Supplemental Material: Bayesian Estimation and Comparison of Idiographic Network Models

Björn S. Siepe[1] and Daniel W. Heck[1]

[1]University of Marburg

Author Note

Björn S. Siepe https://orcid.org/0000-0002-9558-4648 Daniel W. Heck https://orcid.org/0000-0002-6302-9252

Correspondence concerning this article should be addressed to Björn S. Siepe, Department of Psychological Methods, University of Marburg, Gutenbergstraße 18, Marburg, Germany.

E-mail: bjoern.siepe@uni-marburg.de

Simulation Study 1

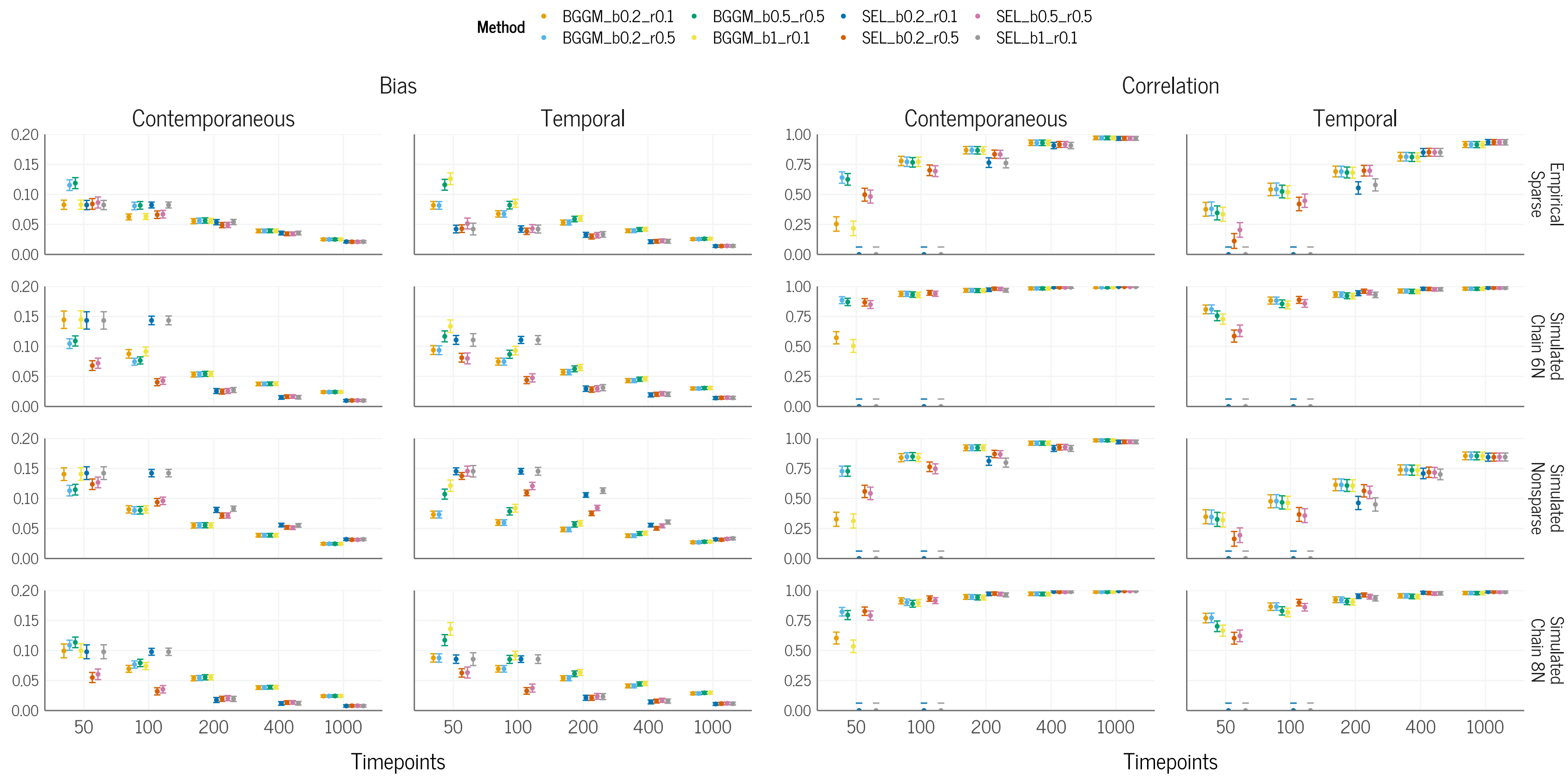
In this section, we present additional analyses for the first simulation study of the paper. Specifically, we show the performance of both *BGGM* and *graphicalVAR* across a range of hyperparameters, and compare them when edges should not be thresholded. We also present an analysis of the influence of different Credible Interval widths on the sensitivity and specificity of thresholding with *BGGM*.

BGGM Evaluation

In plot 1, we show bias and correlation with true edges of Bayesian gVAR for various prior hyperparameters. In general, narrower priors seem to perform better than wider priors. However, thresholding with a narrow prior of $s_\beta = 0.2$ and $s_\rho = 0.1$ or $s_\beta = 1$ and $s_\rho = 0.1$ does not work well, especially when using correlations with true values as performance metric. Likely, resulting networks are mostly empty due to estimates being pushed towards zero, especially with a narrow prior on the contemporaneous network. In the sparse data-generating processes, thresholding with $s_\beta = 0.2$ and $s_\rho = 0.5$ performed best with regards to bias, while non-thresholding performed better in the nonsparse data-generating process. For correlations with true parameters, non-thresholded estimation with $s_\beta = 0.2$ and $s_\rho = 0.5$ generally seemed to work best, although differences with other hyperparameters may be due to sampling variability as indicated by the error bars.

Figure 1

Performance of BGGM with Different Priors.



Note. Bias and correlation with true edges for different simulation conditions for the contemporaneous and temporal network (columns) and the data-generating processes (rows) for different priors of BGGM. 'BGGM' indicates non-thresholding, 'SEL' indicates thresholding. The value after 'b' indicates the prior standard deviation s_β for the temporal network. The value after 'r' indicates the prior standard deviation s_ρ for the contemporaneous network. Vertical bars indicate $1.96 \times SE$.

graphicalVAR Evaluation

In Plot 2, we show bias and correlation with true edges for LASSO gVAR for different settings of *graphicalVAR*. Specifically, we varied the choice of the EBIC hyperparameter γ and using regularization, indicated by using the default values for λ, or not performing regularization. Regarding bias, regularized estimation performs better in small sample sizes, whereas non-regularized estimation performs equally well or better in large sample sizes. The picture is not so clear for correlation with true parameters, where nonregularized estimation is often slightly or substantially better at smaller sample sizes for three out of four of the contemporaneous networks, and two out of four of the temporal networks. Correlations with true parameters tend to be relatively low for regularized estimation in small sample sizes, which can likely be explained by many coefficients being set to zero by LASSO.

