Dear Dr Bulanadi,

Thank you for considering our paper for publication, and for your and the reviewers comments, which will improve the final form of our publication.

We 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.

Editor Comments:

There are a number of concerns regarding your manuscript and contribution to the field. If you choose to resubmit your work, please ensure that you include a full comparison against existing tools (the most often used and the newest tools), and please emphasise the contribution of your work.

We have now added a section 0.4 to address this (page 6, [10]), with comparisons between the most highly cited or widely used bioinformatic workflows and pipelines. Standalone variant annotation tools are also compared with the annotation stage of our pipeline.

Two tables have also been added to the Supplementary Data (page 2) detailing the comparison specific for both the overall pipeline and variant annotation component.

Reviewer reports:

Reviewer 1: Overview

This paper described a new variant analysis tool. It consists of the basic components of variant analysis pipeline, including annotation/mapping and filtering. Recent advances in genome sequencing technologies have led to a thriving development of analysis tools. Hundreds of variant analysis tools are available from previous researches. This tool (OVAS) identifies a subset of variants of interests under an inheritance context. From the aspect of implementation, it emphasizes on transparent methods and in-situ service.

Detailed Comments

This is a research paper, which should focus on the methods and experiments, but the author described the software and web interface in too much details. For example, section 0.1 could move to the supplementary file or the README file for the software.

Indeed, section 0.1 has been rewritten to provide a more concise overview about the implementation, as well as the overall modular components of the pipeline (pages 3-4, [4,5]). The Supplementary Data file now provides a more detailed implementation of the pipeline (page 3), and a thorough module-by-module description of the overall pipeline (pages 4-5).

The author made good arguments about cloud-based services. But I suggest the author to build a web service as well to benefit users by accessing the internet, as the majority of users will have access to the internet. A simple scenario is that the user wants to test their data by typing some parameters in the web browser before actually using it, and doesn't want to go through the complicated installation process on their local computer.

Hosting OVAS as a global web server is readily achievable (see the main software repository for deployment specifics), but undermines the overall ethos of the pipeline as a secure standalone bootable environment. It is also not practical for modern sequencing data (see page 6 [7]).

In the results section, the author showed one example about HIPKD, and didn't put much discussion on the trait-penetrance modeling as it highlighting in the title. It would be more convincing to show more examples to support the conclusion.

After much consideration, a second and third case study were added to the results that elucidate the inheritance modelling feature for autosomal and X-linked families (see page 5 [6] and Supplemental Data file (page 1)), and a paragraph was added to the conclusion (page 7 [11]).

Reviewer 2: Summary

Mozere et al. present an open-source variant analysis suite called OVAS. From the description it is not clear to me what the novel features and contributions of this analysis suite compared to the plethora of existing ones is here. The main contribution seems to be that this suite is a standalone pipeline and is operatable offline compared to cloud-based tools. This, however, is achieved via a complete, bootable system distribution, or - alternatively - via extensive system requirements. The authors say that this "can provide a safer and more accessible environment […] especially in circumstances where continuous internet usage is limited". This might be true for less extensive data where one does not need a cloud or larger servers. Still, I do not see the benefit compared to existing non-cloud-based pipelines.

The discussion (page 6 [7]) better reiterates the case for more accessible bioinformatic tools, and cites a recent paper highlighting the difficulties of running genomic analyses in Africa. A comparison of non-cloud based pipelines has also been included in the section following (page 6 [10]).

Major comments

[Content/contribution]

Although the authors are very detailed in the description of their implementation, they are not precise in the functional description of their tool.

The implementation section has been altered at the behest of reviewer#1 to provide a more functional overview of the pipeline, with more detailed implementation-specifics moved to the Supplemental Data (pages 3-5 [4,5]).

For example, p.2 the authors say "OVAS is rooted firmly in trusted public domain databases such as RefGene, dbSNP, UniProt […] UCSC Genome Browser." Do the authors use these databases within their pipeline? Do they use the same algorithms?

UCSC actively curates and collates data from upstream databases, maintaining an excellent repository of data consolidated in a single extremely accessible database. It is solely this data that OVAS processes during the annotation stage. The scripts/algorithms used to the maintain the databases are of no interest to OVAS, and this has been better stated in the text (page 2 [2]).

Why do we need their tool, if there is a "beneficial accordance"?

Though UCSC does indeed host a repository of bioinformatic utilities, unfortunately none of these of these have ever been equipped for variant analysis.

As of 2016, they do now provide a “Variant Integrator” tool which performs variant annotation to some degree, but it is not yet fully-developed and is not at all designed to be included within a variant analysis workflow, as discussed in the main text (bottom paragraph of column 1, page 3 [4]).

A comparison of variant annotation tools is also provided in Table S2 of the Supplemental Data (page 2).

