

A Comparison of Stan Versus WinBUGS Software for Conducting Bayesian Network Meta-Analyses of Binary Outcomes

Youn J-H¹, Petersohn S¹, Jevdjevic M¹, Ainsworth C², Piena M¹, Gittfried A¹

¹OPEN Health Evidence & Access, Rotterdam, Netherlands, ²OPEN Health Evidence & Access, Manchester, UK



INTRODUCTION

- Bayesian network meta-analysis (NMA) is a common tool for evidence synthesis in health sciences. WinBUGS, released in 1997, has become the software of choice for conducting Bayesian NMA using Markov chain Monte Carlo (MCMC) sampling methods, with wide employment in health sciences.¹
- However, alternative sampling algorithms to the WinBUGS Gibbs sampling, such as Hamiltonian Monte Carlo (HMC), have been proposed as a more efficient solution in MCMC. Stan, a programming language released in 2012 and in active development today, uses a variant of HMC, the No-U-Turn Sampler (NUTS).² This may offer additional functionalities and ease of use over WinBUGS.
- This poster presents a case study comparing WinBUGS to Stan, using publicly available data on safety of different treatment regimens for patients living with human immunodeficiency virus (HIV).³

OBJECTIVES

- Compare the viability of Stan against WinBUGS when conducting a fixed effects Bayesian NMA for binary outcomes by means of a case study.
- Establish a framework of comparison between two different software when conducting a binomial NMA.

METHODS

Data

- The NMA used publicly available safety data for patients living with HIV.³ The endpoint assessed was the proportion of treatment-naïve patients experiencing an adverse event within the first 96 weeks.
- The network included 8 treatments with data from 9 trials, with both direct and indirect evidence represented (Figure 1).

WinBUGS

- WinBUGS generally performs MCMC by considering one parameter at a time and updating it according to the underlying joint distribution. This ‘one-by-one’ updating method thereby considers each dimension of the log density individually, but has the advantage of only performing the computations necessary per iteration.¹
- WinBUGS was implemented in R using the R2WinBUGS package.⁴