Non-Thresholded Evaluation

In 3, we compare non-thresholded estimation with no edges being set to zero in the Bayesian gVAR framework in *BGGM* with non-regularized estimation in *graphicalVAR*. We see that a narrow prior in *BGGM* performs as good or better as unregularized estimation in *graphicalVAR* and the wider prior in *BGGM*.

Different Credible Intervals

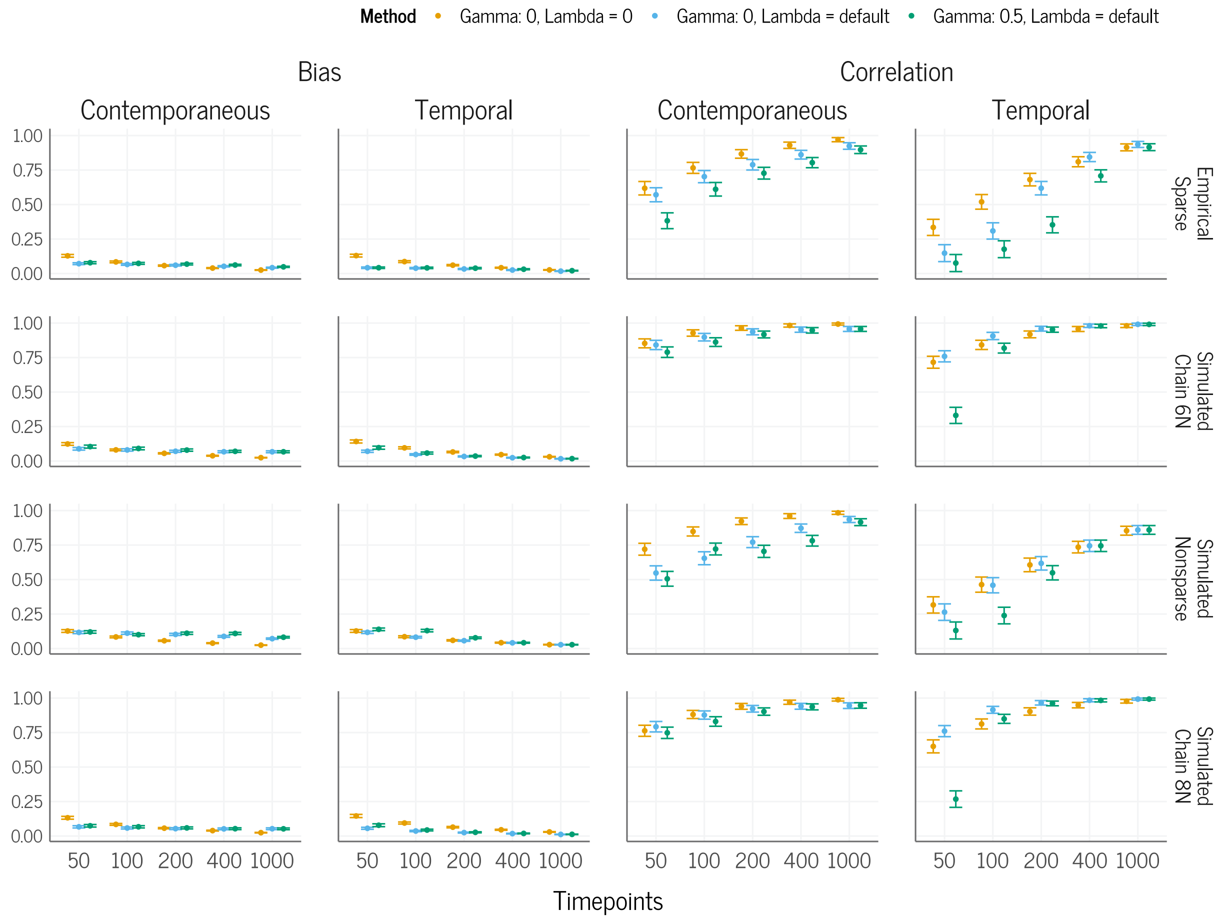
In plot 4, we show specificity and sensitivity of the Bayesian gVAR with different credible interval (CI) widths used for thresholding. As expected, sensitivity is considerably higher when using a more narrow CI width of .90 compared to .99, which is especially pronounced in smaller sample sizes. As a trade off, specificity is smaller for the narrower CIs. Notably, this does not seem to be the case for the temporal network with smaller sample sizese, where specificity is comparable across all CI widths.

Simulation Study 2

In this section, we present additional analyses for the second simulation study. We show an example of the change matrices that were used to manipulate the data-generating

Figure 2

Performance of graphicalVAR with different settings.



Note. Bias and correlation with true edges for different simulation conditions for the contemporaneous and temporal network (columns) and the data-generating processes (rows) for different settings of *graphicalVAR*. γ denotes the EBIC hyperparameter, λ denotes the choice of regularization parameters. Vertical bars indicate $1.96 \times SE$.