Another example, on p.4 "The annotation stages of the pipeline then prime the variants with relevant metadata […]". What metadata, i.e. is it user provided or is it a certain database? Especially since the authors then say that "the annotation stage is the only mandatory stage, and a great portion of filtering occurs at these stages too, with up to 90% of true negatives being discarded." Based on what are up to 90% TN discarded? Overall, I am not able to determine the functional meaning of the modules and hence, the contribution of the tool to the user.

As mentioned in the Pipeline Overview (page 3 [4]), OVAS initially annotates variants within a VCF file with gene-context data.

For example, if variant *X* falls within *Exon1* of gene *ABC123* – then variant *X* will then have metadata pertaining to *Exon1* of gene *ABC123* applied to it.

This metadata is sourced from the UCSC RefGene table, within the database pertaining to whichever human genome build the variants were aligned against during upstream variant alignment/variant-calling.

OVAS also applies metadata calculated from the downstream functional changes that would arise from the mutation acting upon the gene. OVAS is not limited to purely coding changes and handles regulatory changes such as introns, splice sites, UTR, and promoter regions.

However, some variants will not fall within any gene-context such as those which lie within purely non-coding / non-regulatory (i.e. “wholly intergenic”) regions. These variants are completely uninformative for variant analysis and are therefore filtered out during the annotation process.

Considering the small percentage of the genome that is actually functional, it is to be expected that the vast majority of variants within input VCF files will be filtered out. This holds true not only for whole-genome sequencing but also exome-capture sequencing too, as evidenced by Figure 4 and Figure S1.

[Comparison to other tools]

Unfortunately, a comparison to other tools is completely missing. Since there are existing tools/pipelines (205 according to Pabinger et al.), I would expect a comparison, even if they are cloud based? Without this, the contribution of the tool to the large community is not assessable.

The discussion now provides a section (page 6 [10]) dedicated to the comparison of other open-source pipelines, including cloud-based. A wider range of non cloud-based tools are compared in terms of the four main key deliverables that OVAS offers: transparency, security, deployability, and inheritance modelling (see TableS1 in the Supplemental Data file, page 2).

[Usability/Dependencies/Software]

The usability of the tool is also questionable:

The authors say that their tool is a standalone/offline pipeline but then write "Users can upload their data either through the web-interface or by manual file placement […]". This sentence is irritating since it implies that I would really upload data to some server through the internet. Since I have to place my data "somewhere" in my system to be able to "upload" it, I can just directly place it. I do not understand the meaning behind this or of what use this would be.

As mentioned in section () it is is indeed possible to place files somewhere on your system and then initiate the pipeline from the commandline, however most end-users would find this process convoluted and difficult.

The OVAS web-interface therefore provides a more convenient abstraction of this underlying process, using familiar upload/download dialogs to assist in the background placement of these files without the user having to worry about the filesystem specifics. They are also accurate depictions of how the web-interface interacts with the web-server.

An ‘upload’ to the web-server only results in a movement of a user-provided file to a private location more convenient for the web-server via the standard HTTP protocols required for the browser to communicate with the server. A ‘download’ operates in much the same way; dispatching the file from a private location on the server, and offering it to the user within the web-interface.

Though a web-server is indeed more traditionally based on a remote server, OVAS’s web-server is locally deployed on the user’s machine and contained entirely within the OVAS live environment such that data does not leave the user’s machine at any stage.

This is now better clarified in the main text (page 2 [3]).

The authors use "a series of inter-connecting Bash shell scripts which serve as necessary framework to accommodate wrapper […]". There is a reason why sophisticated frameworks such as Snakemake have been developed (I recommend the review by Leipzig - A review of bioinformatics pipeline frameworks): (Bash shell) scripts are often not robust enough when change of computing environment or updated tool/package/bash versions come to mind. I find the use - especially of bash scripts - highly critical in published bioinformatic software.

This is an excellent review, and a few of the suggested frameworks were initially considered but a difference in design philosophy meant that OVAS opted for a more script-based framework in order to minimize the need to install third-party software, as well as a need reduce the number of dependencies required to run an analysis. This is discussed extensively in Section 0.3 (page 6 [8,9,10]).

It should be noted that other highly-cited pipelines such as HugeSeq and GATK are also script-based (please see Table S2 in the Supplemental Data, page 2).

Specific remarks

p.2 "Optimistic filtering measures will produce a smaller set with the drawback of missing key causative variants, and conservative filtering measures will produce too many false positives." I believe this should be the other way around, usually being conservative means to be more specific and hence have less false positives. There are many words that do not make sense or do not fit the context, e.g. p.3 "groundwork, "programmatically", "overridden".

The initial specific remarks on page 2 have been removed when then section was re-written (page 2 [2,3]).