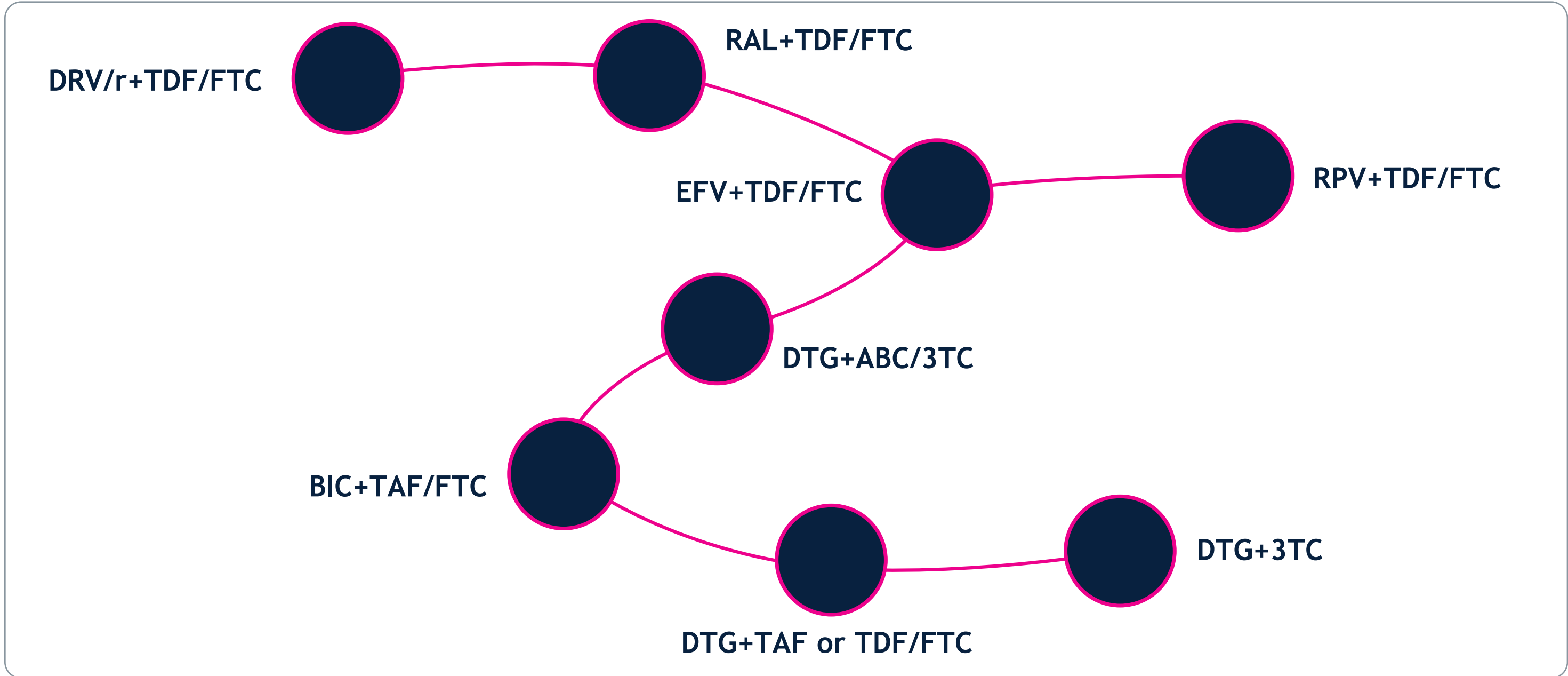
Stan

- In comparison to WinBUGS, Stan’s NUTS algorithm does not process one parameter at a time, but rather computes the entire space of all the parameters simultaneously.²
 - This avoids some of the difficulties which may occur when sampling one dimension in an extremely high-dimensional problem but requires calculating the entire log density for each iteration.
- Stan was implemented in R using the multinma (0.4.2) package and shinystan.^{5,6}

Evaluating the differences

- The following outcomes were considered relevant when comparing the two software and their respective sampling algorithms:
 - Median odds ratios (OR) and 95% credible intervals were assessed alongside treatment rankings, to ensure that the results are consistent between the software.
 - Running time, to determine the computation time required for the same number of iterations. Both NMAs were run on the same computer, using 10,000 iterations, 5,000 burn in iterations with a thinning rate of 1, and four chains.
 - Trace plots, to observe whether each model converged.
 - Qualitative outcomes regarding user experience, in particular error handling and console messages.

Figure 1. Network diagram



Abbreviations: ABC/3TC, Abacavir/lamivudine; BIC, Bictegravir; DRV/r, ritonavir-boosted darunavir; DTG, Dolutegravir; EFV, Efavirenz; RAL, Raltegravir; RPV, Rilpivirine; TD(A)F/FTC, Tenofovir disoproxil (or alafenamide) fumarate/emtricitabine.

REFERENCES

1. Lunn, D. J., Thomas, A., Best, N. et al. WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing* 10, 325-337 (2000).
2. Stan Modeling Language Users Guide and Reference Manual [program], 2022.
3. Nickel K, Halfpenny NJA, Snedecor SJ, et al. Comparative efficacy, safety and durability of dolutegravir relative to common core agents in treatment-naïve patients infected with HIV-1: an update on a systematic review and network meta-analysis.

BMC Infect Dis 2021;21(1):222.

4. Sturtz S, Ligges U, Gelman A. R2WinBUGS: A Package for Running WinBUGS from R. *Journal of Statistical Software* 2005;12(3):1 - 16. doi: 10.18637/jss.v012.i03.
5. Philippo, D. M. (2022). multinma: Bayesian Network Meta-Analysis of Individual and Aggregate Data. <https://doi.org/10.5281/zenodo.3904454>.
6. shinystan Vignettes [program], 2022.

