`ctmaPub` computes 2 indices. For the first one (NEPP), all years before `targetYear` count. For the second one (NEPPrecency), the years between `targetYear` and `targetYear - recency` count.
indFUN	''sum''	any of: `"mean"`, `"sum"`, `"max"`, `"min"`, `"var"`	Specifies the function used to aggregate an *ind*ividual author's publication numbers, e.g., `sum` (recommended) computes the sum of an author's publication before `targetYear`, and `var` computes the variance of the number of publications for an author's first year of publication to `targetYear`.

ctmaPub (-)			Computes publication indices for the group of authors of a study (augments `ctmaGetPub`).
colFUN	''mean''	any of: `"mean"`, `"sum"`, `"max"`, `"min"`, `"var"`	Specifies the function used to aggregate a group of authors (*col*lective) publication numbers, e.g., `mean` computes the mean of all authors' publication scores (created with `indFUN`, e.g., the sums) before `targetYear`, and `max` takes largest of all authors' publication scores (created with `indFUN`, e.g., the sums) to `targetYear`.
addAsMod	FALSE	FALSE / TRUE	Currently disabled.

ctmaPrep (EPIC-BiG-Power)			Combines information of primary studies into a list object and returns this list.
Argument	Default	Possible Values	Explanation
selectedStudies	NULL	vector with integers	Vector of primary study numbers (numeric values with no leading 0; e.g., '2' but not '02')

ctmaPrep (EPIC-BiG-Power)			Combines information of primary studies into a list object and returns this list.
excludedElements	NULL	vector with integers	Could be used to exclude some predefined objects from the results reported. Note that some predefined objects are strongly defined; they have to be used in a special way because they are actually used in subsequent analyses. Some other objects could be used at the researcher's convenience (information is just collected). Strongly predefined objects are `delta_t` (should be of the type `c(NA, NA)` in cases when raw data are provided, with the number of NAs corresponding to the number of time intervals), `sampleSize` (single number), `pairwiseN` (matrix of pairwise N; could be used if correlation matrix is based on pairwise N), `empcov` (correlation matrix), `moderator` (vector of numbers; could be continuous or categorical), `alphas` (vector of reliability estimates of the variables of a primary study), `startValues` (vector of start values), `rawData` (information about file name and structure of raw data), `empMeans` (means for variables; usually 0), and `empVars` (variances for variables; usually 1.0). Weakly predefined objects are `studyNumber` (intended as a special number used for the outputs of subsequently fitted CoTiMA models), `source` (intended as vector of authors' names and publication year), `ageM` value intended for indicating the mean age of participants in a primary study, `malePercen'` (intended as value indicating the percentage of male participants in a primary study), `occupation` (intended as vector of character strings representing the occupations of participants in a primary study), `country` (intended as single character string representing the country in which a primary study was conducted), and `targetVariables` (intended as vector of character strings representing information about the variables used).

ctmaPrep (EPIC-BiG-Power)			Combines information of primary studies into a list object and returns this list.
addElements	NULL	vector of character strings	Could be use to add user-defined objects that are handled as the weakly predefined objects. The major purpose is to collect information a researcher regards as important, e.g. `c("Important", "Interesting")`
digits	4	value ≥ 0	Rounding used in output.
moderatorLabels	NULL	vector of character strings	Vector of Names used to label moderators in the output e.g., `c("Mod1", "Control")`
moderatorValues	NULL	list of vectors	List of vector of Names (assignments) used to label moderators in the output e.g., `list(c(1"=Emotional Exhaustion", "2=Exhaustion"), "continuous")`

The `plot` function, which is described next, works slightly different than other CoTiMA functions. Like all other CoTiMA functions, some arguments could be used as always. However, in addition, it is important to note that several plotting parameters ('fitAddSpecs') have to be assigned to the CoTiMA fit-object before plotting it, rather than using plotting paramters as arguments to the `plot` function (e.g., `CoTiMAInitFitObject$xMax <- 200`). This is because the arguments have different effects conditional on the type of fit-object. The number of plotting parameters that can be changed in this way is still limited; we are working on extensions. Further, user-defined plotting parameters differ for fit objects created with `ctmaBiG` versus `ctmaInit` and `ctmaFit`). Finally, if problems with `plot` are encountered, we recommend trying `ctmaPlot` instead.

plot/ctmaPlot			Generates plots.
Argument	Default	Possible Values	Explanation
ctmaFitObject	NULL	vector with integers	A CoTiMA fit-object created by `ctmaInit`, `cmtaFit`, or `ctmaBig`.
activeDirectory	path used to create ctmaInit-Fit object	path to directory	Specifies the directory where required files are found and saved. Should end with "/".
saveFilePrefix	"ctmaPlot"	vector of character strings	Labels for the generated plot, which might be automatically augmented by further information (e.g., "ctmaplot V1toV1,png")
activateRPB	FALSE	FALSE/ TRUE	Messages (warning, finished fitting) could be send to mobile phone if `TRUE`.
plotCrossEffects	TRUE	TRUE/ FALSE	Affects plotting of `ctmaInit` or `ctmaFit` fit-objects only. Plotting of discrete time cross effects can be suspended.
plotAutoEffects	TRUE	TRUE/ FALSE	Affects plotting of `ctmaInit` or `ctmaFit` fit-objects only. Plotting of discrete time auto effects can be suspended.

