Dear Dr Petr and co-authors,

Three reviewers have now read and commented on your manuscript.

All of them found your work very interesting, scientifically sound and well presented.

They also provide some comments which I think could further improve your work.

Therefore, I recommend to carefully consider their suggestions and submit a revised version of your manuscript alongside a point-by-point rebuttal letter.

Best regards,

Emiliano Trucchi

Dear Dr. Trucchi,

We are happy to present a revised manuscript describing our *slendr* simulation R package.

Since the initial submission of our preprint, *slendr* has already found early adopters in research and teaching. Now, thanks to the reviewers' constructive feedback, we believe that the revised manuscript provides a much better overview of *slendr*, including a clearer discussion of its features in the context of SLiM and *msprime*, its potential for development of future inference methods, and the care that should be taken by future studies attempting to utilize complex population genetic models – three themes which have underlined nearly all of the reviewers' comments.

Below we provide our responses to each of the reviewers' points, all of which we have been able to address with relatively minor additions to the text – no changes to the software itself have been necessary.

Thank you for your work on managing the review of our manuscript. We are looking forward to hearing more about the progress of the evaluation and potential further comments.

Best regards,

Martin Petr and co-authors

Reviewer #1

> *This article by Petr et al presents a new R package called slendr, which is wrapper aiming at facilitating the simulation of genomic data distributed in space and time. This simulated data can then be used to make inferences by comparing it to observed data. The package is divided in three parts, which are intended to be used one after the other but can also be used independently. The first part allows the user to design a spatiotemporal simulation framework of population dynamics and genomic diversity. The second part can be used to call two different already existing simulators (SLiM or msprime) based on the virtual world created during the first part. These two simulators have different characteristics, including individual-based vs coalescent-based. The third part allows to analyze data generated during the second part by directly computing statistics or outputting files to be analyzed with other analytical programs.*
>
> *I fully agree with the authors that the spatial dimension of population genomics is important and often neglected or considered in a simplistic way, due to lack of available tools. From this point of view, I see the interest of the R wrapper developed by the authors, whose aim is to facilitate the use of approaches that consider the spatiotemporal dynamics of populations when studying genomics data. The main strength of slendr is that it encapsulates in the commonly used R language a whole series of programs written in different programing languages. Using R is therefore the only requirement for slendr users, without having to know/learn other languages.Overall, I find that the manuscript and accompanying documentation are well written and clearly present the use of slendr.*

Thank you for your review and your insightful comments on our manuscript.

> *In my opinion, there are several points that need to be improved before publication:*
>
> *I can see the value of slendr and the possible future developments, but these developments are not trivial and I think the authors should distinguish even more clearly between what is currently feasible with slendr and what is for future developments, perhaps with separate sections. For instance, the authors cite different approaches or programs currently available and their limitations, which they aim to overcome. However, some of these improvements have not yet been achieved. For example, they cite in introduction the program SPLATCHE as being limited to two interacting populations, but the current version of slendr only allows the simulation of one population (!) as I understood it in reading the discussion. In this respect, the current version of slender does not overcome the limitations of alternative approaches, although future developments may do so.*

We must clarify that *slendr* does, in fact, support simulation of arbitrary numbers of populations and not just a single population and we apologize that this was not clear in the original

submitted version of the manuscript. In this particular aspect, *slendr* is as flexible as its two internal simulation back ends, SLiM and *msprime*. To clarify this, we have changed the sentence in the discussion section which previously stated

> *"At the moment, slendr can only produce genome sequences from populations of a single species due to the restrictions imposed by its simulation back end."*

to

> *"At the moment, slendr can only produce genome sequences from a single species (**although with an arbitrary number and spatial arrangement of population groups**) due to the restrictions imposed by its simulation back end."* (line 772),

as we believe this is where this misunderstanding originated from.

As for the other reviewer's more general comment, we admit that the manuscript in its original form did not make it clear what kinds of models are possible to simulate with *slendr* compared to standalone SLiM and *msprime*—an issue highlighted by other comments as well. To fix this problem, we have added an entire new section titled "Relationship of *slendr* to SLiM and *msprime*" that discusses this in greater detail (lines 393-422). The section also highlights some of the *slendr* features which are on the roadmap for future versions of the package. We have also added a note encouraging the community to contribute feature requests on our GitHub page. Thank you for this suggestion!

