Response letter to reviewer comments on "RandomForestsGLS: An R package for Random Forests for dependent data"

We thank the Editors, the Associate Editor and the reviewer for their positive feedback and thoughtful comments which have helped to improve the manuscript. Please find our point-by-point responses below. The reviewer comments are in blue font and the responses are in black font.

1 Reviewer: Philip Waggoner

Reviewer Summary
Overall, this package is great. A useful extension of RF, and a great complement to the paper introducing the method. My feedback is mostly focused on high level items and involves fixes to ease consumption of the paper and code, and thus application and interpretation. No PRs as nothing major needed to be changed, by me at least. I hope there are some useful comments here for the authors. Thanks!

Authors' response: Thank you for the very encouraging feedback and the very useful suggestions that helped to improve the manuscript. In the revision, we have tried to address the technical comments with a focus on explaining the need and value of the software for a wider non-specialist audience as suggested by the reviewer.

Please find our point-by-point responses to the reviewer comments below.

1. the code, how is optimization defined when `param_estimate = TRUE` in the context of unknown covariance parameters? More defining and defending this (ideally in the paper and code/documentation) would be useful.

   Authors' response: Thanks so much for raising this point. We have expanded upon this in the manuscript and in code documentation as follows:

   - **Spatial Data:** More often than not, the true covariance parameters of the data are not known apriori. Given a choice from a prespecified list of covariance functions (software presently allows for exponential, spherical, Matérn and Gaussian covariances), the software accommodates parameter estimation with the argument `param_estimate = TRUE`. For this, first we fit a classical RF ignoring the correlation structure. Next, we fit a zero mean Gaussian process with the covariance structure of choice on the residual to obtain the estimate of the model parameters through maximum likelihood estimation. This initial RF fitting and estimation of the spatial parameters from the RF residuals is justified by theoretical results in Saha, Basu, and Datta, 2021 proving that even under dependence the naive RF is consistent (although empirically suboptimal). The estimated model parameters are then used to fit RF-GLS. As demonstrated in Saha, Basu, and Datta, 2021, this leads to significant improvement over classical RF both in terms of estimation and prediction accuracy. This procedure is again analogous to the linear model setup, where fitting a feasible GLS model often involves pre-estimating the covariance parameters from residuals from an OLS fit.

     It has also been demonstrated (Section 4.4 in Saha, Basu, and Datta, 2021) that even under misspecification in covariance structure (i.e. when the true spatial effects are generated from a different covariance structure than the specified covariance for estimation or arbitrary smooth

function), using RF-GLS with exponential covariance (the default choice of covariance function in the software) leads to noticeable improvement over simply using classical RF.

One additional consideration for model parameter estimation involves the computational complexity of the maximum likelihood estimation of Gaussian process. In its original form, this involves $O(n^3)$ computation and $O(n^2)$ storage space, which may be prohibitive for large n. This problem is thoroughly studied in spatial statistics literature (we refer the readers to Datta et al., 2016 for a detailed review). In the present scenario, we perform a fast, linear-time estimation of the model parameters with BRISC Saha and Datta, 2018, which is directly implemented in the present software through the R package BRISC (https://CRAN.R-project.org/package=BRISC).

- **Autoregressive (AR) Time Series Data:** For time-series data, most often the model parameters for the autoregressive process (order of autoregression and the autocorrelation coefficients) are unknown. These are estimated using a strategy similar to the spatial case. With the option `param_estimate = TRUE`, the software estimates the model coefficients by fitting a zero mean autoregressive process of user defined order on the residuals from a naive RF fit. The autoregressive parameter fitting is done using `arima` from base R package `stats`.

2. the code, and specifically this criterion from JOSS: "A summary describing the high-level functionality and purpose of the software for a diverse, non-specialist audience.", the summary (and statement of need by extension) don't fully meet this standard. The language does a job focusing on the computational benefits of `RandomForestsGLS`, as well as the value in a statistical sense. But the functionality and focus of the package (rather than the method), is lacking. The details and value of the method, though needed at a high level to understand the package, are fully unpacked in the saha2021random paper. So, I wanted much more focus on introducing and convincing a non-specialist, skeptical audience of the need and value of this software tool. To be sure, the details of the package construction and design are well-discussed. But the implementation of the package, and how it might be tied into the modal ML workflow, for example, are missing.