plot/ctmaPlot			Generates plots.
timeUnit	`"timeUnit (not specified)"`	vector of character strings	Affects plotting of `ctmaInit` or `ctmaFit` fit-objects only. Label used for the x-axis of discrete time plots.
timeRange	1 to 1.5 times the longest interval used in primary studies	vector with 3 values: c(xMin, xMax, stepwidth)	Affects plotting of `ctmaInit` or `ctmaFit` fit-objects only. The range across which discrete time effects are plotted, e.g., `c(10,. 20, .01)` would plot effects from 10 units of time to 20 using steps of .01. Note that a stepwidth < 1 could be specified to obtain more fine-grained figures.
yLimitsForEffects	values slightly exceeding min and max empirical effect sizes	vector with 2 values: c(yMin, yMax)	Affects plotting of `ctmaInit` or `ctmaFit` fit-objects only. The min and max values for the y-axis. Setting explicit values couod be better than relying on the automatically determined range, for example, to ensure identical y-axis across a larger set of plots.
mod.values	`c(-2-1, 0, 1, 2)`	vector with numbers	Affects plotting of `ctmaFit` fit-objects only. The moderator values for which plots of continuous moderators should be generated. Correspondes to the standard deviations below and above the mean value if the continuous moderator has was standardized with `scaleMod=TRUE`. Does not affect plotting of categorical moderators.
aggregateLabel	`""` (=nothing)	character(s)	Affects plotting of `ctmaFit` fit-objects only. Symbol to be attached to the discrete time plot of a `ctmaFit` fit-object. In the case of `ctmaInit` fit-objects, each study is usually idetified in the plot with a dot inside which the study number is shown. In the case of a `ctmaFit` fit-object with aggregated effects, one could use a symbol, e.g. `aggregateLabel=''Σ''`.
xLabels	NULL	vector with numbers	Affects plotting of `ctmaFit` fit-objects only. The numbers indicating the time intervals on the x-axis are usually determined automatically. The could also directly specified, and the values provided are equally distributed across the time range used to plot the discrete time effects, e.g., `c(1, 3, 5, 7, 9)`

Argument	Default	Possible Values	Explanation
fitAddSpecs (ctmaBias fit-objects)			
CoTiMAFit$xMin	0	value $\geq$ 0	Internally, the x axis ranges from 0 to 300 (the values shown in the plot are irrelevant). Setting xMin > 0 creates a plot where the left part is left out. For example, if one wants to leave out the first quarter (i.e. 0 to 300/4 = 0 to 75) you could set xMin to 75. If one wants extra space on the right hand side of the plot, one could lift xMax to values larger than 300, e.g.,c `CoTiMAFit$xMax <- 400`.
CoTiMAFit$xMax	300	value $\geq$ 0	see above

fitAddSpecs (ctmaInit & ctmaFit fit-objects)			Generates discrete time effect plots.
Argument	Default	Possible Values	Explanation
CoTiMAFit$col	"grey" & "black" for `ctmaInit` & `ctmaFit` fit-objects, respectively.	R-type color code	Sepcifies the color of the curve showing the discrete time affects across time, e.g., `CoTiMAFit$col <- "red"`
CoTiMAFit$lwd	1.5 & 2.5 for `ctmaInit` & `ctmaFit` fit-objects, respectively.	value $\geq$ 0	Sepcifies the line width of the curve showing the discrete time affects across time, e.g., `CoTiMAFit$lwd <- 4`
CoTiMAFit$lty	1 & 2 for `ctmaInit` & `ctmaFit` fit-objects, respectively.	R-type integer value $\geq$ 0	Sepcifies the line type of the curve showing the discrete time affects across time, e.g., `CoTiMAFit$lty <- 1`, with, e.g., 1 = solid, 2 = dashed, and 3 = dotted.
CoTiMAFit$xMin	min of `timeRange`	integer values	Limits the displayed range of the discrete time effects plotted. Should only be used in combination with the `xLabels` argument. Overrides other setting if multiple fit-objects are supplied. This is experimental; it is recommended to set `timeRange` instead.

fitAddSpecs (ctmaInit & ctmaFit fit-objects)			Generates discrete time effect plots.
CoTiMAFit$xMax	max of timeRange	integer values	Limits the displayed range of the discrete time effects plotted. Should only be used in combination with the xLabels argument. Overrides other setting if multiple fit-objects are supplied. This is experimental; it is recommended to set timeRange instead.
CoTiMAFit$dot.type	"b"	R-type characters	Type of the plot. Use "p" for points, "l" for lines, "b" for both, "c" for the lines part alone of "b" etc.
... to be continued			