Dear Editor,

We thank the input from you and the reviewers. A revised manuscript has been prepared accordingly, with the changes being highlighted in blue. Please note that Section 5 on real data analysis is newly added, but we only highlighted the section title.

We are communicating with the editorial office and have requested to add Yuwen Liu as the 2nd author for the following reasons.

Yuwen fixed several bugs in the R package, in particular ones in the function cv.springer. The most recent version can be found from https://CRAN.R-project.org/package=springer. The updated package allows us to prepare a new simulation example using the cross-validation function, which can well address the questions from one reviewer. Yuwen has prepared not only the new simulation example, but also the real data analysis that is not available in the 1st version. Besides, he drafted the case study section.

Best regards,

Cen

Responses to the Second reviewer

The authors developed an R package “Springer” for bi-level variable selection of longitudinal data in gene-environment interactions studies under marginal models.

Major comments:

1. *The data generated in Section 4 only considered longitudinally measured response. Does the function “springer” can deal with longitudinally measured clinical covariates, environmental and genetic factors as in general model (1)?*

Response: Thank you for your comments. We have provided more details at the beginning of Section 4 as follows:

“In this section, we demonstrate the fit of bi-level selection using package \emph{springer} based on simulated datasets. Although model (1) is general in the sense that both the response and predictors are repeatedly measured, it can reduce to the case where the predictors, consisting of the clinical covariates, environmental and genetic factors, are cross-sectional under the longitudinal response. Model (1) is flexible in that the predictors can have a mixture of cross-sectional and longitudinal measurements. For instance, the repeated measurements are only taken on E factors, not on the clinical and G factors.

The motivating dataset for the sparse group variable selection developed in Zhou et al. \cite{zhou2022sparse} can be retrieved from the Childhood Asthma Management Program (CAMP) in our case study where the clinical, E and G factors are not repeatedly measured \cite{CAMP1,CAMP2,Covar}. Therefore, the current version (ver 0.1.7) of package \emph{springer} only accounts for such a case. It is worth noting that technically it is not difficult to extend the package to repeatedly measured predictors because the only difference lies in using time specific measurements, rather than repeating the cross-sectional measurements across all the time points, in the estimation procedure. We will discuss potential extensions of the package at the end of this section. In the following simulated example, the longitudinal responses are generated together with cross-sectional predictors. The data generating function is provided as below."

2. *In model (4), the authors considered the group level penalty via vector norm. How to specify Sigma_k for this norm? Which argument is related to Sigma_k in the core function "springer" or other supporting functions?*

Response: Thank you for your comments. We have provided definitions of the vector norm below equation (4). In addition, the following has been added to Section 3.2:

"Both the group- and bi-level penalization involve the empirical norm $||\eta_{nk}||_{\Sigma_{k}}$. In practice, the form of Σ_{k} is not unique. For example, Σ_{k} can be chosen as an identity matrix, then $||\eta_{nk}||_{\Sigma_{k}}$ reduces to a L2 norm. While the alternatives might be equally applicable, the default choice of Σ_{k} in package \emph{Springer} is in the form discussed in Section 2.3."

3. *The authors suggested to use LASSO estimate of beta as the choice of beta0 in "springer" function under the data taken from a single time point. I wonder whether the final variable selection results and the running time are sensitive to the single time point that the users choose. Moreover, the dataset is often unbalanced in practice. How to choose a proper single time point in this case?*

Response: Thank you for your comments. We have added the following paragraph in Section 4:

"Other choices of initial values include fitting ridge regression or LASSO under the average of within subject measurements, which accommodate the case of unbalanced data where a proper single point might be difficult to be determined. In general, the regularized estimates remain relatively insensitive to different choices of initial value $\hat{\beta_0}$, as long as $\hat{\beta_0}$ is reasonable, in other words, not extremely far away from the optimal solution."

