Response to referees

We would like to thank the referees for thoughtful comments. We believe addressing the questions raised by the referees has substantially improved our manuscript and software. The point-by-point reply is as follows.

1 Response to comments from Reviewer 1

1. Authors have included a description file and a README file, but there is no citation file. One should be included.

   We have updated the package and included a citation file.

2. Authors may want to consider restructuring their package to include the released Python source code of SmartVA-Analyze with the reticulate package so the shiny app and R package are the same.

   We agree that including the Python source code in the R package would make the openVA software eco-system more consistent. However, we have not pursued this in the current version due to practical reasons: The authors developing the SmartVA-analyze software are not part of our team and have not collaborated with us in producing the openVA software. Thus it creates more uncertainty on timely updates and maintenance of the package if we included their source code. For example, SmartVA-Analyze is implemented in Python 2, which is no longer supported by the Python Software Foundation (and thus not receiving security updates). There has been no indication that the codes will be updated to Python 3 from its developers. Unlike the shiny app, packages on CRAN are routinely checked with strict rules. Without the support and commitment of the original authors, it is difficult to maintain the R package to the standard of CRAN while keeping it to be a faithful replication of the SmartVA-analyze. Therefore, we have considered this option but have not had the bandwidth to pursue it further.
   However, it is our hope that the effort in the current openVA framework will eventually lead the VA community to become more transparent and collaborative, so that future developments of methods and software can avoid this type of challenge.

3. At the top of page 4, when reviewing the InterVA4 and InterVA5 algorithms, authors have written “where both the prior distribution of each the causes, $\pi_k^{(0)}$.” It seems like something is missing from this portion of this sentence.

Thank you for catching the error. We have updated the sentence to be

> "where both the prior distribution of each of the causes, $\pi_k^{(0)}$ and the conditional probabilities $P(s_{ij} = 1|y_i = k)$ are fixed values provided in the InterVA software."

4. At the top of page 4, when reviewing the InterVA4 and InterVA5 algorithms, authors may consider adding a reference for readers to explain why the algorithm omits the probability that any symptom is absent.

   We have added the following sentence that contains reference to a detailed discussion of the algorithm choice of omitting the absent symptom calculation.

   > " A detailed discussion of this modeling choice can be found in McCormick et al. (2016). "

5. On page 4, when reviewing the InSilicoVA algorithm, authors discuss the hyperprior on $P(s_{ij}|y_i = k)$ but not on μ and σ^2. Give a reference or provide those.

   We added the following description on the hyperpriors

   > " The priors on μ and σ^2 are diffuse uniform priors. "

6. At the top of page 6, the sentence begins "The openVA package internally cleans up...", and it isn't clear which function cleans up the PHMRC data. Authors should consider adding clarity if it is in the getPHMRC_url() or codeVA() function. It is mentioned later on page 8 that data transformations occur in the codeVA() function, but this was not clear when it was initially discussed.

   We have replaced the sentence on page 6 with the following

   > " The **openVA** package internally cleans up the PHMRC gold standard data when calling the `codeVA()` function on the PHMRC data. "

7. At the top of page 6, there is a typo: "... that the columns are exactly the same sas the PHMRC gold standard datasets..."

   We have fixed the typo.

8. In multiple places, documentation for data and functions from the integrated R packages (InterVA4, InterVA5, InSilicoVA, Tariff, and nbc4va) does not appear in the help files. For instance, the dataset, RandomVA1, has no help documentation in the openVA package documentation. Authors should re-export data to redocument in their code.

   We have created a re-exported function entry in the help file that help users better identify the key functions in the dependent packages. However, we have not found a good way to document data from other packages without creating duplicate data files. While we acknowledge that it is not ideal that the help file does not contain an entry for these datasets, we expect users to be able to reach the help files of the data objects simply by '?RandomVA1' for example.

9. With the codeVA() function help page, authors need to update it to include the information about data transformations being performed automatically with the function. It isn't clear unless the user has read this journal article.

   We have added the information on the `codeVA()` function help page.

10. At the bottom of page 10, when the authors utilize the stackplotVA() function with user defined order, the authors use "order.group = order.group" which does not match help. In fact, Figure 2 and Figure 3 are the exact same. The causes in Figure 3 are not reordered.

    We have fixed this error in the package and the paper.

11. Plots are not colorblind friendly. Changing to black and white is not enough. Authors need to update plotting code with ggplot2 extensions (https://exts.ggplot2.tidyverse.org/). Additionally, authors need to use double encoding in their visuals. References for double encoding found at https://www.annualreviews.org/doi/10.1146/annurev-statistics-031219-041252 and https://www.tandfonline.com/doi/full/10.1080/10618600.2016.1209116

    We thank the reviewer for this highly useful suggestion. We have updated all the default colors using a color blind friendly palette.

12. At the bottom of page 12, authors utilize the stackplot() function, and earlier they used the stackplotVA() function. Is there a reason for the switch or can we still use stackplotVA()? Additionally, the documentation for both functions is identical. If we can not use them interchangeably, documentation needs to be clearer.