> *Although SLiM and msprime are very flexible, large migration matrices between many populations are difficult to implement for models with complex geographical features. I am not sure how much the use of slendr simplifies the creation of these population interaction matrices compared to the original programs. As far as I understand, one instruction per population is required, which may be a limitation in the complexity of the models that can be implemented. To my opinion, this should be better explained.*

The reviewer raises a good point that complex models with a large number of migration events will require a non-trivial amount of code regardless of the technology used. Whether or not an R/*slendr* script encoding such a model is simpler compared to a pure *msprime* or SLiM equivalent will be, to a large extent, a matter of personal preference and comfort with each programming language. That said, for users who are more comfortable with R, we hope that *slendr*'s interactive interface does make designing a little easier while hopefully minimizing the chance of some particular types of bugs.

For instance, since populations in an interactive R *slendr* session are represented by standard R objects internally carrying their entire demographic history, every call to a `gene_flow()` function will perform a series of checks making sure each given gene-flow event is consistent with the demography encoded up to that point of a population's history and raises

an informative error if not. Therefore, *slendr* can catch many frequent bugs (such as mis-specified times of gene flow events, i.e. events involving not-yet-existing populations at a particular time point) before a computationally costly simulation is even run. The "populations-as-R-objects" aspect of *slendr* also makes it easy to write code which can programmatically generate even large lists of individual *slendr* gene-flow events automatically by operating on population R objects, while, again, enforcing the correctness of such events during the model-definition phase, before the simulation itself. Finally, the possibility to visualize the entire model including gene-flow arrows (either as a standard tree, or on a spatial map) makes catching issues during model development very easy.

That being said, we agree that these convenient aspects of *slendr* were not highlighted enough in the manuscript. We have now expanded upon this topic in a section of the manuscript which discusses gene-flow events, making it clear that the simple examples shown do not represent the limit of what *slendr* can do in this regard. At the same time, we also point the reader towards examples in the official *slendr* documentation which show examples of programmatically generated models involving large gene-flow networks (lines 455-460).

> *How many populations can be realistically considered with slendr?*

As mentioned in our response to the first comment, *slendr* can simulate demographic histories involving an arbitrary number of populations. In this regard, it is not more limited than standalone SLiM or *msprime* because it internally uses these two frameworks for the simulation functionality and does not impose further restrictions on the size of a model. An updated sentence in the discussion section mentioned above now makes this point clearer (line 772).

> *More importantly, if there are a few tools to simulate spatiotemporal genetic data, there is especially a lack of equivalent tools to simulate genomic data, which is becoming the standard of data produced in the literature. The problem lies mainly in the computing power needed to generate data at the genomic scale jointly to complex population dynamics models. slender aims to fill this gap, but I think that this manuscript is missing one major information useful for the potential users, which is the computation time needed to simulate the example scenarios. Giving examples of computation times will allow the readers to get a better feeling about the potential applications of slendr. Being able to simulate complex models is great, but if the computation time does not allow for a satisfactory exploration of the parameter space, especially for genomic data, this can be a strong limitation.*

The reviewer correctly points out that it is very hard – even for expert users – to predict the computation runtime of population genetic simulations. We also agree that the examples in our paper (examples designed specifically to demonstrate *slendr*'s features on use cases approximating potential research situations) did not provide information needed for the reader to

make informed decisions about performance. To fix this issue, we have now added run times of each code example, as measured on a fairly powerful but relatively common personal computer at the time of writing (2021 model of the 16'' MacBook Pro powered by the M1 chip). You can find the runtimes at the end of the caption text belonging to each of the four code examples (lines 488-490, 527-529, 575-577, 652-654).Thank you for the suggestion.

Reviewer #2 (Liisa Loog)

> *Martin Petr and colleagues present an R package -slendr- for generating and analysing simulated genomic data under spatiotemporally explicit demographic scenarios. More specifically, the framework provides a single easy-to-use front end that integrates with widely used, powerful and flexible genetic data simulation and analyses frameworks – SLIM, msprime and tskit. As such, the slendr package has great potential for simplifying and facilitating population genetics tool development and testing, with a much-needed functionality for explicitly incorporating spatial aspects into demographic models. Due to its single interface implemented in a popular R scripting and statistical analyses environment, the slendr package has the potential to make the field of computational population genetics more accessible to researchers and students with little computational or analytical background, as well as, improving overall reproducibility of research in the field.*
>
> *The presented description of the key workings and key features of the package is clear and concise. The authors also provide examples of varying complexity, as well as links to external sources of further description, guidance and help, including a dedicated webpage for the R package.*
> *I am not able to provide any comments on potential errors in the code as I have not extensively tested the described package or familiarised myself with the underlying code. However, the open-source nature of this software facilitates efficient flagging (and fixing) of any problems by the user-community. The package is also already part of the CRAN R package repository.*

Thank you for your kind words and for recognizing the importance of community-driven open-source development and the impact it has for reproducible research. Indeed, since its first public release, *slendr* has been benefiting from users reporting issues, suggestions, and fixes to the software and we hope that its future developments will involve the community at even greater extent in planning and prioritizing new features.

