Introduction to CrossVA & openVA

Jason Thomas

March 5th, 2018

NEW SLIDES & R Package

https://github.com/verbal-autopsy-software/Indonesia

also download. . .

- the *new* CrossVA_0.9.6.zip package
- practice data (from WHO 2016): `odk151_practice.csv`

remember where we save these(!) so we can use `setwd()` to set our working directory

Overview

Morning

- Installing R packages
    - **CrossVA** as a special case
- Example Workflow with practice data
- Workflow with Indonesian data

Afternoon

- Using **openVA** to run InterVA5 algorithm
- Additional tools in the **openVA** package

Setting up R Packages

Recall that one of the most useful features of R is the long list of packages

Setting up R Packages

Recall that one of the most useful features of R is the long list of packages

The list all of the packages currently installed in your R library with the following command: `library()`

Setting up R Packages

Recall that one of the most useful features of R is the long list of packages

The list all of the packages currently installed in your R library with the following command: `library()`

- to load a package in your library, use: `library(package_name)`

Setting up R Packages

Recall that one of the most useful features of R is the long list of packages

The list all of the packages currently installed in your R library with the following command: `library()`

- to load a package in your library, use: `library(package_name)`
- e.g., `library(stats)` loads the **stats** package

Setting up R Packages

Recall that one of the most useful features of R is the long list of packages

The list all of the packages currently installed in your R library with the following command: `library()`

- to load a package in your library, use: `library(package_name)`
- e.g., `library(stats)` loads the **stats** package

The R package openVA depends on other packages, so we must take care of these dependencies first.

R Packages: Installing Java for openVA

The InSilcoVA algorithm (part of **openVA**) has to do a lot of computing, and it relies on Java to do the leg work (since it is much faster than R)

Dependency for our dependency: **rJava**

- we need to install: Java JDK
- then we must configure R so it can find Java

```r
Sys.setenv(JAVA_HOME = "C:/Path/to/Java/jdk")
## For example, on my computer I used
## Sys.setenv('JAVA_HOME' = 'C:/Program
## Files/Java/jdk-11.0.1')
```

Now we can install **openVA** and all if the other R packages it depends on

R Packages: openVA

We can install R packages (from CRAN) using the menus:

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages* -> *Install Package(s)*

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages* -> *Install Package(s)*
- (or we could use the command: `install.packages`)

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages* -> *Install Package(s)*
- (or we could use the command: `install.packages`)

A new window will pop up called *Secure CRAN mirrors* (choose the mirror closest to your geographic location & click OK)

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages* -> *Install Package(s)*
- (or we could use the command: `install.packages`)

A new window will pop up called *Secure CRAN mirrors* (choose the mirror closest to your geographic location & click OK)

Select the **openVA** package, click OK, and then R will install the dependencies (as well as the dependencies for our dependencies, and so on)

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages* -> *Install Package(s)*
- (or we could use the command: `install.packages`)

A new window will pop up called *Secure CRAN mirrors* (choose the mirror closest to your geographic location & click OK)

Select the **openVA** package, click OK, and then R will install the dependencies (as well as the dependencies for our dependencies, and so on)

- (ok to install in a personal library or folder in your personal directory)

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages* -> *Install Package(s)*
- (or we could use the command: `install.packages`)

A new window will pop up called *Secure CRAN mirrors* (choose the mirror closest to your geographic location & click OK)

Select the **openVA** package, click OK, and then R will install the dependencies (as well as the dependencies for our dependencies, and so on)

- (ok to install in a personal library or folder in your personal directory)
- InSilicoVA v1.2.5 (latest version March 4, 2019)

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages* -> *Install Package(s)*
- (or we could use the command: `install.packages`)

A new window will pop up called *Secure CRAN mirrors* (choose the mirror closest to your geographic location & click OK)

Select the **openVA** package, click OK, and then R will install the dependencies (as well as the dependencies for our dependencies, and so on)

- (ok to install in a personal library or folder in your personal directory)
- InSilicoVA v1.2.5 (latest version March 4, 2019)
- InterVA5 v1.0.2