    We have removed the reference to `stackplot()` and use `stackplotVA()` throughout. The differences between the two functions is that `openVA::stackplotVA()` is an extension of `InSilicoVA::stackplot()` that can be applied to fitted objects from other algorithms other than InSilicoVA. We apologize for the lack of clarity in the paper and document. We have updated the **openVA** document on this function with the following description:

    > "Produce bar plot of the CSMFs for a fitted object in broader groups. This function extends the stackplot() function in the InSilicoVA package to allow for the same visualization for results from InterVA, NBC, and Tariff algorithms."

2 Responses to comments from Reviewer 2

In regions where full autopsies are impractical, verbal autopsies (VA) can be performed by trained fieldworkers. The fieldworkers learn demographic information and symptoms experienced from a deceased person's family or caregivers. The World Health Organization (WHO) and the Institute for Health Metrics and Evaluation (IHME) created VA questionnaires that are similar but not identical. People who work with VA data typically want to accomplish two goals:

- Assign a cause of death (COD) to an individual for whom a VA was performed
- Estimate population-level cause-specific mortality fractions (CSMF)

Five R packages – InSilicoVA, InterVA4, InterVA5, Tariff, and nbc4va – have been created over the years to accomplish these goals. Each package implements a different statistical model and has its own capabilities and drawbacks. The openVA package makes it easy for the user to run the models from each of these packages and compare the models' performance on the same VA dataset.

Without the openVA package, if a user wants to compare different VA models in R they need to run different R packages each with its own data format requirements and functions. The openVA package makes it easier to compare models by implementing a standard data format and a common set of functions. Because the package is specific to VA data, I think it would have limited use outside the VA community.

The approach is straight-forward. The openVA package has helper functions to transform data into a standard format. Then a common set of functions can be used to fit the different models and summarize the results. The plotting functions also work for each of the models.

The article gives sufficient background on the topic and discusses some alternative methods, namely using one of the five VA packages listed above individually. The authors also include functionality that they would like to add to the package in the future. The examples are clear and helpful except for the code right before the Individual COD Summary subsection. I didn't understand what this code is doing.

Overall, the article is clear and concise. The following minor things were confusing or unclear to me:

We thank the reviewer for the overall positive feedback and the suggestions. We address the specific points in the following.

1. Because I am unfamiliar with VA, I would have found it helpful if the authors gave examples of regions or locations where VAs are commonly used.

   We have added the following sentences in the first paragraph to give more background on the use of VA:

   > " VAs are routinely used both by researchers in health and demographic surveillance systems (Maher et al. 2010; Sankoh and Byass 2012) and multi-country research projects (Nkengasong et al. 2020; Breiman et al. 2021), and in national-scale surveys in many low- and middle-income countries (LMICs). For a more comprehensive overview of the current use of VA, we refer readers to Chandramohan et al. (2021). "

2. In the line of code directly before the start of the Overview of VA cause-of-death assignment methods subsection, the authors use set.seed(12345) but don't explain why they are setting the random seed. The subsequent lines of code in this section don't use the random number generator. I think it would be helpful for the authors to specify which functions use the random number generator and set the seed closer to those functions in the paper.

   We have added the following description before the `set.seed()` line:

   > " Since posterior inference is carried out using MCMC in the InSilicoVA algorithm, we set the seed for the random number generator to make the paper reproducible. "

3. The authors mention "missing" and "absent" data in the last paragraph before the Data Preparation section, but they don't explain what they mean by these terms until later in the paper (the last paragraph in The WHO Standard Format subsection).

   We have moved the explanation of missing and absent indicators to the paragraph before the Data Preparation section. It now reads:

   > " In addition to the different model specifications underlying each algorithm, there is also a major conceptual difference in handling missing symptoms across the algorithms. Missing symptoms could arise from different stages of the data collection process. For example, the respondent may not know whether certain symptoms existed or may refuse to answer a question. From a statistical point of view, knowing that a symptom does not exist provides some information to the possible cause assignment, while a missing symptom does not. Although in theory, most of the VA algorithms could benefit from distinguishing 'missing' from 'absent', InSilicoVA is the only algorithm that has been implemented to acknowledge missing data. Missing indicators are assumed to be equivalent to 'absence' in InterVA, NBC, and Tariff. "

4. The first sentence in The PHMRC Format subsection begins "The Population Health Metrics Research Consortium (PHMRC) gold standard VA data..." Why are the authors referring to this data as the gold standard? Also, the first paragraph in the Fitting cause-of-death assignment models section mentions "gold standard labels" but does not explain this term.

   We echo the reviewer's concern about the dataset being branded as the 'gold standard'. The term 'gold standard' is the formal name of the dataset as published by original authors (`https://ghdx.healthdata.org/record/ihme-data/population-health-metrics-research-consortium-gold-standard-verbal-autopsy-data-2005-2011`. the While the VA literature contains different opinions on the usefulness of the dataset as a 'gold standard', we prefer to refer to it using its full published name. We have added the following sentence in this subsection to clarify how the gold standard are obtained:

> “ All deaths occurred in health facilities and gold-standard causes are determined based on laboratory, pathology and medical imaging findings. ”

To avoid confusion, we changed the other mention of ‘gold standard labels’ into ‘cause-of-death labels’.