> *My main concern is that, while it is high time for a framework that would allow researchers with various degrees of analytical background to explicitly simulate and consider spatial factors affecting patterns of genetic variation, these tools (when used for model comparison for demographic inference, as also proposed by the authors) present a great opportunity to introduce (implicit) biases that are not easy to detect without proper statistical controls. (Similarly to the frequent bad practices in use of agent-based modelling in social sciences.) This issue of special concern here because human population history research is of elevated popular interest and frequently (mis)used by groups with strong political agendas. To that end it would be great if the authors could elaborate on the discussion of the basic requirements for the downstream use of simulated data for model comparison and model parameter estimation that would incorporate (1) an emphasis on including a realistic null-model, (2) exploring a wide range of demographic scenarios and (3) performing hypothesis testing by formal model comparison (e.g. using AIC within the Approximate*

Bayesian Computation (ABC) framework), as well as, as providing examples of best practices and/or citing research with some theoretical discussion on the topic.

Thank you for highlighting this extremely important issue. In hindsight, our discussion of the promise of *slendr* for simulation-based inference in the submitted version of the manuscript has likely painted an overly naive picture about the potential complications involved in simulation-based inference using *slendr*, particularly those involving complex histories of spatio-temporal population dynamics. Inference of geographically-explicit models is an extremely difficult problem. As you pointed out, even a relatively simple question such as "What is a good null model for testing a hypothesis of a 2D population polygon-based range expansion across a landscape?" is unlikely to have an easily defined answer given the number of spatial parameters in question (the shapes of population ranges over time, dispersal and mating distances of individuals, etc.), all of which are expected to impact genetic diversity in the data. Because the original paragraph in the discussion did not acknowledge this sufficiently, we have now completely rewritten the discussion dedicated to the promise of inferences of complex models. Now spanning lines 691-726, the text should paint a more realistic picture of the current possibilities, future prospects and expectations, as well as potential pitfalls.

Briefly, the new section first (in paragraph on lines 691-705) discusses the attractive properties of *slendr*'s simulation and tree-sequence R interfaces as building blocks which make simulation-based methodologies such as ABC significantly easier to write and more reproducible. The paragraph in question focuses on *slendr* as a programming tool alleviating the many technical challenges involved in ABC inferences, highlighting this as an area where *slendr* can make an immediate impact on traditional ABC workflows (in fact, we have just begun developing a new "spin-off" R package which will spearhead this effort in the future: https://github.com/bodkan/demografr).

Following this programming-focused discussion, a separate paragraph (lines 706-726) now brings up the statistical and philosophical challenges of fitting complex geographically-explicit models, including the example of the seemingly simple "null-model" problem for spatial modeling in a continuous space. We also explicitly state the large multidimensional parameter space of spatial models involving often unappreciated (and in non-spatial models often unaccounted for!) parameters, all of which play a crucial role on genetic patterns which emerge from the data (dispersal and mating distance parameters being just one example). Therefore, rather than giving the false impression that fitting full spatial *slendr* models is a solved problem the way non-spatial ABC could be (at least conceptually), we now phrase the discussion in terms of the utility of *slendr* as a tool to help us understand the complexity of spatial modeling in general, to characterize the patterns that can be expected to emerge from the data through various combinations of spatial model parameters and, ultimately, aid in establishing the guidelines for best practice in spatio-temporal inference in population genetics in continuous space, moving forwards. With regards to this point, we are not aware of extensive literature on this relatively cutting-edge topic but we now provide a reference to your own paper in Philos Trans R Soc Lond B Biol Sci. (2021) (line 722) which provides a useful and concise overview of the multifaceted nature of fitting complex demographic models.