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages* -> *Install Package(s)*
- (or we could use the command: `install.packages`)

A new window will pop up called *Secure CRAN mirrors* (choose the mirror closest to your geographic location & click OK)

Select the **openVA** package, click OK, and then R will install the dependencies (as well as the dependencies for our dependencies, and so on)

- (ok to install in a personal library or folder in your personal directory)
- InSilicoVA v1.2.5 (latest version March 4, 2019)
- InterVA5 v1.0.2
- InterVA4 v1.7.5

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages -> Install Package(s)*
- (or we could use the command: `install.packages`)

A new window will pop up called *Secure CRAN mirrors* (choose the mirror closest to your geographic location & click OK)

Select the **openVA** package, click OK, and then R will install the dependencies (as well as the dependencies for our dependencies, and so on)

- (ok to install in a personal library or folder in your personal directory)
- InSilicoVA v1.2.5 (latest version March 4, 2019)
- InterVA5 v1.0.2
- InterVA4 v1.7.5
- Tariff v1.0.5

R Packages: openVA

We can install R packages (from CRAN) using the menus:

- *Packages* -> *Install Package(s)*
- (or we could use the command: `install.packages`)

A new window will pop up called *Secure CRAN mirrors* (choose the mirror closest to your geographic location & click OK)

Select the **openVA** package, click OK, and then R will install the dependencies (as well as the dependencies for our dependencies, and so on)

- (ok to install in a personal library or folder in your personal directory)
- InSilicoVA v1.2.5 (latest version March 4, 2019)
- InterVA5 v1.0.2
- InterVA4 v1.7.5
- Tariff v1.0.5
- (nbc not included when installing **openVA**)

R Package

If we configured R and Java, then **openVA** should load without a hitch...

```
library(openVA)
------- Attaching packages for openVA 1.0.7 ------
v InSilicoVA 1.2.5
v InterVA4   1.7.5
v InterVA5   1.0.2
v Tariff     1.0.5
-- Optional packages (require manual installation
x nbc4va
If you need to use these methods, you may need to load or
the packages: nbc4va.
You can run in your R terminal:
library('nbc4va')
```

R Packages: error loading openVA

There is a chance R will complain

```
library(openVA)
Error: package or namespace load failed for `openVA':
.onLoad failed in loadNamespace() for 'rJava', details:
call: dirname(this$RuntimeLib)
error: a character vector argument expected
```

so we need to re-configure R so it can find Java

```r
options(java.home = "C:/Path/to/Java/jdk")
library(openVA)
```

If this fails, then we need to try and set an environment variable:
On Windows

R Packages: CrossVA (special case)

CrossVA *could* be installed in a similar manner. . .

R Packages: CrossVA (special case)

CrossVA *could* be installed in a similar manner. . .

BUT, I had to make some changes to the package to work well with the Indonesian version of the ODK form

R Packages: CrossVA (special case)

CrossVA *could* be installed in a similar manner. . .

BUT, I had to make some changes to the package to work well with the Indonesian version of the ODK form

Install the new **CrossVA** package we downloaded:

R Packages: CrossVA (special case)

CrossVA *could* be installed in a similar manner. . .

BUT, I had to make some changes to the package to work well with the Indonesian version of the ODK form

Install the new **CrossVA** package we downloaded:

- *Packages* -> *Install package(s) from local file. . .*

R Packages: CrossVA (special case)

CrossVA *could* be installed in a similar manner. . .

BUT, I had to make some changes to the package to work well with the Indonesian version of the ODK form

Install the new **CrossVA** package we downloaded:

- *Packages* -> *Install package(s) from local file. . .*
- a new window will open, navigate to the CrossVA_0.9.6.zip, and click OK

R Packages: CrossVA (special case)

CrossVA *could* be installed in a similar manner. . .

BUT, I had to make some changes to the package to work well with the Indonesian version of the ODK form

Install the new **CrossVA** package we downloaded:

- *Packages -> Install package(s) from local file. . .*
- a new window will open, navigate to the CrossVA_0.9.6.zip, and click OK

Load our new package with: `library(CrossVA)`

Example Workflow with Practice Data

Now let's walk through a simple analysis using our practice data.