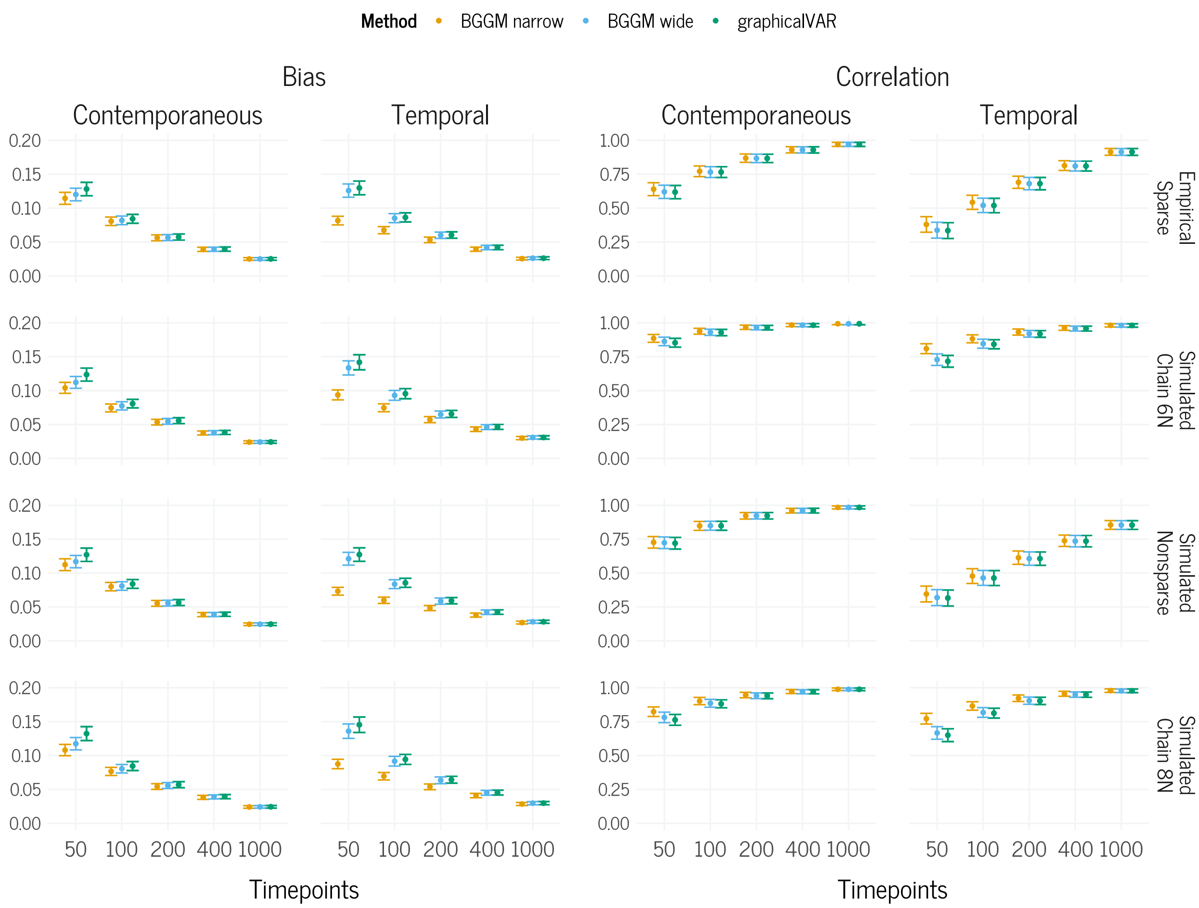
processes to create differences of various strengths. Further, we show the distribution of test values under the Null.

Change Matrices

We did not use pre-determined change matrices, as we occasionally ran into issues with non-semi-positive definite precision matrices. Rather, we iteratively drew change matrices, added them to the precision matrix, and then checked for semi-positive definiteness. The approach is described in detail in the R code at (https://osf.io/9byaj/).

Figure 3

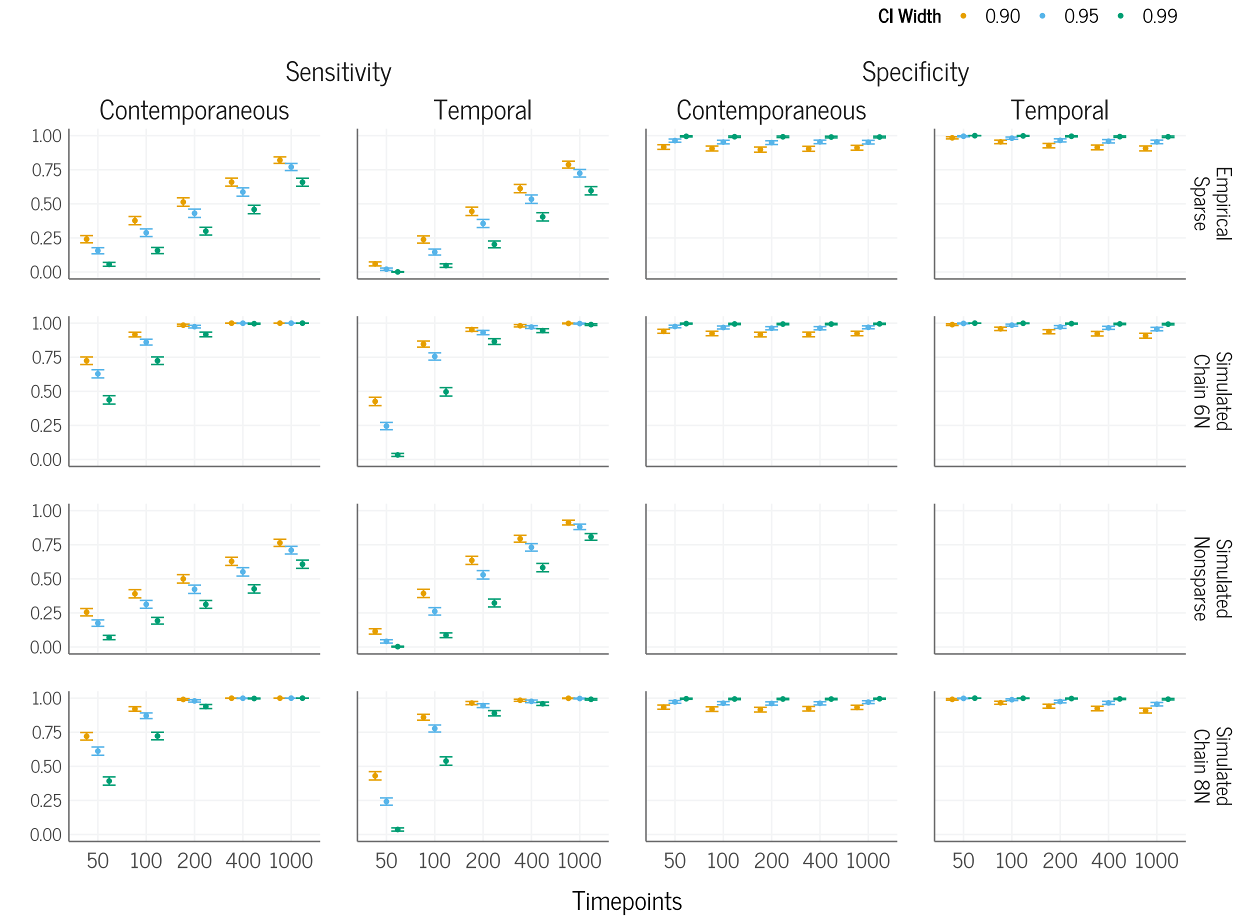
Comparison of BGGM and graphicalVAR withouth thresholding.