5. In the second paragraph in The PHMRC Format section, the authors write “The openVA package internally cleans up the PHMRC...” Which function cleans up the PHMRC data or where in the pipeline is the data cleaned?

   We have expanded the sentence into

   > “ The **openVA** package internally cleans up the PHMRC gold standard data when calling the `codeVA()` function on the PHMRC data. The procedure follows the steps described in the supplement material of McCormick et al (2016). ”

6. The statement “We use the empirical distribution in the test data to calculate the true CSMF distribution and evaluate the CSMF accuracy using the getCSMF_accuracy() function” doesn’t give me enough information to understand how the true CSMF distribution is calculated.

   We have changed the sentence into

   > “ We use the empirical distribution in the test data to calculate the true CSMF distribution, i.e., $CSMF_j^{(true)} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{y_i=j}$. Then we evaluate the CSMF accuracy using the `getCSMF_accuracy()` function. ”

7. It’s unclear to me what the code right before the Individual COD Summary subsection is doing and why.

   The code snippet computes the CSMF accuracy defined above. We included this computation as this is a very common step in the data analysis pipeline among VA practitioners. While the computation is straightforward, we received many requests to make it possible to compute the metric directly, without users manually dealing with vectors in R. Thus we think it is useful to present the function here. We have reordered the text around this code snippet to make it easier to read. It now starts with:

   > “ The CSMF accuracy can be readily computed using functions in **openVA**, as shown in the following codes. ”

8. Figure 2 and Figure 3 contain the same plot. The causes are not reordered in Figure 3.

   We have fixed this issue in the revised paper.

9. Figures 4-6, 8 don't explain what the bars and the points represent

   We have added additional descriptions in these three figure captions such as the following:

   > " Estimated CSMFs for different sub-populations. The points indicate posterior medians of the CSMF and the error bars indicate 95% credible intervals of the CSMF. "

   and

   > " Aggregated CSMFs for four different sub-populations. Left: comparison using the stacked bar chart. Right: comparison using the bar chart arranged side to side. The height of the bars indicate posterior means of the CSMF and the error bars indicate 95% credible intervals of the CSMF. "

10. Overall, the package seems to be well organized with each script containing similar functions. The exported functions purposes and names make sense. The following items don't meet the guidelines described in https://devguide.ropensci.org/building.html:

    We thank the reviewer for the careful evaluation of the package source code and many great suggestions. We have made substantial changes to clean up the codes to adhere to the style guide as closely as possible.

    - Some helper function names start with a period. E.g. ".phmrc_adult_convert"

      We have renamed these functions and removed any names starting with a period.

    - Some exported function names contain periods. E.g. "interVA.train"

      We have renamed the function to interVA_train. We anticipate the change to have minimal effect on users since this function is mostly called internally.

    - Some argument names contain periods

      We acknowledge that the use of periods in the function arguments is not the recommended best style. However, for the existing functions in the openVA package, we feel the need to keep these arguments as they have been used extensively in many downstream applications where the package is deeply integrated within significant civil registration and vital statistics data pipelines. We are concerned that changing the arguments could affect the existing users negatively even if we allow backward compatibility of the previous argument names. This is particularly concerning considering many users of the package are not experienced R programmers and seeing two versions of the same argument in the document and their working R script (e.g., `data_train` and `data.train`) could lead to a lot of confusion and potential mistrust. Respectfully, we have not changed this aspect, but we do pledge to follow the best naming practice in our future effort more closely.

- Some functions do use warning() as recommended but the openVA_update() function uses cli::cat_line() instead of message()

  We removed the use of cli::cat_line() in the package and replaced them with message().

- There is no CITATION file

  We have added a citation file.

- The README has badges for R-CMD-check (passing), CRAN (1.0.14), downloads(518/month), and downloads(36K), but no badge for test converge

  We have included a test coverage badge.

- The README doesn't introduce or describe the package

  We have added an introduction that describes the package

- The README gives installation instructions and help but does not include any examples

  We have included a simple example in the readme file and also link to the vignette for more detailed examples.

- The README doesn't include info about or links to the other packages that it uses heavily

  We have added links to the component packages in the readme file.

- There is no vignette

  We have included the current version of the paper of the vignette in the package.

- The package uses ROxygen tags for exported functions, but it does not group related functions with the @family tag and the internal functions don't have ROxygen tags at all so the @noRd tag is not used

  We have grouped related functions into several main categories and added the ROxygen tags for the internal functions.

- The package only includes a handful of tests

  Thank you for pointing out the lack of unit tests. We have included additional tests in the package in the new version. Our test coverage increases from 25% in the last version to 75% in the current version. We will continue to improve our unit tests in future developments.

- Some functions have code that is commented out but not deleted

  We have removed the commented out codes in the package functions.

- Some scripts in the R folder are named ".r" while others are ".R"

  We have fixed this issue by making all scripts end with ".r"