1. Open Script file (I'll use day2_openVA.R)
2. Read our (CSV) data file into R
   - remember to set your working directory
3. Run **CrossVA** to prepare our data
4. Use **openVA** to run the InSilicoVA algorithm
5. summarize results

Example Workflow with Practice Data: working directory

Set our working directory and make sure our practice data file is there. . .

```r
setwd("C:/Users/jarat/Indonesia/")
```

```r
dir()
```

```
## [1] "CrossVA_0.9.6.zip"
## [2] "crossVA_openVA.pdf"
## [3] "day2_openVA.R"
## [4] "errorlog_insilico.txt"
## [5] "odk151_practice.csv"
```

Example Workflow with Practice Data: read in data

Read in our CSV data file

```r
odkExport <- read.csv("odk151_practice.csv", stringsAsFactors = FALSE)
str(odkExport)
```

```
## 'data.frame':    54 obs. of  529 variables:
##  $ SubmissionDate
##  $ presets.Id10002
##  $ presets.Id10003
##  $ presets.Id10004
##  $ respondent_backgr.Id10007
##  $ respondent_backgr.Id10008
##  $ respondent_backgr.Id10009
##  $ respondent_backgr.Id10010
##  $ respondent_backgr.Id10012
##  $ respondent_backgr.Id10013
##  $ respondent_backgr.Id10011
##  $ consented.deceased_CRVS.info_on_deceased.Id10017
##  $ consented.deceased_CRVS.info_on_deceased.Id10018
##  $ consented.deceased_CRVS.info_on_deceased.Id10019
##  $ consented.deceased_CRVS.info_on_deceased.Id10020
##  $ consented.deceased_CRVS.info_on_deceased.Id10021
```

Example Workflow with Practice Data: read in data (cont.)

When reading in a CSV file, the resulting object is a data frame with rows and columns

```
is.data.frame(odkExport)
```

```
## [1] TRUE
```

```
dim(odkExport)
```

```
## [1]  54 529
```

Another useful command is

```
names(odkExport)
```

which will print all of the variable names (or column names)

With a data frame, we can access a single variable/column using $

```
table(odkExport$presets.Id10004, useNA = "always")
```

```
##
##   DK  dry  wet <NA>
##    1   24   29     0
```

Example Workflow with Practice Data: CrossVA

Use **CrossVA** to prepare the data **openVA**

```
library(CrossVA)  ## make sure package is loaded
data1 <- odk2openVA(odk = odkExport)
```

```
## Assuming WHO questionnaire version is 1.5.1
```

```
dim(data1)
```

```
## [1]  54 354
```

```
names(data1)
```

```
##   [1] "ID"    "i004a" "i004b" "i019a" "i019b"
##   [6] "i022a" "i022b" "i022c" "i022d" "i022e"
##  [11] "i022f" "i022g" "i022h" "i022i" "i022j"
##  [16] "i022k" "i022l" "i022m" "i022n" "i059o"
##  [21] "i077o" "i079o" "i082o" "i083o" "i084o"
##  [26] "i085o" "i086o" "i087o" "i089o" "i090o"
##  [31] "i091o" "i092o" "i093o" "i094o" "i095o"
##  [36] "i096o" "i098o" "i099o" "i100o" "i104o"
##  [41] "i105o" "i106a" "i107o" "i108a" "i109o"
##  [46] "i110o" "i111o" "i112o" "i113o" "i114o"
##  [51] "i115o" "i116o" "i120a" "i120b" "i123o"
```

Example Workflow with Practice Data: openVA

Now our data are ready to feed into the InSilicoVA algorithm.

Example Workflow with Practice Data: openVA

Now our data are ready to feed into the InSilicoVA algorithm.

InSilicoVA (and InterVA) can be run using the `codeVA` function that is part of the **openVA** package.

Example Workflow with Practice Data: openVA

Now our data are ready to feed into the InSilicoVA algorithm.

InSilicoVA (and InterVA) can be run using the `codeVA` function that is part of the **openVA** package.

Remember that InSilicoVA estimates uncertainty and brings consistency between. . .