To further address your concern, we rerun the codes by using different single time points, as well as the average of within subject measurements. We can observe that the metrics on running time on a regular laptop, estimation accuracy (TMSE, MSE, NMSE) and identification (TP, FP, etc.) remain insensitive under different initial values.

	Time (minutes)	TMSE	MSE	NMSE	TP	FP	TP1	FP1	TP2	FP2
y[,1]	28.797	0.003454	0.06564	0.0002168	21	3	6	0	15	3
y[,2]	30.739	0.003491	0.06695	0.0001859	21	3	6	0	15	3
y[,3]	29.601	0.003910	0.07493	0.0002114	18	4	6	0	12	4
y[,4]	30.791	0.003556	0.06764	0.0002180	20	3	7	0	13	3
y[,5]	29.245	0.003874	0.07379	0.0002335	18	3	6	0	12	3
ybar	32.327	0.003661	0.06996	0.0002080	20	2	6	0	14	2

Here ybar= (y[,1]+y[,2]+y[,3]+y[,4]+y[5,])/5.

4. The authors mentioned that there are three different working correlations can be called through the input argument “corr” in function “springer”. How about other working correlation structure, for example, unstructured working correlation matrix, which is often used when we do not have enough information about the true within-subject correlation?

Response: Thank you for the suggestions. We have added the following paragraph at Section 4:

“The current version of package springer (Version 0.1.7) has implemented three working correlation matrices, that are independence, AR1 and exchangeable, for individual-, group- and bi-level variable selection under continuous longitudinal responses in both the GEE and QIF framework. The future improvement includes incorporating other working correlations, such as the unstructured working correlation. A question worth exploring is the computational feasibility of unstructured working correlation under QIF as the large number of covariance parameters will potentially lead to much more complicated extended score vector, incurring prohibitively heavy computational cost for high dimensional data. We will also consider extensions to discrete responses such as binary, count and multinomial responses, as well as longitudinally measured clinical, environmental and genetic factors, especially after these data are available. ”

5. *Both QIF and GEE can deal with continuous and discrete responses. Theoretically, model (4) can deal with both continuous and discrete phenotypic responses. Can function “springer” be extent to general case?*

Response: Thank you for your suggestions. Please refer to the response to the question above.

Responses to the Third reviewer

The authors development a new R package for bi-level variable selection. However, I found some major flaws on this paper.

1. *Please add some real data for demonstration purpose, also add some functions so user can simply library(Springer), use function_x1(input.csv,...) to get Data.list include e.train, g.train, y.train, beta0, etc., then use springer() to get parameters, and finally use funciton_x2(fit.beta) to print results.*
Response: Thank you for the suggestions. We have added a case study to showcase the usage of the package on high dimensional omics data with longitudinal responses in Section 5. By following the NIH guideline, we cannot share the data publicly. The data can be applied from dbGap through accession number phs000166.v2.p1.

The *e.train, g.train, y.train* are simulated data which are directly generated using R code in the R console, so we do not convert them back to external files and load the data again through external files using *function_x1(input.csv,...).* Currently the *springer* function only returns the fitted coefficient vector, which can be viewed by R code "*print(fit.beta)*".

For real omics data, the input.csv file is not available because the environmental, clinical, phenotypic and genetic data are saved on different types of files from public databases such as TCGA, dbGap, etc. The genetic data are usually saved in multiple non-csv files given the large scale so it is not likely to have one *function_x1(input.csv,...)* that can suit all the cases of data loading in practice. The most effective way is to load these data from separate files sequentially into R console.

2. *Since it is a package, the authors may add vignette for quick start.*

Response: Thank you for the suggestions. In this paper, we have illustrated the methodology of implemented statistical methods. In addition, examples of using the package to analyze both the simulated and real data have been provided. At this point, we do not pursue to add vignette as it will have a significant overlap with the software article. In the future, we will consider adding the vignette if the bi-level selection methods are extended to handle generalized longitudinal responses including the binary, count and multinomial data.

