

**Response Letter to the Associate Editor's Comments for RJournal 2021-140:
metapack: An R Package for Bayesian Meta-Analysis and Network
Meta-Analysis with a Unified Formula Interface**

Comments to Author: Both reviews are fairly short, especially that of Reviewer 1. Both reviewers are fairly positive and recommends "Minor Revision", but the issues described by reviewer 3 are not only minor, and I believe the authors should carefully consider those.

Response: We appreciate the helpful comments. We have tried our best to address issues raised by both reviewers. The reproducibility issue has been taken very seriously and since been resolved. The modified parts in the main manuscript of the revision have been marked in a different color (**magenta**) for enhanced visibility.

Response Letter to Reviewer #1's Comments for RJournal 2021-140: metapack: An R Package for Bayesian Meta-Analysis and Network Meta-Analysis with a Unified Formula Interface

In this paper Lim et al. present a new package that implements functionalities for Bayesian and network meta-analysis that are based on two previous works from the same group Hui Yao et al. Journal of American Statistical Association 2015 and Li et al. Statistics in Medicine 2021.

Meta-analysis is gaining a lot of attention from the research community, especially in the biomedical field, and I think that the proposed work provides functionalities that, although the scope can be limited, are enough relevant to be published in R journal.

Nevertheless, I think that the manuscript would be improved if

1. *They should provide more details about what they call “standard data structure for meta-analysis”, either describing it in the manuscript and package documentation (preferred) of references where this data format is presented. Depending on the research context, the input data for meta-analysis studies may be different and it is not clear for example what are DesignM1 and DesignM2 in the data structure.*

Response: Our understanding is that, in the case of aggregate meta-analyses, the response consists of trial- or arm-level effect sizes and their corresponding dispersion measures (i.e., standard errors). In some occasions, the standard errors are not reported but they are recoverable from respective confidence intervals in most cases. For noncontinuous data, there are ways to back-calculate effect size (e.g., odds ratio, relative risk, and proportion) and its standard error (Hu et al., 2020). We consider these to be the minimum requirement for typical meta-analytic data, although arm-level sample sizes are additionally required, which are conventionally always reported and available.

DesignM1 is the fixed-effect matrix, so if meta-regression is considered, it will have to be specified. On the other hand, DesignM2 is a context-specific second design matrix. For example, in the meta-analysis setting considered for metapack, this corresponds to the random-effect design matrix, whereas in the network meta-analysis setting, it contains the covariates used for variance modeling.

To address the possible confusion, we have unified the notation for DesignM2 to \mathbf{W} with suitable subscripts. Furthermore, we refrain from using “standard” in our revised manuscript to describe the required data structure for metapack.

2. *Although I imagine that detailed descriptions about the interpretation of results of each method can be found in the original papers, the manuscript should provide a brief explanation of results and how to interpret them in the examples rather than simply provide the output of the analysis.*

Response: Thank you for your comment. We have added brief explanations of the results in both the meta-analysis and network meta-analysis cases. In page 11, see the paragraph

“The suffixed $_j$ where j can be 1, 2, or 3 corresponds to the response endpoint. Since `scale_x = TRUE` is equivalent to `scale(<var>, center=TRUE, scale=TRUE)`, the covariates have been centered, which affects the interpretation of the intercepts. This allows us to interpret $(\text{Intercept}(1-2nd)_1)=-42.8675$ as the statin effect in the first-line studies, where `2nd` represents the indicator variable for second-line studies evaluating to one if second-line and zero otherwise. On the other hand, the coefficient estimate -12.1435 for `treat*(1-2nd)_1` is the Statin+Ezetimibe effect, compared to administering statin alone. For the second-line studies where patients had already been on statin, $(\text{Intercept})*2nd_1=-3.2219$ came out insignificant, according to the 95% HPD interval, as anticipated because the treatment for this group was merely a continuation of taking statin. The coefficient estimate -20.0843 for `treat*2nd_1` shows that ezetimibe on top of statin has a greater cholesterol-lowering effect than statin alone.”*

In page 14, see the paragraph

“We observe that with covariate adjustment, all active treatments (A, AE, E, L, LE, P, PE, R, S, SE) reduce triglyceride (TG) more effectively than the placebo (PBO), although E and P have 95% HPD intervals including zero.”

3. Define the meaning of IPD models in the manuscript.

Response: The current version of metapack does not include models for IPD data, so all mention of IPD models has been deleted.