Example Workflow with Practice Data: openVA

Now our data are ready to feed into the InSilicoVA algorithm.

InSilicoVA (and InterVA) can be run using the `codeVA` function that is part of the **openVA** package.

Remember that InSilicoVA estimates uncertainty and brings consistency between. . .

- the distribution of deaths in the population (CSMF) &

Example Workflow with Practice Data: openVA

Now our data are ready to feed into the InSilicoVA algorithm.

InSilicoVA (and InterVA) can be run using the `codeVA` function that is part of the **openVA** package.

Remember that InSilicoVA estimates uncertainty and brings consistency between...

- the distribution of deaths in the population (CSMF) &
- the assigned causes at the individual level

Example Workflow with Practice Data: openVA

Now our data are ready to feed into the InSilicoVA algorithm.

InSilicoVA (and InterVA) can be run using the `codeVA` function that is part of the **openVA** package.

Remember that InSilicoVA estimates uncertainty and brings consistency between. . .

- the distribution of deaths in the population (CSMF) &
- the assigned causes at the individual level

This estimation takes a little bit of time (and we should see Java step in to do some of the heavy lifting)

Example Workflow with Practice Data: openVA

openVA code for running InSilicoVA with data from the WHO 2016 questionnaire:

```
results1 <- codeVA(data = data1, data.type = "WHO2016",
    model = "InSilicoVA", warning.write = TRUE)
```

```
## Performing data consistency check...

## .....

## Data check finished.

## Warning: 66 symptom missing completely and added to missing list
## List of missing symptoms:
##  i059o, i091o, i093o, i201b, i203a, i204o, i205a, i214o, i216a, i217

## Not all causes with CSMF > 0.02 are convergent.

## Increase chain length with another 10000 iterations

## Not all causes with CSMF > 0.02 are convergent.

## Increase chain length with another 20000 iterations

## Not all causes with CSMF > 0.02 are convergent.
```

Example Workflow with Practice Data: summary

Let's take a look at the results (note the errorlog_insilico.txt file with information on the data consistency checks.

```
dir()
```

```
## [1] "CrossVA_0.9.6.zip"
## [2] "crossVA_openVA.pdf"
## [3] "day2_openVA.R"
## [4] "errorlog_insilico.txt"
## [5] "odk151_practice.csv"
```

Example Workflow with Practice Data: summary (cont.)

Let's take a look at the results (note the `errorlog_insilico.txt` file with information on the data consistency checks.

```r
summary(results1, top = 8)
```

```
## InSilicoVA Call:
## 54 death processed
## 40000 iterations performed, with first 20000 iterations discarded
##  2000 iterations saved after thinning
## Fitted with re-estimated conditional probability level table
## Data consistency check performed as in InterVA4
##
## Top 8 CSMFs:
##                                    Mean Std.Error
## Acute resp infect incl pneumonia 0.2315    0.0572
## Diarrhoeal diseases              0.1062    0.0422
## HIV/AIDS related death           0.1020    0.0413
## Acute cardiac disease            0.1009    0.0399
## Pulmonary tuberculosis           0.0722    0.0355
## Birth asphyxia                   0.0655    0.0318
## Stroke                           0.0654    0.0362
## Malaria                          0.0540    0.0312
```