Reviewer #3

> *I was pleased to be offered the opportunity to comment on this preprint, having been excited by its initial release. The manuscript is very well-written, and I am convinced that slendr succeeds at what it sets out to achieve; principally improving accessibility to complex population genetic modelling. The target audience for slendr is molecular ecologists, who as the manuscript highlights are often familiar with R as opposed to python, bash and in particular Eidos (the bespoke SLiM language). The package also provides a range of general functions for interacting with tree-sequencing outputs within the R environment that will likely have widespread interest for anyone using SLiM and/or msprime. I am confident that slendr represents an important addition to the ever-expanding SLiM/msprime ecosystem, and it is reassuring to see that this manuscript is co-authored by the developers of SLiM and msprime. In general, I found the manuscript did an excellent job in justifying the case for slendr and documenting and demonstrating its functionality. I have a few minor comments that I hope the authors find insightful, but otherwise commend their excellent work and look forward to making use of slendr in the future.*

Thank you for the kind words on our work and for appreciating our effort to make population genetic modeling accessible to a wide audience of R users in ecology and evolutionary biology.

> *Coming from the perspective of someone who has worked with SLiM in the past, something that I felt could have been expanded on in the manuscript was a more explicit discussion of functionality that is not available. The manuscript does a good job of explaining functionality that is available, which for those with limited experience of simulation is useful, however at times I was unsure of whether simulations I've run in the past would be possible with slendr. For example, defining mutation types, genome element types, recombination landscapes in the initiate phase, or defining fitness and mating callbacks. It might be interesting to know which of SLiM's recipes are reproducible in the slendr framework to further provide a sense of what is and isn't possible for those who have worked with SLiM previously. Expanding on these would be beneficial for those readers who wish to take advantage of slendr's more general functionality, i.e. improved reproducibility, whole analyses contained within R, interacting with tskit etc.*

Thank you for pointing out this omission in our manuscript. As we acknowledged in our responses to another reviewer's comments, it is clear that a discussion on the topic of where *slendr* differs (and where it matches) the capabilities of pure *msprime* or SLiM scripts was missing from the first version of our manuscript. This is now elaborated on in a new section titled "Relationship of slendr to SLiM and msprime" (lines 393-422) as well as in a new paragraph in the discussion (lines 782-791), rectifying the situation.

To answer your particular question directly here (but the following is discussed also in the additions to the text just mentioned), the current version of *slendr* is indeed limited to neutral simulations with a uniform recombination landscape. In the context of the SLiM back end implemented in *slendr*, this implies a single type of neutral mutation type, uniform recombination rate, etc. That said, we are certainly planning to unlock the possibility to customize the genomic and mutational landscape of simulations in one of the future versions of *slendr*. We have been gathering feedback and suggestions from the community on this topic (both in terms of facilitating this from SLiM's side https://github.com/MesserLab/SLiM/issues/309 but also in terms of expanding *slendr* itself https://github.com/bodkan/slendr/issues/96), and are working on a roadmap for expanding *slendr* in this regard in a future version.

An interesting avenue we are currently exploring for supporting the above mentioned features is to keep the current behavior of *slendr* as the default (i.e., keeping the *slendr* R interface as simple as possible), but provide an option to modify this default by "injecting" user-provided SLiM snippets at appropriate places in the SLiM back-end code of *slendr* (one possibility is discussed here https://github.com/MesserLab/SLiM/issues/309#issuecomment-1111590907). This would avoid complicating the basic workflow for novice users, while allowing simulation experts to customize their simulation code by injecting small snippets of SLiM code where needed without having to leave the convenient environment of R interactive sessions.

It is difficult to predict what would such expandable interface of *slendr* look like which is why we avoided discussing the possibilities of this extension in technical detail in the manuscript. However, a brief note about the possibility of this extension is now added in the discussion section of the manuscript so that the readers are aware that such features are planned in the future (lines 782-791).

> *Related to the above, my take-away from the manuscript was that the absence of mutation types and fitness callbacks limits slendr to simulating neutral models (with the exception of fitness through competition). If this is the case it may be worth mentioning briefly at some point in the discussion.*

Your takeaway is correct. As we write above, a new section has been added to the manuscript (lines 782-791), discussing the current limits of specifying the mutational and genomic landscape of SLiM-based *slendr* simulations, and some potential outlooks to remove this limitation in a future version of *slendr* (this is a high priority item on our roadmap).