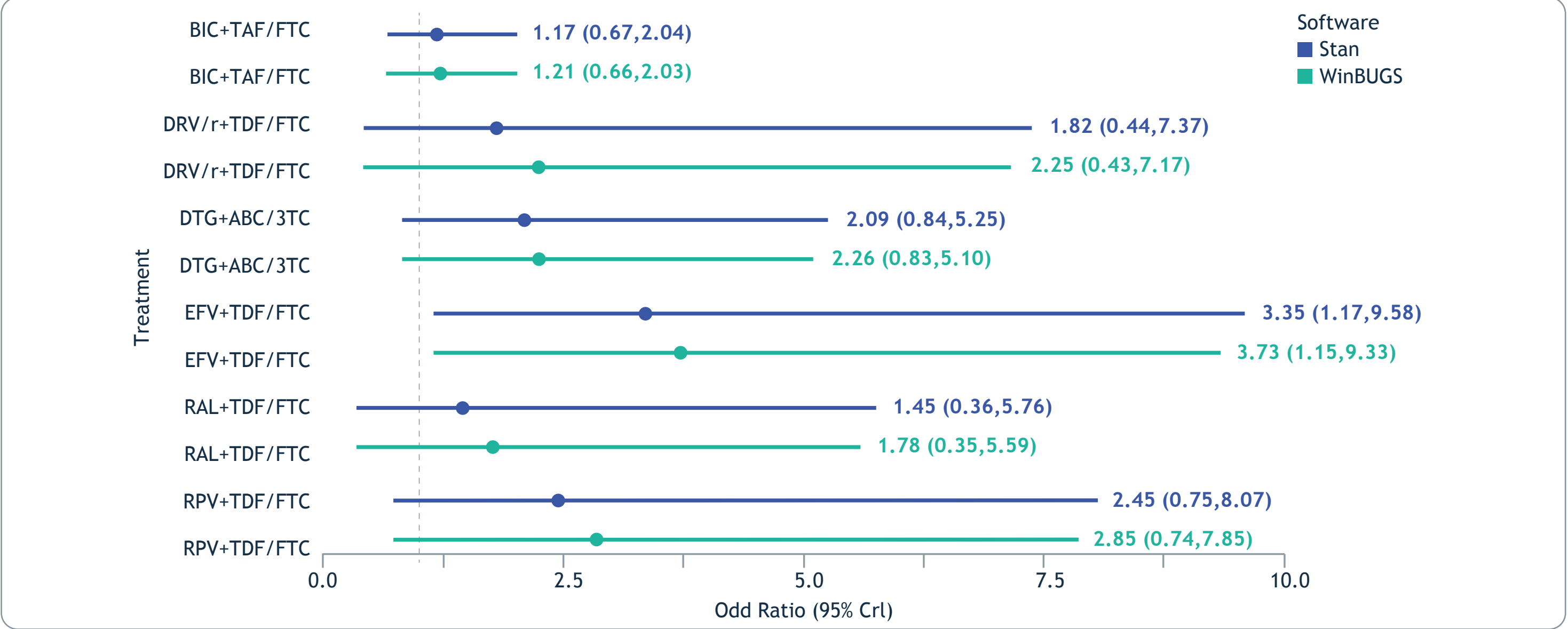
7. Lunn, D., Spiegelhalter, D., Thomas, A., & Best, N. (2009). The BUGS project: Evolution, critique and future directions. *Statistics in medicine*, 28(25), 3049-3067.

ACKNOWLEDGEMENTS

We want to thank Mark Lambton, Niels Dunnewind and Matt Knowles for their software support, Elisabeth Fenwick for her review of the analysis, and Sonja Kroep for her contributions conceptualizing this project.