   Authors' response: Thanks for pointing this out. We have modified the summary of the article as follows:

   The primary purpose of the package 'RandomForestsGLS' is to fit nonlinear regression for spatial and temporal data. This package performs estimation through a novel rendition of Random Forests (RF); namely, RF-GLS, which makes use of the dependence structure in the data. With increasing computing capacity of personal computers, non-linear Machine Learning (ML) methods have become increasingly commonplace for regression and classification tasks due to their ability of capture complex interactions among the variables that cannot be modeled by linear regression. However, many modern ML software lack the capability to efficiently account for the dependence structure in the data which leads to sub-optimal estimation. On the other hand, specialized software for spatial/temporal data are capable of properly modeling data correlation, but usually assume the relationships between the response and the covariates. `RandomForestsGLS` brings the best of both worlds together by bridging the gap between these two approaches by explicitly modeling the spatial/serial data correlation in the Random Forests fitting procedure to substantially improve estimation of the regression function. In ML workflow, `RandomForestsGLS` can substitute existing ML methods in model training to take care of the dependence structure. Similarly, for traditional spatial modeling frameworks, `RandomForestsGLS` can be used instead of linear model based methods to account for nonlinearity. Additionally, `RandomForestsGLS` seamlessly leverages kriging to perform predictions at new locations for geo-spatial data, a primary objective in many spatial analysis.

3. Why only choose autoregression for the time series dependency? As with any method, there are several assumptions with this approach/method (namely, assuming autoregressive errors). Its definitely widely used and AR is often the most common type of history dependence, and thus a good starting place. But I'd recommend, perhaps even for later package versions, other time

series methods to be included in this framework, both parametric and nonparametric (e.g., ECM, ARFIMA, random walk, and so on).

Authors' response: The reviewer makes a very valid comment. There are many different models of temporal dependence and auto-regressive (AR) structure is only one particular (albeit, very popular) choice. We have opted for AR models for two reasons.

(a) AR models yield banded sparse precision (inverse covariance) matrices that substantially expedite (linear time) computations for RF-GLS

(b) Saha, Basu, and Datta, 2021 provides theoretical guarantees for RF-GLS under autoregressive temporal dependence. There is no theoretical support yet for other types of temporal dependence.

In future, as recommended by the reviewer, we plan to implement additional forms of time series dependency and perform thorough empirical validation, prior to making it available in the software.

4. the paper, there were many grammatical issues throughout (e.g., "felicitates" in the Statement of Need), as well as informal syntax (contractions like "doesn't" used throughout). I recommend cleaning up and revising the manuscript several times across several readers. These types of mistakes are a bit distracting.

   Authors' response: We have tried to clean up the manuscript to the best of our abilities.

5. the paper, I wanted to see a more explicit and clearer definition of the core concept, "dependency" up front. It is mentioned a lot throughout and in the title. The authors do a good job of relating the similarity of OLS $\rightarrow$ GLS, for the current move from RF $\rightarrow$ RF-GLS. And there is a reference to "spatial and temporal correlation" in the Summary. But other than this, I was a bit confused and often left wondering about the many other contexts, definitions and cases that "dependency" could mean. So a crisper set up and definition for such a central concept would really benefit the paper and help situate the reader right off the bat.

   Authors' response: Thanks so much for the suggestion. in the revised manuscript, we have clarified the notion of dependency more formally at the beginning of the article as follows:

   With the modern advancements in geographical information systems, remote sensing technologies, low-cost sensors , we are increasingly encountering massive datasets indexed by geo-locations and time-stamps. Modeling such high throughput spatio-temporal data needs to carefully account for spatial/serial dependence in the data, i.e., model the structures (patterns) of the data along space or time. A general model for describing dependent observations $(y_1, y_2, \cdots, y_n)$ and covariates $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ can be formulated as $y_i = f(\mathbf{x}_i) + \epsilon_i$, where $f(\mathbf{x}_i)$ denotes the mean component and ϵ_i is the residual error, which accounts for the dependency in the data, beyond what is explained by the covariates. In this article, we work with second order dependency, i.e. we assume that the dependence is captured through the covariance function of the correlated stochastic process ϵ_i. Since the data are indexed by locations (spatial data) or time-points (temporal data), the correlation is a function of the "spatial distance" or "time-lag" between two observations.

6. Though in the vignette, I don't get the purpose of the following in the `RFGLS_estimate_timeseries.Rd` manual page:

```
rmvn <- function(n, mu = 0, V = matrix(1)){
    p <- length(mu)
    if(any(is.na(match(dim(V),p))))
    stop("Dimension not right!")
    D <- chol(V)
    t(matrix(rnorm(n*p), ncol=p)\%*\%D + rep(mu,rep(n,p)))
}
```

I couldn't see anywhere `rmvn` was called. Could've missed something.

Authors' response: Thanks so much for pointing out this unnecessary inclusion in `RFGLS_estimate_timeseries.Rd`. We have removed it accordingly from the GitHub version of the package in my personal repository, and will soon push the update in CRAN.

