Workshop & Algorithms

Python openVA Workshop
Dar es Salaam, Tanzania
August 8-10, 2023

Objectives

- Introduce Python openVA
- Train potential users
  - Inputs
  - Operation
  - Understanding outputs
- Obtain feedback that will help the openVA development team to improve Python openVA and related software

Thanks to our sponsors

- Vital Strategies
- CDC Foundation
- Bloomberg Philanthropies *Data for Health Initiative*
- Zambian Department of National Registration, Passport and Citizenship (DNRPC)

Organizers

- Vital Strategies - **Robert Mswia**
- openVA Team
- **Isaac Lyatuu**
- **Collins O’Chieng**

Workshop overview

Day 1

- Morning: Introductions and verbal autopsy algorithms
- Afternoon: Python openVA Software

Day 2

- Morning: Worked examples using Python openVA, including hands-on work
- Afternoon 1: Interpreting openVA results
- Afternoon 2: Dashboard tools for verbal autopsy

Workshop overview

Day 3

- Morning: Supervised hands-on work using Python openVA on your own data
- Afternoon: Feedback and advice for future openVA development

Pacing & breaks

- *First time we are doing this workshop*
- Don’t know how much time will be necessary for each section; therefore, no set times in schedule
- Breaks
  - Mid-morning coffee/tea: 10:30 - 10:50
  - Lunch: 1:00 - 2:00
  - Afternoon coffee/tea: 3:40 - 4:00

Introduction to verbal autopsy algorithms

1. Verbal autopsy algorithms
    a. Physician-coded VA
    b. Computer-coded VA
2. InterVA
3. InSilicoVA
4. Tariff

Verbal autopsy cause-coding algorithms

- Objectives of VA
  - Identify likely causes for individual deaths
  - Estimate cause-specific mortality fractions for population of deaths
- Components of VA
  - VA **questionnaire**
  - VA **interview**
  - VA **data**
  - Symptom-cause information (**SCI**)
  - Cause-classification **logic**
  - **Results: individual-level cause-specific metric** (**ICSM**) - propensities, probabilities, tariff scores, etc.
  - **Results: population-level cause-specific mortality fractions (CSMF)**

Verbal autopsy cause classification

- **This workshop focuses on automated cause-classification**

- Cause classification **applies classification logic to VA data using SCI** to calculate a ICSM for each cause for each death and CSMFs
- **Three elements required for classification**
  - **VA data**
  - **SCI**
  - **Classification logic**
- ***All three elements affect the classification***

Physician-coded VA - **PCVA**

- Physicians read VAs and identify likely causes
  - **VA data**
  - **SCI** is physicians' knowledge of relationships between VA indicators and causes
  - **Logic** is physicians' reasoning about VA data using their knowledge
- Can identify specific causes
- Time-consuming
- Potentially costly
- Usually slow
- Physicians often disagree ⇒ not replicable
- **Not useful for large-scale VA**

Computer-coded VA - **CCVA**

- Computer runs a statistical/computational algorithm
  - **VA data**
  - **SCI** is tabular data summarizing the relationship between VA indicators and causes
  - **Logic** is a formal statistical/mathematical/computational algorithm that combines VA data and SCI to produce ICSMs and CSMFs
- Less able to identify specific causes
- Very fast compared to physicians
- Effectively costless
- Highly replicable - same or very nearly the same results every time algorithms are run
- **Only realistic/feasible option for large-scale VA**

InterVA

- **Inter**pret **VA** ⇒ InterVA
- Developed by late Peter Byass and colleagues and refined over many years from roughly 2000 - 2020 (DOI)
- Supports standard WHO VAs: 2007, 2012, 2016
- Software available from Peter Byass' website: http://www.byass.uk/interva/
  - Windows installable and source code
  - ***No longer maintained***
- Software available from openVA Team - exactly replicated Byass' software
  - R package: https://cran.r-project.org/web/packages/InterVA5/index.html - ***continuously maintained***
  - Python openVA

InterVA

- VA data
  - Requires WHO standard VA: 2007, 2012, 2016
- SCI
  - ***Probbase***: conditional probabilities of observing a VA indicator given a specific cause
  - Elicited directly from physicians, **represents physicians' knowledge**
  - Quantifies relationship between each VA indicator and each cause in the WHO standard VA cause list
  - *Does not quantify relationships between groups of causes and causes*