Note. Bias and correlation with true edges for different simulation conditions for the contemporaneous and temporal network (columns) and the data-generating processes (rows). Different estimation methods shown in different colors. Both Bayesian and LASSO gVAR without thresholding. Vertical bars indicate $1.96 \times SE$. The narrow prior has hyperparameters $s_\rho = 0.3$ and $s_\beta = 0.2$, the wide prior has hyperparameters $s_\beta = 1$ and $s_\rho = 0.5$.

Figure 4

Performance of BGGM with Different Credible Intervals.



Note. Sensitivity and Specificity for different interval widths for *BGGM* gVAR using a narrow prior. Specificity not defined for the simulated nonsparse graph, as all edges are nonzero.

To illustrate this approach, we show the change matrix for the 'constant 0.15' condition for the simulated nonsparse graph. First, we see the change matrix that was added to the temporal network:

$$\begin{bmatrix} 0.15 & -0.15 & -0.15 & -0.15 & -0.15 & 0.15 \\ -0.15 & -0.15 & 0.15 & -0.15 & -0.15 & 0.15 \\ 0.15 & -0.15 & -0.15 & -0.15 & -0.15 & -0.15 \\ 0.15 & -0.15 & 0.15 & -0.15 & -0.15 & -0.15 \\ -0.15 & -0.15 & 0.15 & 0.15 & 0.15 & -0.15 \\ -0.15 & 0.15 & 0.15 & -0.15 & 0.15 & -0.15 \end{bmatrix}$$

Second, we see the change matrix that was added to the precision matrix Θ:

$$\begin{bmatrix} 0 & 0.233 & -0.157 & 0.155 & 0.152 & -0.152 \\ 0.233 & 0 & -0.136 & 0.15 & 0.152 & 0.15 \\ -0.157 & -0.136 & 0 & 0.151 & 0.152 & -0.149 \\ 0.155 & 0.15 & 0.151 & 0 & 0.149 & 0.108 \\ 0.152 & 0.152 & 0.152 & 0.149 & 0 & 0.138 \\ -0.152 & 0.15 & -0.149 & 0.108 & 0.138 & 0 \end{bmatrix}$$

It is scaled to the diagonal elements of the precision matrix to achieve the desired change in the partial correlations.

Distribution under the Null

The following plots show the distribution of test values under the null when using the Frobenius norm, with the narrow prior of $s_\rho = 0.3$ and $s_\beta = 0.2$ in Figure 5 the wide prior $s_\beta = 1$ and $s_\rho = 0.5$ in Figure 6. For the narrow prior, test values were left-skewed, especially for the temporal network. This can likely be explained by the fact that a narrow prior draws all estimates towards zero, thereby making networks more similar across different data-generating processes. This tendency is therefore less pronounced for the wider prior, where sampling distributions of the test value are more uniform.

Empirical Example

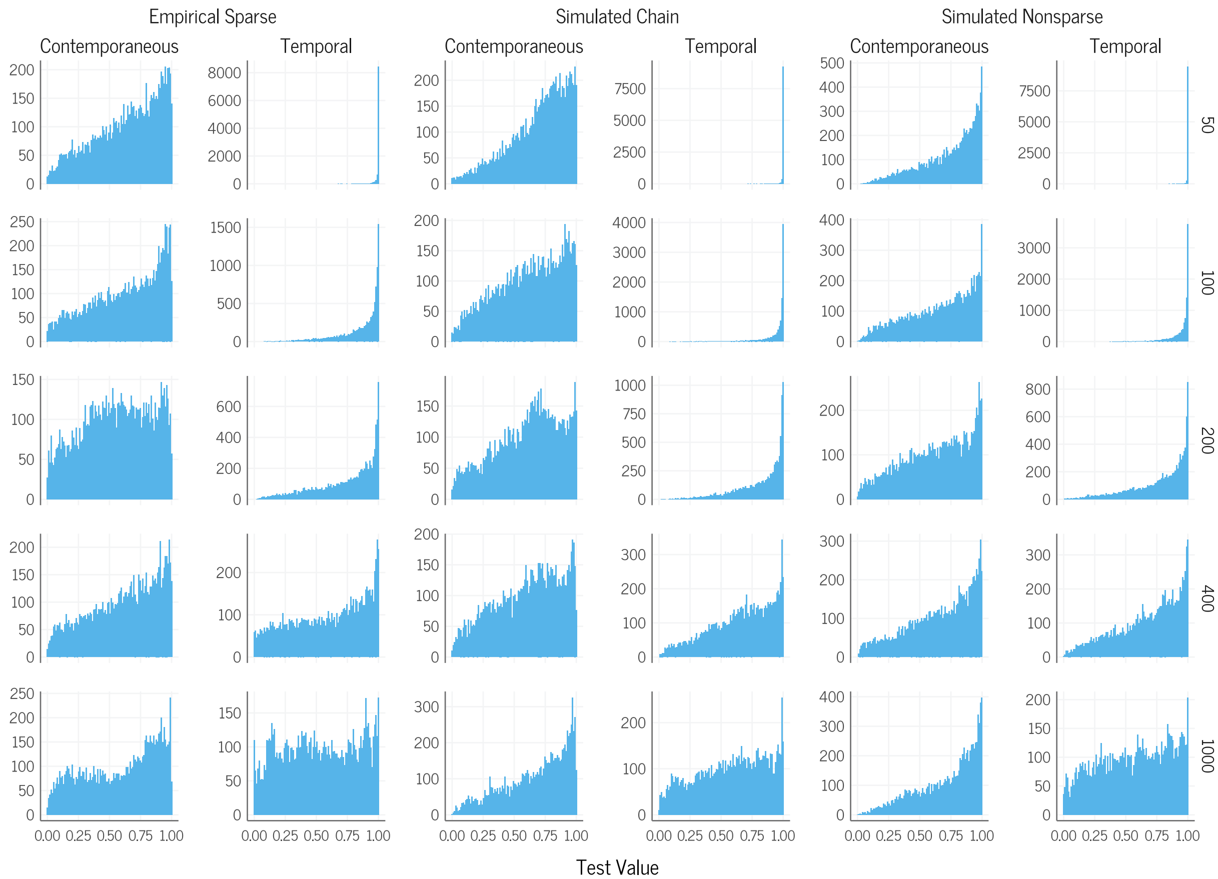
In this section, we present additional analyses for the empirical example. We show multiple sampling diagnostics and the prior sensitivity of the test in the example data.