> *I was curious whether there were any be benchmarking issues incurred as a result of running slim simulations through slendr as opposed to as standard on the command-line. Does it scale similarly as users increase pop size, mutation/recombination rate?*

Interestingly, when our `slim()` function is called in R, the SLiM back end script bundled with *slendr* is executed on the command line in exactly the same manner as a normal SLiM script would be. In fact, each compiled *slendr* model is stored in a standalone "project folder" on disk which includes a copy of such a SLiM script, and can be run from the command-line outside of R, without *slendr* even being present on the system. As such, users do not have to worry about performance hits of *slendr* simulations as opposed to running SLiM using the standard command-line workflow—when it comes to simulations in *slendr*, both are effectively one and the same.

To demonstrate the above in more detail, here is a relevant piece of the *slendr* R codebase which first composes a command-line SLiM command as a plain R string (code below starting from https://github.com/bodkan/slendr/blob/13133ed9cc51a31fe1dc6f10a93909754d825068/R/compilation.R#L456):

```
slim_command <- sprintf("%s %s %s \\
    -d 'MODEL=\"%s\"' \\
    -d 'OUTPUT_TS=\"%s\"' \\
    -d SPATIAL=%s \\
    -d SEQUENCE_LENGTH=%s \\
    -d RECOMB_RATE=%s \\
    -d BURNIN_LENGTH=%s \\
    -d SIMULATION_LENGTH=%s \\
    -d 'OUTPUT_LOCATIONS=\"%s\"' \\
    -d COALESCENT_ONLY=%s \\
    -d MAX_ATTEMPTS=%i \\
    %s 2>&1",
  binary, # path to the SLiM binary on the command line
  seed,
  [ … additional arguments to be passed on command-line using -d ]
)
```

This command-line string is then executed in the background via this R function call (line 493 on the GitHub link provided above):

```
system(slim_command, intern = TRUE)
```

which internally calls the SLiM command on the standard unix command-line as

```
slim -d <path to model directory> [-d <additional command-line arguments as above> …]
```

Therefore, as with any other standard SLiM simulation, modifying parameters of a *slendr* model (such as recombination rate) is simply passed to the SLiM binary via an appropriate command-line argument. The R `slim()` function only acts as a convenient interface for a standard SLiM command-line call, without additional runtime penalties.

In addition to this, does the slendr framework lend itself to running simulations in parallel across multiple CPU, or would this be handled with general R parallelisation e.g. running models within doParallel or similar?

There are tentative plans to implement automated parallelization in *slendr* itself (notes and discussion with the community on this topic are being tracked on GitHub https://github.com/bodkan/slendr/issues/88). However, due to technical challenges with implementing this feature in a truly platform-independent way (macOS, GNU/Linux, Windows), we have not yet started implementing it. As such, parallelization must be taken care of with available R features, such as those highlighted by the reviewer.

Based on this comment, a brief note on this topic is now included in the discussion section of our manuscript, mentioning the current options using standard R parallelization features (lines 702-705). This will certainly help novice users who are not familiar with R's powerful parallelization packages to speed up their analyses. Thank you for the suggestion!

Why would a pre-requisite to spatial raster models have to necessarily involve non-WF functionality (line 659)? Presumably a WF-model that could describe the expected spatial population structure observed when a WF population's dispersal or mating is limited by local environment would be useful for various applications of interest to molecular ecologists, including connectivity for example. I should add that this section (line 653-671) does a good job overall of highlighting future applications.

Our plan for adding an optional non-WF-based mode for *slendr* simulations to a future version of *slendr* has been motivated by our aim to take into account the habitability of each location (i.e., with each pixel of a raster map directing how likely are individuals to survive at that location), which would dynamically regulate population abundance via carrying capacity of the population range. This has been inspired by the techniques shown in Chapter 16 of the SLiM manual (http://benhaller.com/slim/SLiM_Manual.pdf, version 4.0.1) whose introduction section states that

"[...] in nonWF models population regulation is a consequence of the balance between individual reproduction and individual mortality, just as it is in natural populations, rather than being enforced through a set population size as in WF models, and so the model typically needs to treat population regulation explicitly [...]".

Therefore, rather than modeling a population with a fixed population size parameter N in each time point (i.e., a WF model), population size would be a more complex function of the habitability at each point of the raster, with the total size of the population being an emergent property of the habitability metric aggregated across the map. This is what we meant by the requirement of SLiM's non-WF mode of operation.

We have clarified our intentions with this future extension of *slendr* in the discussion (lines 729-758).