7. could imagine core functions (e.g., `RFGLS_estimate_spatial`) being slow with big data sets. On replicating some of the parts of the vignette, it was pretty fast. But perhaps wrapping computation in a progress bar would be a nice UI addition.

   If you like this, happy to open a PR and drop it in the functions for each if it would help. Let me know.

   Authors' response: This would be great addition to the package. It would be really helpful if you could please open a PR and drop this in the function for each. Thanks so much for coming up with this. We will duly acknowledge this contribution in the paper.

2 Reviewer: Marvin N. Wright

Reviewer Summary

I think this is a very useful extension of random forests and a promising package. The examples where the methods outperforms standard RF are quite impressing! I have a few general questions, some on the package and some on the paper:

Authors' response: Thank you for your positive feedback and constructive comments on the package and the paper. We address the comments in detail as follows:

2.1 General Comments

1. From what I understand, there are two major differences to standard RF: The bootstrap procedure and the splitting rule. Why not take an existing RF package such as `randomForest` or `ranger` and make these changes instead of setting up a new package "borrowing some code from randomForest"?

   Authors' response: There are a few reasons, for which we thought that creating a separate package made more sense.

   - There is substantial novelty in the RF-GLS algorithm that makes its implementation quite different from that of RF. The reviewer is correct, that the novel aspects of our algorithm primarily concerns the splitting criterion and the bootstrap. However, these changes, which makes RF-GLS a generalization of RF, fundamentally changes some of the pieces of the algorithm.
     - Take for example, estimation of the node representatives. The classic RF estimates are actually OLS estimates (given the design matrix). However, one does not actually need to evaluate the traditional formula of the OLS estimate. Due to the orthogonality of the design matrix, the OLS estimate simply becomes the node means which can be calculated as a scalar quantity only once for each node created. However, the GLS estimate is an oblique (and not orthogonal) projection and does not have such a convenient form, and needs to be evaluated as a vector quantity for the entire set of nodes. Thus the node representatives need to be repeatedly and jointly updated for RF-GLS.

     - Similarly, for RF node splitting, the cost function is simply expressed in terms of the node means and variances of the parent node and the two potential children nodes. Hence, for each potential split location, the cost function can be evaluated by shifting one datapoint and using the well-known formulae for calculating leave-one-out means and variances. For RF-GLS, the cost function is a complicated quadratic form involving all the nodes and closed form expressions of the leave-one-out cost function is not available. To expedite brute-force evaluation of this expensive GLS cost function for every potential split,

we carefully exploit the sparse spatial dependence structure implied by nearest neighbor Gaussian Process. We derive an algorithm to update the design matrices for each potential split with minimal added computation. Please see "Scalable node splitting" under "Package Features" for details on this.

- Another reason we set it up as a new package is due to the nature of our target audience for the package. The package is primarily geared more towards the spatial (or time-series) audience and has added components in the algorithm for estimating the spatial/temporal covariance parameters (See sections corresponding to estimating "Unknown model parameters" in the software paper). Hence it made more sense to us to build a standalone package, which didn't require practitioners working with spatial/time-series data to go through a vast and multifaceted package like `randomForest` to find the suitable code.
- Additionally, we are working on extending RF-GLS for handling binary or categorical spatial data. As a result, the package will be going though a number of significant updates and expansion and provide a versatile inventory of tools for spatial analysis using Random Forests. It is easier for us to do that on a standalone package, than on `randomForest`.

2. Fitting RF-GLS is slower than standard RF. How much is it slower? How does it scale with the number of observations, covariates or other data or model parameters?

   Authors' response: The primary difference in computational overhead between RF and RF-GLS arises during evaluation of the split criteria. Cost is computed for each potential splits, by considering all the "gaps" among the corresponding covariate. In RF, this amounts to a $O(1)$ process due to convenient and separate leave-one-out updates to the scalar node means. In RF-GLS, this involves obtaining a joint minimum-norm vectgor-valued solution for least square problem, which is implemented through `DGELSY/DGELSD` in `Fortran`, with $O(t^3)$ computation, where t is the number of leaf nodes at the present split. This added computation is not a problem at the early stages of building a tree when the number of leaf nodes is small or if the maximum allowed number of leaf nodes is controlled. However, for very deep trees, this would lead to significant added computational burden in the later stages of tree growing. Hence, we recommend not using very large number of leaf nodes.

   Apart from this, RF-GLS and RF scale identically in terms of number of observations (linear), covariates and other model parameters (e.g. `mtry`, i.e. the number of randomly chosen covariates to be tested for optimizing each split). Linearity in the number of observations for RF-GLS is achieved by using sparse Nearest Neighbor Gaussian Processes Datta et al., 2016 for modeling the spatial dependence or autoregressive covariance for modeling temporal dependence.