Example Workflow with Practice Data: getTopCOD

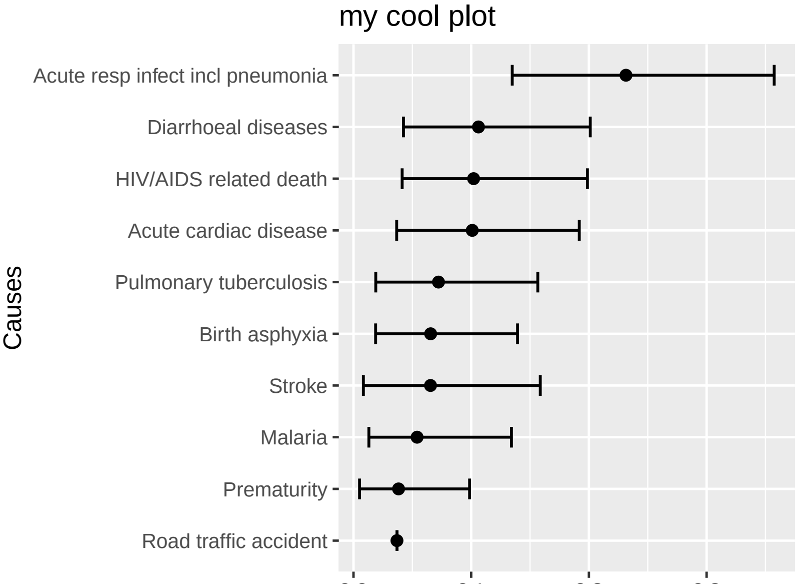
Get the top causes of death

```r
results1_cod <- getTopCOD(results1)
head(results1_cod)
```

```
##                                           ID
## 1 uuid:fe4c4809-d3e9-45e4-bf63-4effed64ae7a
## 2 uuid:20cd4d64-86f6-4428-b24b-9cdd9695a057
## 3 uuid:0f22cef1-dcfd-42c5-ab53-50fe6ff904ea
## 4 uuid:9c764b75-46f3-4102-810a-42912ccecc43
## 5 uuid:ef4a567e-1f8b-469c-ba74-89d13afd0be4
## 6 uuid:a4b7d705-77b8-4721-8962-d17fdb7f223d
##                              cause
## 1           HIV/AIDS related death
## 2           HIV/AIDS related death
## 3 Acute resp infect incl pneumonia
## 4 Acute resp infect incl pneumonia
## 5 Acute resp infect incl pneumonia
## 6           HIV/AIDS related death
```

Example Workflow with Practice Data: summarize

```r
plotVA(results1, title = "my cool plot", top = 10)
```



my cool plot

Workflow with Indonesian data

Now try and repeat these steps, but with Indonesian data

NOTE: an important difference is **CrossVA**

- use the option: `strictNames = TRUE` with the `odk2openVA()` function

Example with InterVA5

openVA is a one-stop shop, and it is very easy to run InterVA5 as well

- with the same tools for summarizing results

```
results2 <- codeVA(data = data1, data.type = "WHO2016",
    model = "InterVA", version = "5.0", HIV = "l",
    Malaria = "v", directory = ".")
```

```
## .....9% completed
## .....19% completed
## .....28% completed
## .....37% completed
## .....46% completed
## .....56% completed
## .....65% completed
## .....74% completed
## .....83% completed
## .....93% completed
## ....
```

Example with InterVA5: summary

```r
summary(results2, top = 8)
```

```
## InterVA5 fitted on 54 deaths
## CSMF calculated using reported causes by InterVA5 only
## The remaining probabilities are assigned to 'Undetermined'
##
## Top 8 CSMFs:
##  cause                            likelihood
##  Acute resp infect incl pneumonia 0.2245
##  HIV/AIDS related death           0.1115
##  Acute cardiac disease            0.1087
##  Diarrhoeal diseases              0.0943
##  Stroke                           0.0755
##  Pulmonary tuberculosis           0.0754
##  Congenital malformation          0.0377
##  Birth asphyxia                   0.0377
##
## Top 6 Circumstance of Mortality Category:
##  cause          likelihood
##  Knowledge      0.3704
##  Emergency      0.1852
##  Inevitable     0.1852
```