InterVA

- Logic
  - *Inspired by Bayes' Rule* but **not mathematically probabilistic**
  - **Only uses information on VA indicators that are present**; ignores information on VA indicators that are absent
  - For each death, produces individual-level propensity associated with each cause
  - Propensities with values less than 0.4 are ignored
  - Standard software reports top three propensities for each death as long when they exceed 0.4
  - **Produces undetermined causes:** if no propensity exceeds 0.4 for a given death, the cause is classified as ***undetermined***
  - CSMFs calculated by summing up propensities for each cause across all deaths

InterVA

- Considerations
  - Assumes VA indicators and causes related in a 1:1 fashion, ignores the effect of groups of VA indicators
  - Only utilizes information on VA indicators that were observed, ignores information associated with the fact that some VA indicators were not observed
  - Because of the 0.4 threshold for propensities, **often identifies *undetermined* as the most likely 'cause' - typically about 20% of deaths**
  - Algorithm is very simple ⇒ very fast to run
  - *Raw VA data require checking and transforming into algorithm inputs* - ***this takes considerable time in all software***
  - **Part of openVA software**

InSilicoVA

- InSilicoVA ⇒ *in-silicon* VA, like *in-vivo* and *in-vitro*, i.e. on a computer chip
- Created by Zehang (Richard) Li, Tyler McCormick, and Sam Clark (DOI) to improve on InterVA by:
  - Utilizing information on both present and absent VA indicators
  - Eliminating undetermined causes
  - Estimating uncertainty/confidence associated with both ICSMs and CSMFs
- Supports standard WHO VAs: 2012, 2016
- Software available from openVA Team
  - R package: https://cran.r-project.org/web/packages/InSilicoVA/index.html - ***continuously maintained***
  - Python openVA

InSilicoVA

- VA data
  - Requires WHO standard VA: 2012, 2016
- SCI
  - ***Probbase***: conditional probabilities of observing a VA indicator given a specific cause
  - Elicited directly from physicians, **represents physicians' knowledge**
  - Quantifies relationship between each VA indicator and each cause in the WHO standard VA cause list
  - *Does not quantify relationships between groups of causes and causes*

InSilicoVA

- Logic
  - **Mathematically valid probabilistic model**
  - Estimates joint distribution of individual-level cause-specific probabilities and population-level cause-specific mortality fractions - this keeps individual and population levels consistent with each other
  - Uses full Bayes' Rule to calculate individual-level cause-specific probabilities
  - **Utilizes information on both present and absent VA indicators**
  - Produces distributions of cause-specific probabilities for each cause for each death
  - Produces distributions of cause-specific mortality fractions for a population of deaths
  - Distributions used to identify central probabilities and uncertainty/confidence
  - ***No undetermined causes***
    - Easily identified causes have high probability and low uncertainty
    - Hard-to-identify causes have low probability and high uncertainty

InSilicoVA

- Considerations
  - Assumes VA indicators and causes related in a 1:1 fashion, ignores the effect of groups of VA indicators
  - Uses all information on both present and absent VA indicators
  - **Does not produce undetermined causes**
  - Produces information about uncertainty/confidence in ICSMs and CSMFs
  - Algorithm is very complex compared to InterVA ⇒ takes much longer to run
  - *Raw VA data require checking and transforming into algorithm inputs - **this takes considerable time in all software***
  - **Part of openVA software**

Tariff

- Created by Institute for Health Metrics and Evaluation (IHME)
- Uses only information from VA indicators that are present
- Supports Population Health Metrics Research Consortium (PHMRC) standard questionnaires: separate versions for *PHMRC long* and *PHMRC short*
- Software available from IHME: SmartVA

Tariff 2.0 implemented in SmartVA software

- VA data
  - Requires ***PHRMC short*** questionnaire
- SCI
  - **SCI calculated as tariff values** directly from 'PHMRC Gold Standard Dataset' of reference deaths
    - **Represents relationships between VA indicators and causes in PHRMC-short standard VA in about 10,000 reference deaths**
    - Deaths from six facility locations in early 2000s - ***limited epidemiological context and not community deaths***
  - *Does not quantify relationships between groups of causes and causes*

