Cover letter

R Journal	:	Manuscript No. 2023-79
Title	:	SNSeg: An R Package for Time Series Segmentation via Self-Normalization

June 10, 2024
Prof. Rob Hyndman
Editor, R Journal

Dear Professor Hyndman,

We would like to submit a revised version of our manuscript titled "SNSeg: An R Package for Time Series Segmentation via Self-Normalization" (Manuscript ID: 2023-79) for possible publication in *R Journal*.

We have carefully considered the comments from the two reviewers and have revised the paper accordingly. The detailed responses to the two reviewers are attached as separate files.

Below we highlight the major changes in the revision:

- We have adjusted the inner computation step of the function `SNSeg_Uni()` when users define their own functions to estimate changes in specific types of functionals that are not covered by our pre-defined functions. The new function possesses the same inputs and outputs as that of the old version but the execution time drastically decreases.
- We have further added a new Appendix section, which includes the detailed settings of the functions in R packages used in our comparison study reported in Tables 3 and 4.
- All other minor comments have been addressed.

We thank you and the reviewers for your time in reviewing our work. The comments helped improve the quality of our paper. For easy referencing, new materials in the first-round revision are highlighted in blue and those in the second-round revision are highlighted in red in the manuscript. We hope that you now find it a suitable contribution to the *R Journal*.

Sincerely,
Shubo Sun, Zifeng Zhao, Feiyu Jiang, Xiaofeng Shao

Reviewer 1

We thank you for careful reading of the paper and providing us with detailed guidance to further improve the work. We have considered each remark. Below is our response to each concern/comment. The review comments are listed in *italic* while our responses are in normal font. We hope that you now find the paper a suitable contribution to *R Journal*.

General Comments.

The concerns raised in the first report have mostly been addressed by the authors - thank you. I just have a few additional comments on the current version of the paper that have not been resolved satisfactorily.

Response: Thank you for your feedback and recognition on our improvements!

Major Corrections

Comment 2

The authors still need to specifically make it clear and state that the SNCP is a test statistic designed to effectively deal with temporal dependence within the data, which then employs a Binary Segmentation (BS) technique to find multiple changepoints. Whilst the authors insist this is not the case in their response 2, in response 3 they use their test statistic with Wild and Seeded BS which demonstrates the ease with which it can be switched. I agree with the authors that their test statistic doesn't suffer from the same drawbacks of BS due to the construct of their test statistic (using the local windows within each test) but this can be explained in the manuscript. Using BS is not a negative, it is a very useful technique which easily allows people to extend single changepoint models to multiple changes. Explicitly stating this will help readers to understand what you are doing in reference to what they already know and will improve readability.

The authors are still confusing search method and test statistic - explicitly in the new paragraph before section 2.3 and in Section 4.2. SNCP should refer to your test statistic that could be employed with different search strategies, for which you choose to use BS. The original BS uses a CUSUM test statistic and the presented approach replaces this with a SNCP test statistic, they both use the same sequential binary splitting algorithmic structure. For comparison with other methods you cannot say "BS" and "PELT" as these are search methods and not test statistics. In particular, the authors state that they use the cpt.mean() and cpt.var() functions from the changepoint package which have multiple test statistics implemented. Thus writing PELT says nothing about what test statistic was used.

Response: Thank you for the detailed comment.

First, we agree with you that we can indeed say that SNCP employs the BS algorithm and it remains effective for identifying multiple change-points due to the design of our test statistics. To clarify this point, we have added a new paragraph in red color after the last paragraph of

Section 2.2 to explain that another way to view our method is to treat SNCP as the test statistic and BS as the search algorithm. We hope this additional explanation will make SNCP more comprehensible to readers.

Second, in response to your comment on BS and PELT, we clarify that we treat SNCP as a change-point detection method and compare it with other change-point detection methods implemented via search methods BS and PELT. For convenience, we stick with the name BS and PELT. We agree that the type of test statistics used by the functions `cpt.mean()` and `cpt.var()` were not clearly stated in the manuscript. To address this, we provided a detailed description that BS and PELT use the normal test statistic (i.e., time series data are assumed to be normally distributed) in our comparisons. These details can be found in the Section "Further Details of Comparison Settings" in the Appendix.

Comment 4

The abstract still contains misleading terms like "laborious tuning parameters" when referring to other methods and "effortless tuning" when referring to their own methods and these terms need to be changed or removed. This is misleading as many existing methods offer default values for practitioners that often work well and so are not "laborious". Whether these default values are helpful, for existing methods or these methods, is dependent on the situation and is for the user to tune. Arguably, some other methods have more intuitive parameters to set than SNCP.

Response: Thank you for the comment. We acknowledge that the terms "laborious tuning parameters" and "effortless tuning" are misleading. The default values of specific functions in other packages can perform well in our comparison study. Considering this point, we removed "laborious tuning parameters" and "effortless tuning" from the abstract.