Sampling Diagnostics

Here, we report sampling diagnostics for the 40 models in the empirical example, focussing on the prior combination we used to present our results ($s_\rho = 0.25$ and $s_\beta = 0.5$). These can either be obtained by using the convergence-function in *BGGM* for trace and autocorrelation plots, or by using the ess_gvar-function in the *tsnet* to compute the effective sample size. An overview over sampling diagnostics for MCMC sampling is given in Roy (2020).

Figure 5

Distribution of Test Values under the Null for the Narrow Prior.



Note. Using Frobenius norm only. X-axis displays different data-generating processes, y-axis dipslays different sample sizes.

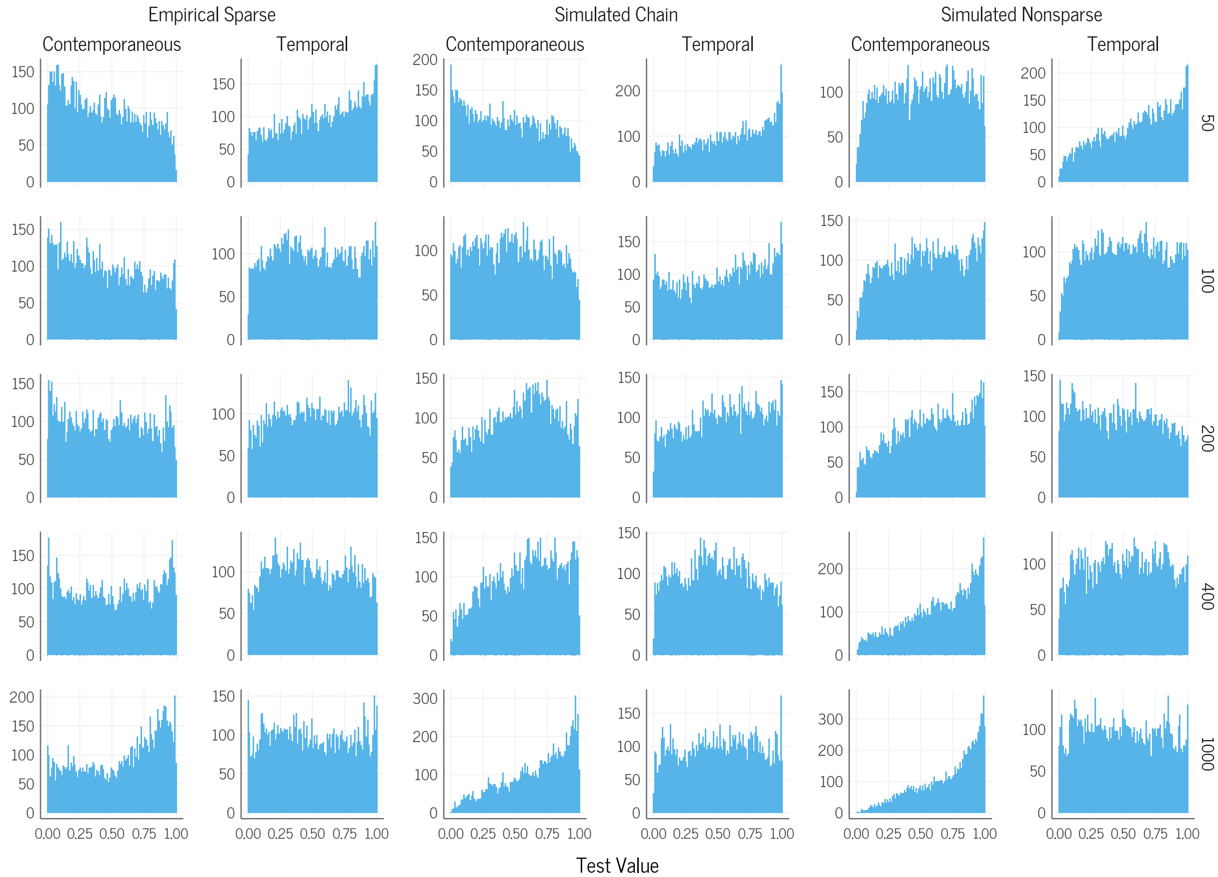
Trace Plots

Trace plots are an important MCMC diagnostic tool to check the mixing of a chain. Trace plot shows the value of the Markov chain at each point in the sampling process. Overall, results here looked similar to the example in Figure 7, indicated by the resemblance of the line to a hairy caterpillar.

We include code to create all trace plots for all parameters in the electronic supplement.

Figure 6

Distribution of Test Values under the Null for the Wide Prior.



Note. Using Frobenius norm only. X-axis displays different data-generating processes, y-axis dipslays different sample sizes.

Autocorrelation

Assessing the autocorrelation of samples is important to gain an insight into the efficiency or speed of the sampling as well as the quality of the resulting posterior distribution. Here, the lag-k autocorrelation is the correlation of a specific sample with a sample k values spaced apart. A high autocorrelation across larger values of k indicates slow mixing, and more iterations are needed to obtain reliable estimates. Results for the autocorrelation were very similar across individuals, but clearly different across different parameters. Coefficients of the temporal network exhibited a low autocorrelation past lag-1, which is desirable. However, coefficients of the contemporaneous network showed high