Tariff 2.0 implemented in SmartVA software

- Logic
  - Sums cause-specific tariff values for VA indicators that exist to produce tariff score for each cause; largest tariff score is most probably cause - **not mathematically probabilistic**
  - **Only uses information on VA indicators that are present**; ignores information on VA indicators that are not present
  - For each death, produces individual-level tariff score associated with each cause
  - If no cause has large enough tariff score, cause is **undetermined**, similar to InterVA
  - CSMFs calculated by summing individual tariff scores

Tariff 2.0 implemented in SmartVA software

- Considerations
  - Assumes VA indicators and causes related in a 1:1 fashion, ignores the effect of groups of VA indicators
  - Uses all information on both present and absent VA indicators
  - **Produces undetermined causes**
  - Algorithm is very simple ⇒ very fast to run
  - **NOT part of openVA software**

History of VA algorithms/software ⇒ Python openVA

1. WHO Standard verbal autopsy, InterVA, InSilicoVA, & openVA
2. PHMRC verbal autopsy and SmartVA

WHO standard VA, InterVA, InSilicoVA, & openVA

- WHO Verbal Autopsy Reference Group (**VARG**) responsible for determining and updated WHO VA standards
- First widely used standards is VA 2007, important contribution from INDEPTH standard VA
  - CCVA supported by InterVA3
- WHO 2012 standard
  - CCVA supported by InterVA4 and InSilicoVA
- WHO 2016 standard
  - Electronic data capture standard using Open Data Kit (**ODK**)
  - CCVA supported by InterVA5 and InSilicoVA
- WHO 2020
  - Standard instrument available in ODK - ***shorter and streamlined compared to 2016***
  - CCVA not yet supported; will be supported by InterVA5 and InSilicoVA within about one year

WHO standard VA, InterVA, InSilicoVA, & openVA

- Pre-openVA
  - Before InterVA5, InterVA was closed-source, proprietary software distributed free of charge as a Windows executable - ***no one except Peter Byass really knew what it was doing***
  - Until recently SmartVA was also closed-source, proprietary software distributed as a Windows executable - ***no one except IHME really knew what it was doing***
- openVA
  - **Purpose: to make VA software open source and free so that anyone can fully understand what it is doing and use it**
  - Reverse-engineered both InterVA and Tariff 1.0 and created open source R packages that implement the algorithms: InterVA, Tariff 1.0
  - Created and added InSilicoVA
  - Created R package openVA as a wrapper around the algorithms, including ability transform raw data and view/save results in various formats: openVA

openVA

- openVA Team
  - Led by Sam Clark
  - Software primarily Richard Li and Jason Thomas
  - Many other members working on various aspects of VA
- openVA Pipeline
  - Uses Python, R, and SQL to automate retrieval of VA deaths from ODK, cause-classification, and pushing causes and raw VA data to DHIS-2
  - Command line and chron job
- All openVA software is open source
  - R CRAN, many packages
  - Github repositories at https://github.com/verbal-autopsy-software

Python openVA

- Context
  - Original openVA written using R and Java and distributed as R packages
  - All open source and free
  - R code utilizes myriad pre-existing packages
  - openVA R is a command line tool
  - Both openVA software and all packages and related software must be kept up to date - OK for researchers but not production users
- **Installing, maintaining, and running R-version of openVA is challenging/impossible for many users**

Python openVA

- Python openVA
  - **Aims to solve the useability issue facing many users of R openVA**
  - Written exclusively in Python
  - Compiled into Windows and MacOS executables
  - **Installs like a ‘normal’ application** on both operating systems
  - Has a windowed graphical user interface like a ‘normal’ application
  - User interacts with Python openVA by pointing and clicking - **no command line**
  - **Still free and open source** - source code (will be) available on Github repository after initial testing

PHMRC VA and SmartVA

- Tariff 1
  - Uses PHMRC long questionnaire
  - CCVA using old version of SmartVA
- Tariff 2
  - Uses PHMRC short questionnaire - ***much shorter***
  - CCVA supported by SmartVA Analyze software
    - Available as Windows and Linux executables
    - Also available on open-source Github repository
- **openVA does not support PHMRC questionnaires** (might in future)