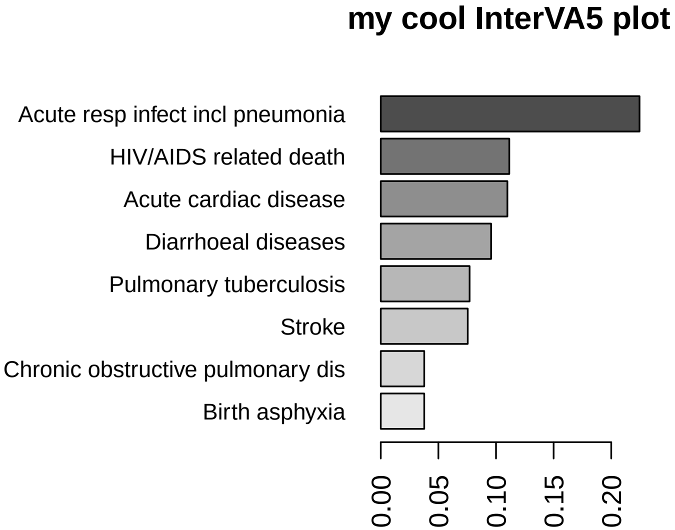
Example with InterVA5:

```
results2_cod <- getTopCOD(results2)
head(results2_cod)
```

```
##                                           ID
## 1 uuid:fe4c4809-d3e9-45e4-bf63-4effed64ae7a
## 2 uuid:20cd4d64-86f6-4428-b24b-9cdd9695a057
## 3 uuid:0f22cef1-dcfd-42c5-ab53-50fe6ff904ea
## 4 uuid:9c764b75-46f3-4102-810a-42912ccecc43
## 5 uuid:ef4a567e-1f8b-469c-ba74-89d13afd0be4
## 6 uuid:a4b7d705-77b8-4721-8962-d17fdb7f223d
##                              cause
## 1           HIV/AIDS related death
## 2           HIV/AIDS related death
## 3 Acute resp infect incl pneumonia
## 4 Acute resp infect incl pneumonia
## 5 Acute resp infect incl pneumonia
## 6           HIV/AIDS related death
```

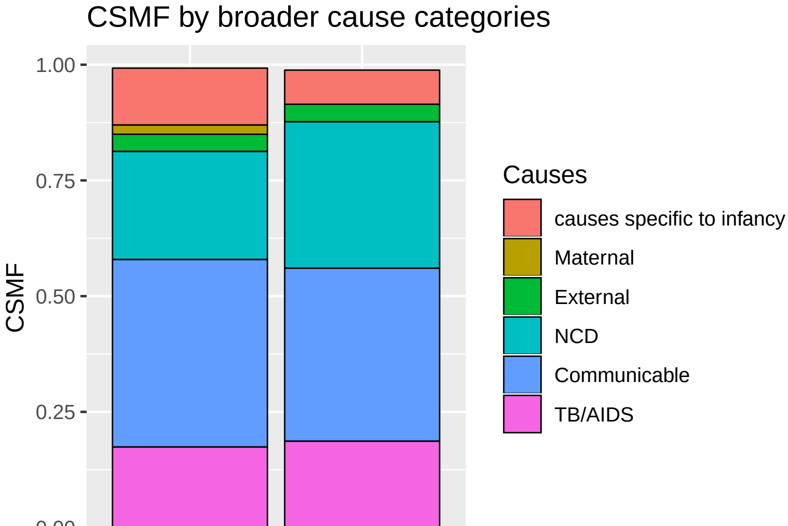
Example Workflow with Practice Data: summarize

```
plotVA(results2, title = "my cool InterVA5 plot", top = 8)
```



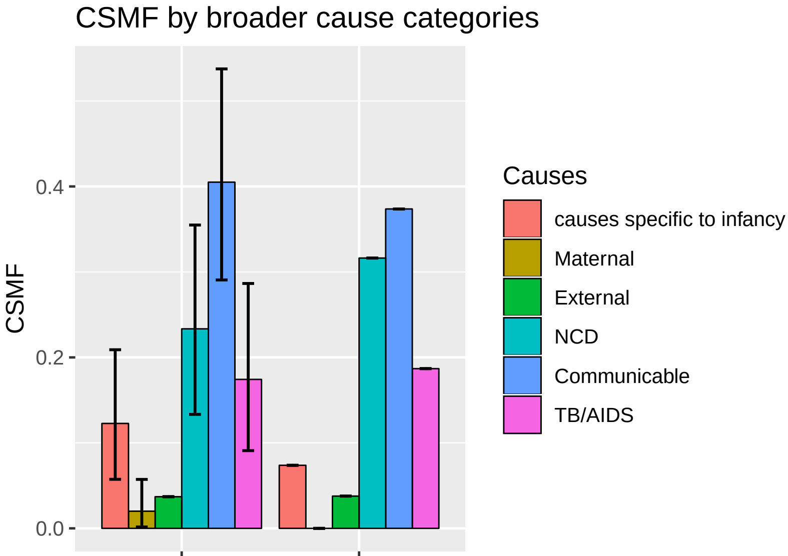
Comparing InSilicoVA & InterVA

```
compare <- list(InSilicoVA = results1, InterVA5 = results2)
stackplotVA(compare, sample.size.print = TRUE, xlab = "",
    angle = 0)
```