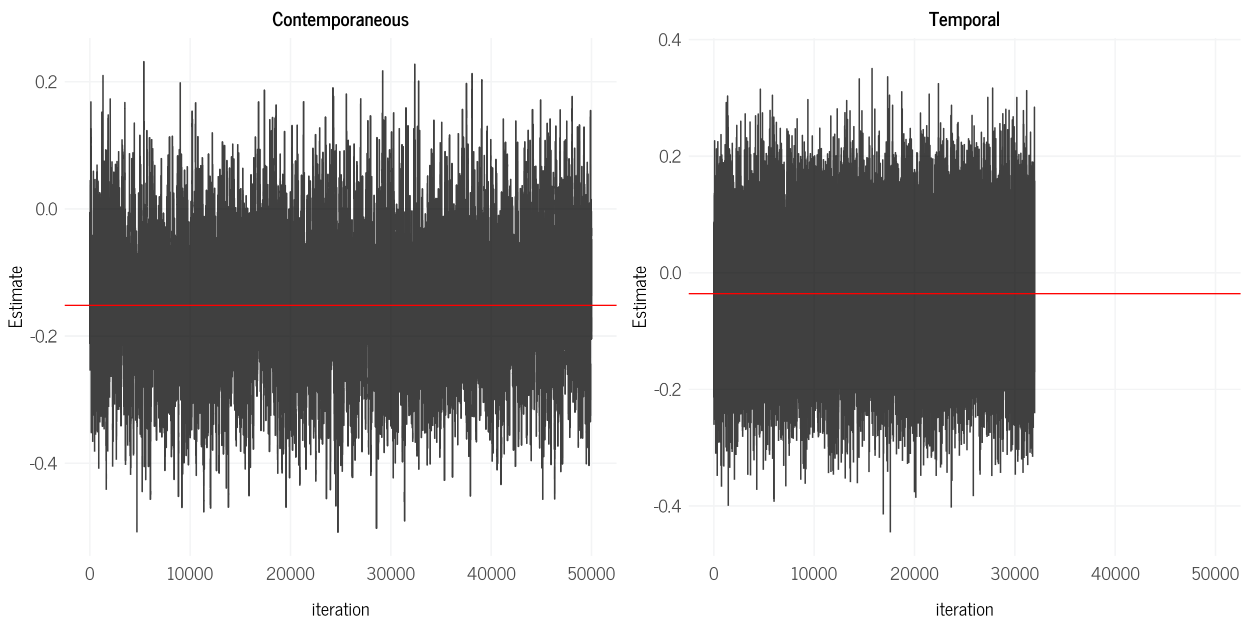


Figure 7

Example Trace Plot for Items Content and Concentration for ID 19.

autocorrelations across a range of lags. Example plots representative for this tendency are shown in Figure 8, where the values of the Autocorrelation function (ACF) on the y-axis are plotted against the lag value k on the x-axis.

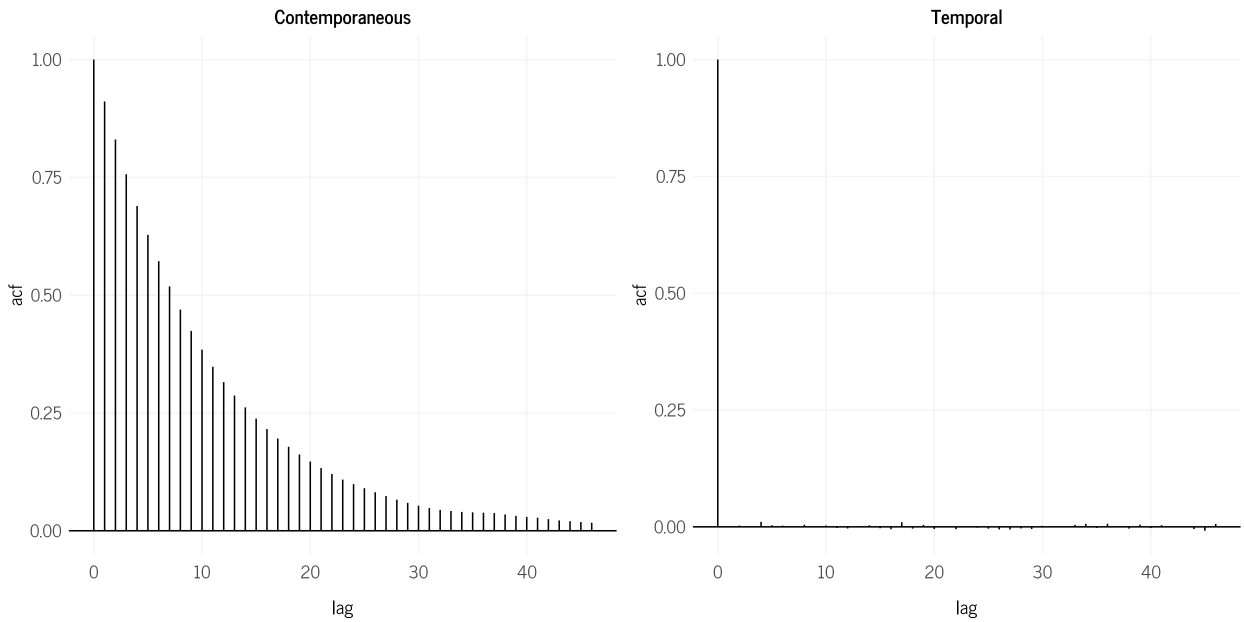
We include code to create all autocorrelation plots for all parameters in the electronic supplement.

Effective Sample size

We calculated the Effective Sample Size (ESS) using the `ess_gvar`-function in the *tsnet*, which implements the calculation of the *coda*-package (Plummer et al., 2020) for *BGGM* objects. It reflects the number of uncorrelated MCMC samples equivalent to a set of correlated MCMC samples (Roy, 2020). Due to the autocorrelation in the sampling for the contemporaneous network, the effective sample size for these parameters was substantially lower than for the temporal network parameters, which prompted us to use 50.000 iterations across all simulations. Specifically, the average effective sample size across all parameters in all temporal networks was 50020.31, whereas it was 2687.32 for the contemporaneous networks. Parameters within the networks were very similar overall. Across individuals, the

Figure 8

Example ACF Plot for Items Content and Concentration for ID 19.



ESS of the temporal parameters were very similar as well, ranging from a mean individual ESS of 49868.97 to 50161.13. On the contrary, the ESS of the contemporaneous network showed more heterogeneity, ranging from a mean individual ESS of 2142.43 to 4361.70.