RESULTS

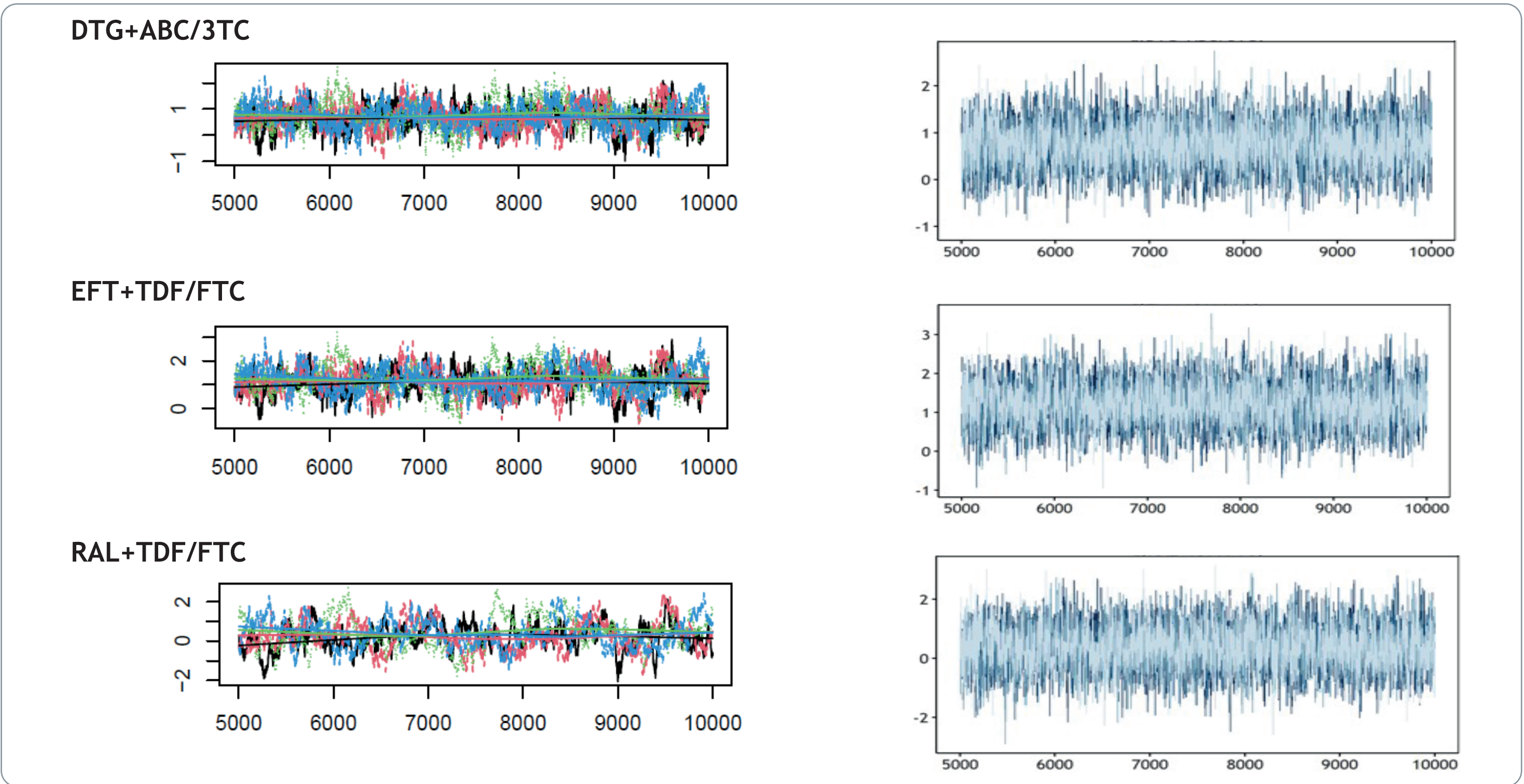
Figure 2. Forest plot showing the outcomes of DTG+3TC vs. comparators for analyses run in WinBUGS and Stan



Quantitative outcomes

- Figure 2 shows the point estimates and CrIs of the two analyses. Differences in the median OR were minor, ranging from 0.004 to 0.141. Treatment rankings and the interpretation of the results also did not differ between the software.
- Running time was improved with use of Stan, compared to WinBUGS which took 20.0 seconds. However, as WinBUGS did not converge with 10,000 iterations based on visual inspection of the trace plots, the results for point estimates and CrIs were assessed after 25,000 iterations, when the model had converged. This took 76.9 seconds.
- The trace plots (a selection of which are shown in Figure 3) show that the Stan algorithms converged with 10,000 iteration runs, while some WinBUGS traces did not.

Figure 3. Trace plots of WinBUGS (left) and Stan (right) for selected treatments



Qualitative outcomes

Users reported that Stan was perceived as easier to debug because of the additional information provided in error messages. Console warnings were seen as more actionable.

DISCUSSION

- The deterministic outcomes of OR, CrI and treatment rankings were very similar, confirming that Stan and WinBUGS lead to similar results even though they use different algorithms.
- Noticeably, WinBUGS was not able to converge using the same number of iterations as Stan, even though Stan required less time. The differences in running time were therefore exacerbated when considering convergence. Overall, this might indicate advantages in using Stan for more complex networks, as increasingly complex networks may be more likely to converge or take less running time using Stan.
- Differences in convergence and shorter running time indicate an advantage in using Stan, along with more user-friendly error messages. Stan made diagnosing NMA models exceedingly user-friendly with the shinyStan functionality.⁶ This likely follows due to the age differences between the software. WinBUGS has not been updated since 2007 due to development shifting to OpenBUGS,⁷ while Stan is still in active development.

CONCLUSIONS

- This case study allows for some conclusions to be drawn about the differences between WinBUGS and Stan in a binomial Bayesian NMA with a medium-sized network.
- Considering the similarity of the analysis results obtained from WinBUGS and Stan, and the advantages in running time, model convergence and user experience regarding for Stan, Stan can be considered the more convenient and reliable software for conducting binomial Bayesian NMA with a medium-sized network.