Comparing InSilicoVA & InterVA (cont.)

```
stackplotVA(compare, sample.size.print = TRUE, xlab = "",
    angle = 0, type = "dodge")
```



Running InSilicoVA with Subgroups

```r
results1b <- codeVA(data = data1, data.type = "WHO2016",
    model = "InSilicoVA", subpop = list("i019a", "i019b"))
```

```
## Performing data consistency check...

## .....

## Data check finished.

## Warning: 66 symptom missing completely and added to missing list
## List of missing symptoms:
##  i059o, i091o, i093o, i201b, i203a, i204o, i205a, i214o, i216a, i217

## Not all causes with CSMF > 0.02 are convergent.

## Increase chain length with another 10000 iterations

## Not all causes with CSMF > 0.02 are convergent.

## Increase chain length with another 20000 iterations

## Not all causes with CSMF > 0.02 are convergent.
##  Please check using csmf.diag() for more information.
```

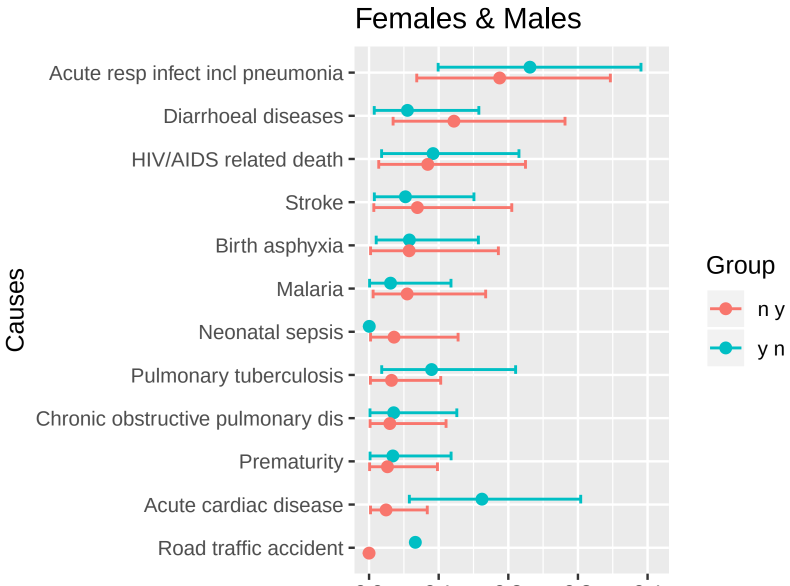
Running InSilicoVA with Subgroups: summarizing results

```r
summary(results1b)
```

```
## InSilicoVA Call:
## 54 death processed
## 40000 iterations performed, with first 20000 iterations discarded
##  2000 iterations saved after thinning
## Fitted with re-estimated conditional probability level table
## Data consistency check performed as in InterVA4
## Sub population frequencies:
## n y y n
##  24  30
##
## n y - Top 10 CSMFs:
##                                      Mean
## Acute resp infect incl pneumonia  0.1876
## Diarrhoeal diseases               0.1219
## HIV/AIDS related death            0.0843
## Stroke                            0.0694
## Birth asphyxia                    0.0576
## Malaria                           0.0547
## Neonatal sepsis                   0.0359
## Pulmonary tuberculosis            0.0322
```

Running InSilicoVA with Subgroups: plot

```
plotVA(results1b, type = "compare", title = "Females & Males")
```



Using Indonesian Data

Run the InSilicoVA algorithm separately for males and females and compare the results

Use the InterVA5 algorithm to assign causes of death

- use different levels for HIV & Malaria to see how the results change

Summarize and compare the results between InterVA5 and InSilicoVA