Prior Sensitivity

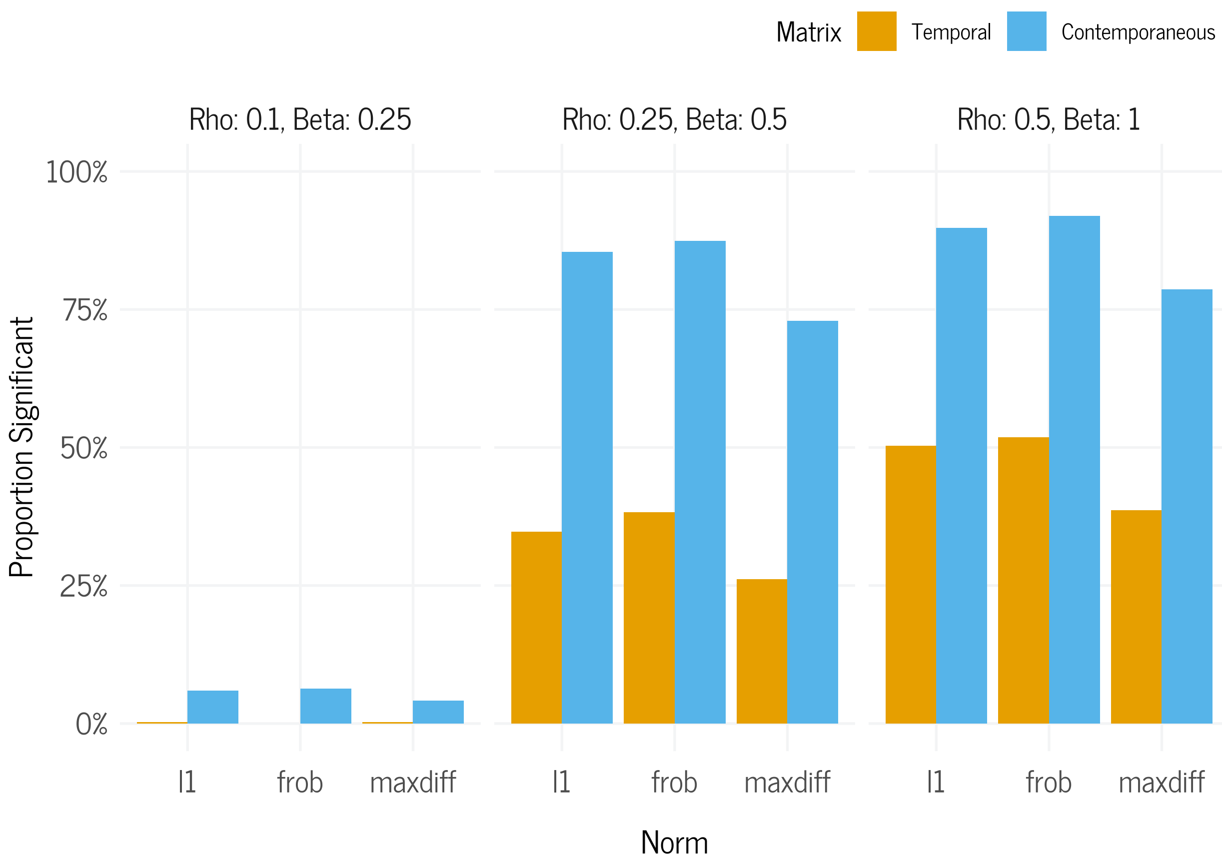
In Figure 9, we show the prior sensitivity of the number of positive test results for all 780 possible comparisons in the empirical example, computed as described in the manuscript. Clearly, using a very narrow prior prevents the detection of reliable differences between networks, as all estimates are pulled together towards zero across participants. The other two priors led to more similar results, where a wider prior leads to more positive test results. This difference is relatively more pronounced in the temporal network. We did not account for possible issues with multiple testing here.

Session Information

This section contains the R session information for the local machine that was used to analyze the simulations as well as the Server that was used to compute the simulations.

Figure 9

Prior Sensitivity Empirical Example.



Note. X-axis shows the three different norms, different grids along the x-axis show increasingly wide priors with different prior standard deviations for the contemporaneous and temporal network. Y-axis shows the proportion of positive test values across all possible pairwise comparisons.

Local Environment.

- R version 4.2.2 (2022-10-31 ucrt), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=German_Germany.utf8`, `LC_CTYPE=German_Germany.utf8`, `LC_MONETARY=German_Germany.utf8`, `LC_NUMERIC=C`, `LC_TIME=German_Germany.utf8`
- Running under: `Windows 10 x64 (build 22621)`
- Matrix products: default

- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, utils
- Other packages: BGGM 2.1.0, cowplot 1.1.1, doParallel 1.0.17, doRNG 1.8.6, dplyr 1.1.1, forcats 1.0.0, foreach 1.5.2, Formula 1.2-5, ggh4x 0.2.4, ggplot2 3.4.2, graphicalVAR 0.3, here 1.0.1, Hmisc 4.8-0, imputeTS 3.3, iterators 1.0.14, lattice 0.20-45, lubridate 1.9.2, Matrix 1.5-1, mgm 1.2-13, mlVAR 0.5, mvtnorm 1.1-3, purrr 1.0.1, qgraph 1.9.4, readr 2.1.4, reshape2 1.4.4, rngtools 1.5.2, rrapply 1.2.6, stringr 1.5.0, survival 3.4-0, sysfonts 0.8.8, tibble 3.2.1, tidyr 1.3.0, tidyverse 2.0.0
- Loaded via a namespace (and not attached): abind 1.4-5, arm 1.13-1, backports 1.4.1, bain 0.2.8, base64enc 0.1-3, BFpack 1.0.0, boot 1.3-28, checkmate 2.1.0, cli 3.6.0, cluster 2.1.4, clusterGeneration 1.3.7, coda 0.19-4, codetools 0.2-18, colorspace 2.1-0, compiler 4.2.2, corpcor 1.6.10, curl 5.0.0, data.table 1.14.8, deldir 1.0-6, digest 0.6.31, evaluate 0.20, extraDistr 1.9.1, fansi 1.0.4, farver 2.1.1, fastDummies 1.6.3, fastmap 1.1.1, fdrtool 1.2.17, forecast 8.21, foreign 0.8-83, fracdiff 1.5-2, generics 0.1.3, GGally 2.1.2, ggokabeito 0.1.0, ggridges 0.5.4, ggtext 0.1.2, glasso 1.11, glmnet 4.1-6, glue 1.6.2, grid 4.2.2, gridExtra 2.3, gridtext 0.1.5, gsubfn 0.7, gtable 0.3.3, gtools 3.9.4, hms 1.1.3, htmlTable 2.4.1, htmltools 0.5.4, htmlwidgets 1.6.2, httr 1.4.5, igraph 1.4.2, interp 1.1-3, jpeg 0.1-10, jsonlite 1.8.4, knitr 1.42, labeling 0.4.2, latticeExtra 0.6-30, lavaan 0.6-15, lifecycle 1.0.3, lme4 1.1-32, lmtest 0.9-40, magick 2.7.4, magrittr 2.0.3, MASS 7.3-58.1, matrixcalc 1.0-6, minqa 1.2.5, mnormt 2.1.1, MplusAutomation 1.1.0, munsell 0.5.0, network 1.18.1, nlme 3.1-160, nloptr 2.0.3, nnet 7.3-18, pander 0.6.5, patchwork 1.1.2, pbapply 1.7-0, pbivnorm 0.6.0, pillar 1.9.0, pkgconfig 2.0.3, plyr 1.8.8, png 0.1-8, pracma 2.4.2, proto 1.0.0, psych 2.2.9, quadprog 1.5-8, quantmod 0.4.20, R6 2.5.1, ragg 1.2.5, rbibutils 2.2.13, RColorBrewer 1.1-3, Rcpp 1.0.10, Rdpack 2.4, reshape 0.8.9, rlang 1.1.0, rmarkdown 2.21, rpart 4.1.19, rprojroot 2.0.3, rstudioapi 0.14, scales 1.2.1, shape 1.4.6, showtext 0.9-5, showtextdb 3.0, sna 2.7-1, splines 4.2.2, statnet.common 4.8.0, stats4 4.2.2, stinepack 1.4, stringi 1.7.12,