Response Letter to Reviewer #3's Comments for RJournal 2021-140: metapack: An R Package for Bayesian Meta-Analysis and Network Meta-Analysis with a Unified Formula Interface

This paper presents metapack R package which implements univariate and multivariate meta-analysis and univariate network meta-analysis using Bayesian inference. One of the important contributions of the package is the introduction of a unified formula structure that flexibly represents the type of responses and the number of treatments.

1. *In the Introduction Section you reviewed some meta-analysis R packages and it is not clear why you have selected them. There are much more R packages to do meta-analysis than the ones you have mentioned. If there is a reason for your selected packages you should mention it in the paper. Maybe you have selected them because they are the most cited R packages to do meta-analysis or they are more similar to your package than others or for other reasons. Also can be useful as a reference to Meta-Analysis R packages to mention "CRAN Task View: Meta-Analysis" maintained by Michael Dewey which is up to date. If you have not checked this Task View you should do it and extend at least the Bayesian meta-analysis R packages you have mentioned. For example why don't you include MetaStan?*

Response: Thank you for these useful comments and suggestions. There are two main reasons behind the selection of R packages mentioned in the introduction of our manuscript. First, there had to be a reasonable number of both frequentist and Bayesian meta-analysis packages, the top-ranking of which are mostly frequentist. Therefore, the frequentist packages like `metafor` or `rmeta` simultaneously represent the most cited R packages. This appears to be consistent with the "Fitting the model" section of CRAN Task View: Meta-Analysis. Following your suggestion, we have cited `MetaStan` in the introduction as well.

2. *(Meta-analytic data) "Figure 1 represents a typical data set for (network) meta-analysis" should say Table. 1.*

Response: Thank you for catching this error. It has been fixed.

3. *The data structure description is not clear enough. In Table 1 you show the structure for network meta-analysis, but I think it will be useful if you include the example structure with more details, like number of treatments, number of studies, possible values for k etc.*

Response: Thank you for pointing out this. In the revision, we have elaborated on the data structure both in the caption of Table 1 and the data structure section of the manuscript.

4. *This is how the data should be structured to use metapack?*

Response: Yes, this is how the data should be formatted for metapack.

5. *The Table should include a column for StudyID if this is a genial data structure.*

Response: StudyID is equivalent to Trial. We have added a sentence explaining the equivalence.

6. *In the Table description you type std.Dev and in the Table the column is named SD (should be the same).*

Response: Thank you for catching this. It has since been revised.

7. *You have included two columns DesignM1 and DesignM2, will be useful to describe x_{tk} and w_{tk} because it is the first time some notation appears in the paper and you should mention at least what are the subindices.*

Response: Following your suggestion in Comment 9 below, the section order has been changed. Thus, it should be clear what \mathbf{x}_{kt} and \mathbf{w}_{kt} mean. However, for clarification, we have also added the generic notation, \mathbf{x}_{kt} and \mathbf{w}_{kt} , in the top row to indicate what the subscripts mean.

8. *You mentioned Table 1 is for network meta-analysis then you should comment what is the main difference in the data structure for a classic meta-analysis.*

Response: We have added a sentence explaining the difference between meta-analysis and network meta-analysis. It reads

“Here, meta-analysis refers to when trials included have specifically two treatments (i.e., $t = 0, 1$ for all k), and the treatments are compared head to head. On the other hand, network meta-analysis includes more than two treatments, where each trial can have a different set of treatments, allowing indirect comparison between treatments that are not compared head to head. The data structure is unchanged for network meta-analysis except that **Treat** can have more than two unique values.”

9. *(Considered Models) I think you can improve the paper if you change the section order. Will be good to introduce notation before to talk about Meta-analysis data structure and the basic implementation of metapack. Maybe after introduction you can continue with Methodology background (Considered Models) and before meta-analysis model subsection you can explain the data structure and how to use the formula.*

Response: The order has been switched and the manuscript has been updated accordingly.

10. *You should include a table with a summary of functions in the metapack package and the data you have included with a description of them.*

Response: This is a nice suggestion. A table has been added to the Basic implementation of metapack section, enumerating all functions and data sets provided by the metapack package, as well as their descriptions.

11. *In Equation 1 is not clear the notation for ‘y’ and ‘epsilon’ in univariate and multivariate cases (look the same).*

Response: The following has been included:

Furthermore, for readability, note that boldface lowercase Latin letters are vectors, while uppercase Latin letters are matrices. Boldface Greek letters can either be vectors or matrices and will be defined contextually.

12. *The first example code is on page 5 but the data structure you presented before is not what you need to run this model. You need one column for each variable, you should clarify the data structure section and mention the data structure you need to run the models.*

Response: We have clarified this in the caption of Table 1.