3. *I did not see the accuracy in parameter estimation, as the authors written in the final paragraph of page 12. fit.beta is very different from dat.train$coef, perhaps the authors can provide with a better example?*

Response: The TMSE, MSE and NMSE have been used to measure accuracy in parameter estimation. In the revised version, we have provided an equation to compute TMSE and more details to clarify the three metrics in Section 4.

We have also added the following paragraph below the chunk of codes computing metrics in parameter estimation:

"The *dat.train$coef* only consists of the nonzero coefficients utilized to generate longitudinal response in the data generating model, therefore its dimension is not the same as *fit.beta* as the estimated regression coefficient vector is sparse and includes 0 coefficients, thus having a much larger dimension. In regularized variable selection, the nonzero coefficients from *fit.beta* will not be identical to those in *dat.train$coef* due to the shrinkage estimation in order to achieve variable selection."

4. *The authors claim use LASSO to initialize beta0, but use Ridge instead in numerical example (alpha = 0).*
<u>Response</u>: Thank you for pointing out the typo. We have made a correction in Section 4. We have also revised the following in Section 3.1:

"Typical choices of *beta0* include the LASSO or ridge estimates under the cross-sectional phenotype measured at one of the time points, or the average of the within subject phenotypic measurements."

5. *How to distinguish from tp.main and tp.interaction? In numerical example section the coefficients of these nonzero effects are randomly generated.*

<u>Response</u>: Thank you for the comments. We have clarified it in Section 4:

"In the following codes, *tp*, *tp.main* and *tp.interaction* represent the locations for all the nonzero effects, that is, the column number of the corresponding effects in the design matrix. Although the coefficients are randomly generated from uniform distributions, the locations of nonzero effects are fixed."

6. *I change estimated parameters randomly: coeff[tp] = runif(30,-0.5,0.5), it gives me something like TP=23, FP = 5, TP1 = 6, FP1 = 0, TP2 = 17, FP2 = 5; Is this TP and FP reports really reliable? And actually when I use coeff[tp] = runif(30, 0.2, 0.5), the randomly generated TMSE, MSE, etc. outperform the given example(TMSE = 0.0009950036), this should never happen.*

<u>Response</u>: Thank you for the comments. The reported TP and FP are not always the same, if any changes have been made to alter the way of generating the data, nonzero coefficients or selecting different tuning parameters. Those metrics including TMSE are comparable only when the data are identical. When using "*coeff[tp] = runif(30, 0.2, 0.5)*" to generate coefficients, the data are already different from the example so comparison is not made on the same ground.

In the first version, the tuning parameters are randomly chosen. During the revision, we have fixed a couple of bugs in the cross validation function, allowing us to prepare an improved example. Now we select the optimal tuning parameters using *cv.springer* first, and then fit the model based on the chosen tuning parameters. Please refer to Section 4 for the new example.

The most recent version of the package is available on CRAN:

https://CRAN.R-project.org/package=springer

7. *I'm not sure what coef*mat (page 11 line 2) could produce, it is a vector of coefficients "*" a matrix.*
Response: We have revised the paragraph under the data generating function in Section 4 as follows:

"In the data generating function, *coef* represents the vector of nonzero coefficients, and *mat* is the part of design matrix corresponding to the main and interaction effects associated with nonzero coefficients. With (*n,p,q*)=(400,100,5), *coef* is a vector of length 30, and *mat* is a 400 by 30 matrix. The R code *coef*mat* denotes element-wise multiplication by multiplying the nonzero coefficient to the corresponding main or interaction effects. Therefore, *rowSums(coef*mat)* returns a 400 by 1 vector. The code "*0.5+rowSums(coef*mat)*" stand for the combined effects from those important main and interaction effects, as well as the intercept, with 0.5 being the coefficient multiplied to the intercept."

8. *Package "splines" and "mvtnorm" are not involved.*

Response: We have removed the two packages from the example.