Dear Dr Bulanadi,

Once again we extend our thanks for considering our paper for publication, and for your and the reviewers expert comments, which will improve the final form of our publication.

As before, we have addressed all the concerns and indicated them clearly (in red) in the following letter. Markers to specific changes are enumerated in [violet] within the manuscript, and highlighted in red.

An unmarked revised manuscript is also included for publication purposes.

Kind regards

Monika Mozere, Mehmet Tekman, Jameela Kari, Detlef Bockenhauer, Robert Kleta, and Horia Stanescu.

Reviewer reports:

Reviewer 2: The authors did a good job in rewriting the implementation section. The functional description of the pipeline module is much more readable and informative, and also highlights the novel aspect of the inheritance filtering. The authors might want to adapt their abstract accordantly.

We have now better emphasized the inheritance-modelling focus of our pipeline within the abstract (page 1, [1,2,3]).

The technical and functional comparison of OVAS to other annotation tools is much appreciated as it also highlights the functional novelty of OVAS in the inheritance filter. The results on the case studies are still only shown for OVAS however, so a full comparison is still missing. At least the most relevant alternative tools should be run on the same test data cases shown in Figure 4 and Supplementary Figure S1 to support the conclusion that only OVAS is able to filter those variants.

Table S1 presents 6 workflows for variant analysis, of which only Galaxy and Taverna are similar to OVAS in terms of the target end-user and their filtering capabilities.

We considered producing a side-by-side filtering module comparison of these workflows against OVAS, but realised that the standard filtering modules (such as filter on variant position, quality control, confidence level) are by design engineered to produce an expected output, rendering a comparison of these void.

In contrast, the more sophisticated filtering modules such as the inheritance-modelling as well as the alternative allele filtering, did not exist in the other workflows and hence no directly applicable comparison of these modules could also be made.

Thank you for explaining the motivation against using existing frameworks such as Snakemake. The authors might find it more flexible to use Bash scripts, but the downside is already obvious: a static, bootable OS is needed, and I am still not convinced that this is a better alternative. All other software that one would want to use in parallel will be missing on this OS for example.

The end-user can still install temporary additional software onto the live system should they wish to (and should they have an active internet connection), but your point is well taken.

As a minor side note, I feel it should be mentioned that we did initially experiment with running OVAS upon a writeable USB medium that would preserve package installations, but experienced technical stoppages related to memory / paging issues and abandoned this approach in favour of a smoother user experience.

Also, all points stated on p.6 [9] in favour for Bash scripts over especially Snakemake are also true when using Snakemake: it is very well unix-driven, and its shell command supports any linux commands including streaming or calling external tools/programs in whatever language, it has a re-entry concept and checks for already processed files by checking for existing output of rules. Also, command line wise, using bioconda to package tools and handle external software dependencies is more and more adapted.

My point is, I believe that relying on the current bundle of scripts largely reduces the number of bioinformaticans using OVAS. So it should be in the authors' interest to consider frameworks such as Snakemake - at least in the future. However, I am willing to let this point go and accept the current stage of OVAS.

Thank you kindly for the suggestions, they have been taken to heart and we are now in the process of incorporating the annotation components (funcannot and genepender) of our pipeline as new recipes within Bioconda (mentioned on page 7 [4]). This will hopefully be of good use to bioinformaticians working within already established workflows such as Snakemake, and enable us to update our work for the benefit of an increasingly more commandline-oriented audience.