13. *(**Demonstration with real data**) Will be useful when you describe the data if you show the data structure and also when you describe some of the variables for cholesterol include the name of them in the data set in parenthesis.*

Response: The variable names have been included in parentheses when describing the data set. The introduction paragraph for the cholesterol data set now reads

*“**metapack** includes a data set, **cholesterol**, which consists of 26 double-blind, randomized, active, or placebo-controlled clinical trials on patients with primary hypercholesterolemia sponsored by Merck & Co., Inc., Kenilworth, NJ, USA (Yao et al., 2015). The data set can be loaded by running `data("cholesterol")`. The **cholesterol** data set has three endpoints: low density lipoprotein cholesterol (**pldlc**), high density lipoprotein cholesterol (**phdlc**), and triglycerides (**ptg**). The percent change from the baseline in the endpoints, variables prefixed by **p-**, are the aggregate responses, followed by the corresponding standard deviations prefixed by **sd-**.”*

In addition, for each data set included in the package, a table has been added, which lists variable names and describes what each variable means. Please refer to Table 3 and Table 4.

14. *In all the examples you are using `set.seed()` but I’m curious about it because reproducibility is not trivial if you are doing MCMC. If you are using `set.seed` for reproducibility in this case is not working, I’ve run some of your examples in the paper and I didn’t get the same results. How you deal with reproducibility?*

Response: Thank you for the comment. It helped us identify the idiosyncrasies of the RcppArmadillo package that had great implications for our package. In its early days, our **metapack** package was programmed entirely using R’s random number generators (RNGs). At one point, as we were trying to figure out why one of our tests was failing, we learned that RcppArmadillo defaults to R’s RNGs. However, this was only for uniform and normal, which are `arma::randu` and `arma::randn`, respectively, as explained by Dr. Dirk

Eddelbuettel, who is the maintainer of the RcppArmadillo package. Another little known feature of RcppArmadillo is that samplers other than normal and uniform, including the χ^2 sampler, `arma::chi2rnd`, use different seeds.

From what we've observed, R's `set.seed` does set the seed across different platforms, but the pseudo-random sequence is operating system-dependent. The following test code (`test.cpp`) has been run on macOS Monterey 12.3.1 and Debian 4.19.160-2 with RcppArmadillo version 0.11.0.0.0 (latest):

```
#include <cmath>
#include <RcppArmadillo.h>
#include <Rmath.h>
#include <Rdefines.h>
// [[Rcpp::depends(RcppArmadillo)]]

// [[Rcpp::export]]
arma::mat test(int &n) {
    double df = 10.0;
    return arma::chi2rnd(df, n, n);
}
```

On my Mac,

```
library(Rcpp)
library(RcppArmadillo)
sourceCpp("test.cpp")
set.seed(1111)
test(3)
```

yields

	[,1]	[,2]	[,3]
[1,]	5.887224	17.05700	13.624775
[2,]	17.508067	13.30482	8.697143
[3,]	8.741382	12.24704	13.127976

whereas Debian yields

	[,1]	[,2]	[,3]
[1,]	8.799888	2.989588	10.063611
[2,]	6.388290	17.797957	10.254374
[3,]	6.100068	10.639533	7.728846

We have not been able to get to the bottom of this issue and how Armadillo is getting its seed for other samplers, but for now, we have coded the Wishart sampler ourselves

and verified that our MCMC results are reproducible across all three major operating systems (Windows, macOS, and Debian) with R's `set.seed`. The modified package will be submitted to CRAN upon our revised manuscript's resubmission.

15. (**Package**) *You use dots in all but `bmeta_analyze` function names and based on tidyverse style guide is not recommended. Most of them are ok because they are S3 objects but for example 'model.comp', 'bayes.parobs' and 'bayes.nmr' are not S3 objects.*

Response: Dots in function names will be changed to underscores in our next package update.

16. *Maybe you should include an argument for reproducibility or explain how you get reproducible results using MCMC.*

Response: As explained previously, following our scheduled update, the output will be fully reproducible through R's `set.seed`.

References

- Hu, D., Wang, C., and O'Connor, A. M. (2020). A method of back-calculating the log odds ratio and standard error of the log odds ratio from the reported group-level risk of disease. *PLOS ONE*, 15(3):1–8.
- Yao, H., Kim, S., Chen, M.-H., Ibrahim, J. G., Shah, A. K., and Lin, J. (2015). Bayesian inference for multivariate meta-regression with a partially observed within-study sample covariance matrix. *Journal of the American Statistical Association*, 110(510):528–544.