svglite 2.1.1, systemfonts 1.0.4, texreg 1.38.6, textshaping 0.3.6, tidyselect 1.2.0, timechange 0.2.0, timeDate 4022.108, tools 4.2.2, tseries 0.10-53, TTR 0.24.3, tzdb 0.3.0, urca 1.3-3, utf8 1.2.3, vctrs 0.6.1, withr 2.5.0, xfun 0.39, xml2 1.3.3, xtable 1.8-4, xts 0.13.0, yaml 2.3.7, zoo 1.8-12

Server Environment.

- R version 4.3.0 (2023-04-21), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=de_DE.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=de_DE.UTF-8`, `LC_COLLATE=de_DE.UTF-8`, `LC_MONETARY=de_DE.UTF-8`, `LC_MESSAGES=de_DE.UTF-8`, `LC_PAPER=de_DE.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=de_DE.UTF-8`, `LC_IDENTIFICATION=C`
- Time zone: `Etc/UTC`
- TZcode source: `system (glibc)`
- Running under: `Ubuntu 20.04.6 LTS`
- Matrix products: default
- BLAS: `/usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0`
- LAPACK: `/usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0`
- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, utils
- Other packages: BGGM 2.0.4, cowplot 1.1.1, doParallel 1.0.17, doRNG 1.8.6, dplyr 1.1.2, forcats 1.0.0, foreach 1.5.2, ggh4x 0.2.4, ggplot2 3.4.2, graphicalVAR 0.3.1, here 1.0.1, iterators 1.0.14, lubridate 1.9.2, Matrix 1.5-4.1, mgm 1.2-13, mlVAR 0.5.1, mvtnorm 1.2-1, purrr 1.0.1, readr 2.1.4, reshape2 1.4.4, rngtools 1.5.2, rrapply 1.2.6, stringr 1.5.0, sysfonts 0.8.8, tibble 3.2.1, tidyr 1.3.0, tidyverse 2.0.0

- Loaded via a namespace (and not attached): abind 1.4-5, arm 1.13-1, backports 1.4.1, bain 0.2.8, base64enc 0.1-3, BFpack 1.0.0, boot 1.3-28.1, checkmate 2.2.0, cli 3.6.1, cluster 2.1.4, clusterGeneration 1.3.7, coda 0.19-4, codetools 0.2-19, colorspace 2.1-0, compiler 4.3.0, corpcor 1.6.10, data.table 1.14.8, digest 0.6.31, evaluate 0.21, extraDistr 1.9.1, fansi 1.0.4, fastDummies 1.6.3, fastmap 1.1.1, fdrtool 1.2.17, foreign 0.8-84, Formula 1.2-5, generics 0.1.3, GGally 2.1.2, ggokabeito 0.1.0, ggridges 0.5.4, glasso 1.11, glmnet 4.1-7, glue 1.6.2, grid 4.3.0, gridExtra 2.3, gsubfn 0.7, gtable 0.3.3, gtools 3.9.4, Hmisc 5.1-0, hms 1.1.3, htmlTable 2.4.1, htmltools 0.5.5, htmlwidgets 1.6.2, httr 1.4.6, igraph 1.4.3, jpeg 0.1-10, knitr 1.43, lattice 0.21-8, lavaan 0.6-15, lifecycle 1.0.3, lme4 1.1-33, magrittr 2.0.3, MASS 7.3-60, minqa 1.2.5, mnormt 2.1.1, MplusAutomation 1.1.0, munsell 0.5.0, network 1.18.1, nlme 3.1-162, nloptr 2.0.3, nnet 7.3-19, pander 0.6.5, patchwork 1.1.2, pbapply 1.7-0, pbivnorm 0.6.0, pillar 1.9.0, pkgconfig 2.0.3, plyr 1.8.8, png 0.1-8, pracma 2.4.2, proto 1.0.0, psych 2.3.3, qgraph 1.9.5, quadprog 1.5-8, R6 2.5.1, rbibutils 2.2.13, RColorBrewer 1.1-3, Rcpp 1.0.10, Rdpack 2.4, reshape 0.8.9, rlang 1.1.1, rmarkdown 2.22, rpart 4.1.19, rprojroot 2.0.3, rstudioapi 0.14, scales 1.2.1, shape 1.4.6, sna 2.7-1, splines 4.3.0, statnet.common 4.9.0, stats4 4.3.0, stringi 1.7.12, survival 3.5-5, texreg 1.38.6, tidyselect 1.2.0, timechange 0.2.0, tools 4.3.0, tzdb 0.4.0, utf8 1.2.3, vctrs 0.6.2, withr 2.5.0, xfun 0.39, xtable 1.8-4, yaml 2.3.7

References

Plummer, M., Best, N., Cowles, K., Vines, K., Sarkar, D., Bates, D., Almond, R., & Magnusson, A. (2020). Coda: Output Analysis and Diagnostics for MCMC.

Roy, V. (2020). Convergence diagnostics for Markov Chain Monte Carlo. *Annual Review of Statistics and Its Application*, *7*(1), 387–412. https://doi.org/10.1146/annurev-statistics-031219-041300