   We are presently working on optimization of the implementation with parallelization of cost function evaluation.Additionally, we are also working on approximation methods that can lessen the burden of computational overhead in evaluation of split criteria, which will make the process more suitable for large data. We have also summarize the computational challenges in the "Discussion" section of the software paper.

3. Is a real data example available? That would be of interest for the method itself (not the focus here) but also for the package to see how it scales and for which real data purpose it can be used.

   Authors' response: Presently, we do not have a real data example available. As we have discussed earlier, we are working on few extensions of the proposed approach, which will expand the field of application of RF-GLS in terms of both the nature and size of the observed data set. Subsequently, We also plan to investigate the scope of application of RF-GLS in real data examples with the aforementioned modifications.

2.2 Package

1. The C++ code is not documented/commented well and hard to understand.

Authors' response: Thanks for the suggestion. I have incorporated comments in the C++ code for better understanding. Please let me know if you would prefer more detailed/extensive comments.

2. The DESCRIPTIONS still contains a link to arxiv, not the published paper.

   Authors' response: Thanks for pointing this out. We have corrected this in the Github version, will be pushing it to CRAN in the next update.

3. The README is missing a link to the paper.

   Authors' response: We have included the reference to the paper in the README.

4. Typo in README: criterion.

   Authors' response: Thanks for pointing this out. We have corrected the typo in the README.

5. Tests just run examples and check output types/sizes. That could be improved with more tests and tests that check for correct output.

   Authors' response: Thanks for the suggestion. We have incorporated new tests which check for correct outputs in the packages.

6. Is any kind of continuous integration used? I think it is useful to at least run the tests with each commit/PR.

   Authors' response: We have used continuous integration through GitHub actions in the newer version of the package.

7. Maybe too late to change that, but I think the package name is not a great choice. For example, at first try, I typed "randomForestGLS", then capitalized to "RandomForestGLS" and finally corrected to "RandomForestsGLS". It's also quite long and you have to remember the capitalization.

   Authors' response: Great comment by the reviewer. Since our method incorporates GLS in the RF framework, we wanted to demonstrate that in the name of the package, hence we initially opted for the name `RFGLS`, but we discovered that there already exists an R package with the same name https://CRAN.R-project.org/package=RFGLS , which is not related to this work. As an alternative, we opted for this name.

2.3 Paper

1. The JASA paper is called "Random Forests for Spatially Dependent Data", the software has the same name without "Spatially". Are additional types of dependencies covered by the software, not described in the original JASA paper?

   If yes, please detail in the JOSS paper.

   Authors' response: Great comment! The JASA article primarily focused on spatial data and provided simulation results corresponding to spatial data. It was also demonstrated in the article that the proposed approach can be adapted for temporal data, with AR(1) covariance. In the software, we account for both spatial and temporal dependencies, as described in the paper ("Autoregressive (AR) Time Series Data") and in the package vignette ("Autoregressive Time Series Data")

2. - line 8: Should be "in these models"
   - lines 14-15: "hence is not optimal in mixed-model approach". I don't understand this. Wouldn't RF be used as an alternative to the mixed model and not IN the mixed model approach?

- lines 18-19: Avoid linebreak in package name

Authors' response: Thanks so much for pointing this out. We have rewritten the summary and taken care of this in the revised manuscript.

3.
   - line 63: for or model correlation?
   - line 176: optimizing a cost function (missing a)
   - line 211: of the RF-GLS method (missing the)
   - line 216: "Efficient implementation thorough" should be through?

   Authors' response: Thanks so much for pointing out these typos. We have corrected them in the revised manuscript.

4. line 217: Maybe remove "clever"?

   Authors' response: Great suggestion. We have removed "clever" in the revised manuscript.

5. References: Datta et al. is in title case, others in sentence case.

   Authors' response: Thanks so much for pointing this out. We have corrected this in the revised manuscript.

6. In general, many spelling errors, missing articles, etc.

   Authors' response: Thanks for pointing these out. We have tried our best to correct the typos and grammatical errors in the revised manuscript.

References

[1] Abhirup Datta et al. "Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets". In: Journal of the American Statistical Association 111.514 (2016), pp. 800–812.

[2] Arkajyoti Saha, Sumanta Basu, and Abhirup Datta. "Random forests for spatially dependent data". In: Journal of the American Statistical Association (2021), pp. 1–19.

[3] Arkajyoti Saha and Abhirup Datta. "BRISC: bootstrap for rapid inference on spatial covariances". In: Stat 7.1 (2018), e184.