Comments on the additions

Comparisons

The descriptions of the comparisons are lacking. Please provide a full description of the functions and settings used - perhaps in an appendix. I'm pleased that the authors include the independent case in their comparisons. However, there is no mention in the new section that this is an unfair comparison because the methods were not designed to account for autocorrelation. The authors only compare to methods that assume independence and not methods designed for dependence like those in AR1seg, EnvCpt or fastcpd. It is obvious that methods designed for independence break down when rho=0.7, it would be odd if they didn't!

It is documented that methods that assume independence can modify their penalty by the assumed (or robustly estimated as in Shi et al. ()) correlation to produce methods that don't over/under segment. A note that this is possible, maybe in the concluding paragraph before section 4.3 would be appropriate.

Response: Thank you for the detailed comment. Firstly, the methods included in our comparison study are indeed only designed for time series data with no serial dependence. In response to your comments, we now note in the concluding paragraph of Section 4.2 that by selecting appropriate tuning parameters in the built-in penalty function, the independence-based methods can mitigate over-segmentation or under-segmentation issues when temporal dependence exists in the data. Secondly, we further acknowledged in the revised manuscript that there are indeed packages such as AR1seg (Levy Leduc, 2014), EnvCpt (Killick et al., 2021) and fastcpd (Li and Zhang, 2024) designed for serially dependent time series. However, it may be unfair to compare our fully nonparametric method with theirs as those methods are based on parametric models.

Comment on tables

Units should be included for the time column. Are these presented time in seconds, minutes etc? Please can you embolden the best performing algorithm in each setting for each metric to make the results clearer.

Response: Thank you for the question and constructive suggestions. The presented time in all tables of this paper is in seconds. For clarification, we have added the unit "s" to the time column for each table. We also bolded the best performing algorithm for each comparison metric in each setting. Please refer to Section 4 for these modifications.

References

Levy Leduc, C. (2014). R Package "AR1seg". Available on the CRAN at https://cran.r-project.org/web/packages/AR1seg/index.html.

Rebecca Killick, Beaulieu Claudie, Taylor Simon, and Hullait Harjit. (2021). R Package "EnvCpt". Available on the CRAN at https://cran.r-project.org/web/packages/EnvCpt/index.html.

Xingchi Li and Xianyang Zhang. (2024). "fastcpd: Fast Change Point Detection in R." Available on the CRAN at https://cran.r-project.org/web/packages/fastcpd/index.html. arXiv preprint arXiv:2404.05933.

Reviewer 2

We thank you for careful reading of the paper and providing us with detailed guidance to further improve the work. We have considered each remark. Below is our response to each concern/comment. The review comments are listed in *italic* while our responses are in normal font. We hope that you now find the paper a suitable contribution to *R Journal*.

General Comments:

Thanks to the authors for taking the time and effort to address the questions. My questions have been addressed satisfactorily. I have a few additional comments on the new materials added.

Response: Thank you for your feedback and recognition on our improvements!

Detailed Comments:

Comment 1

For the new tables added from Page 16 onwards, to increase readability:

- *Include in the captions brief descriptions of the column headings.*
- *Add units to 'time'.*
- *Bold the best results.*

Response: Thank you for your constructive suggestions. In the revised version, we have bolded the best result for each comparison metric in each setting across all tables and added the unit "s" in the time column to indicate that the presented time is in seconds. Additionally, we have included brief descriptions of each column heading for all tables, with these new descriptions highlighted in red.

Comment 2

In the demo of user-defined functions on Page 9 and 10, execution times reported are 22 and 13 minutes. Although it is useful to allow user-defined functions, the execution time is prohibitively long. Can the authors provide an additional option to reduce the execution time to a more practical range e.g. by skipping some calculations?

Response: Thank you for your question regarding the execution time. The calculation of statistics $L_n(t_1, k, t_2)$ and $R_n(t_1, k, t_2)$ in equation (7) of Section 2.2 is essential to find the maximal SN-based test statistics. To compute $L_n(t_1, k, t_2)$ and $R_n(t_1, k, t_2)$ for all t_1, k, t_2 combinations, it requires the value of all $\hat{\theta}_{i,j}, 1 \leq i < j \leq n$, where $\hat{\theta}_{i,j}$ is the user-defined functional computed on the subsample $\{Y_t\}_{t=i}^{j}$.

In the previous version of our package, we recomputed $\hat{\theta}$'s each time we compute $L_n(t_1, k, t_2)$ and $R_n(t_1, k, t_2)$ for different t_1, k, t_2. This is not efficient as it does repeated computation for some $\hat{\theta}_{i,j}$ and hence is time-consuming. Following your suggestion, we now calculate $\hat{\theta}_{i,j}$'s for all i, j beforehand and store the results. We then plug the corresponding $\hat{\theta}_{i,j}$ into equation (7) when computing $L_n(t_1, k, t_2)$ and $R_n(t_1, k, t_2)$ for different t_1, k, t_2. The amount of time decreases significantly from 22 and 13 minutes to only around 1 minute in the new version of our package. We refer to Example 2 in Section 3.1 of the manuscript for the updated execution time.