ONTARIO
NEURODEGENERATIVE
DISEASE RESEARCH
INITIATIVE

Data Preparation App 0.9.0.9002 - Reference Guide

On behalf of the Neuroinformatics and Biostatistics (NIBS) platform

Updated as of March 22, 2021

This reference guide helps you install and use the ONDRI Data Preparation App, with instructions on how to run the app, its limitations, and future updates.

Contents

1 Overview

The purpose of the data preparation app is to streamline a pipeline for data exploration and cleaning, and prepare tabular data for downstream analyses such as principal component analysis (PCA), correspondence analysis (CA), and outlier analysis. The app follows ONDRI protocol for preparing data, which consists of the following steps in sequential order as outlined below:

1. Read in tabular data and associated data dictionary.
2. Perform missingness checks and removal of variables and participants.
3. Select variables for analyses and define variable types.
4. Perform transformations to recode different variable types into a disjunctive format.
5. Perform imputation to handle missing values.
6. Residualize data based on age and sex. (OPTIONAL and not added in the current version of the app)

In the NIBS curation process, the data preparation app is used after performing a standards check on the required components of a data package. Likewise, it is used before running outlier analysis on the tabular data. Overall, the data preparation app serves as a middleware between the existing ONDRI Standards App and ONDRI Outliers App.

2 Installation and Deployment

1. Install R first and then RStudio. Please choose the correct installer carefully as it will depend on your computer's operating system.

2. Install the `GSVD` and `ours` packages (which are not available through CRAN) with the following lines of code:

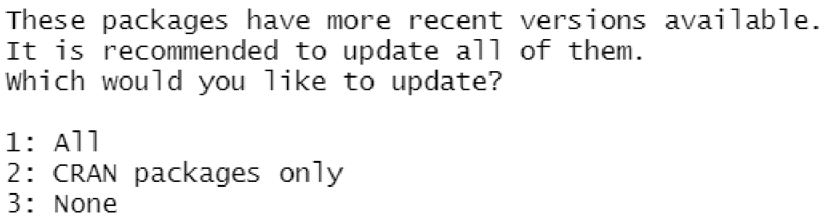
```r
if (!require("devtools")){
  install.packages("devtools")
}

if (!require("GSVD")){
  Sys.setenv(R_REMOTES_NO_ERRORS_FROM_WARNINGS = TRUE)
  devtools::install_github("derekbeaton/GSVD")
}
if (!require("ours")){
  Sys.setenv(R_REMOTES_NO_ERRORS_FROM_WARNINGS = TRUE)
  devtools::install_github("derekbeaton/OuRS", subdir = "/OuRS")
}
```

3. Download and install the shiny app directly with the following lines of code:

```r
if (!require("devtools")){
  install.packages("devtools")
}
devtools::install_github(repo = "ondri-nibs/dataprep_app")
```

4. Type `ONDRIDataPrepApp::installPackages()` to install any missing packages and/or dependencies. If you get the following message in your RStudio console, please type 3.



```
These packages have more recent versions available.
It is recommended to update all of them.
Which would you like to update?

1: All
2: CRAN packages only
3: None
```

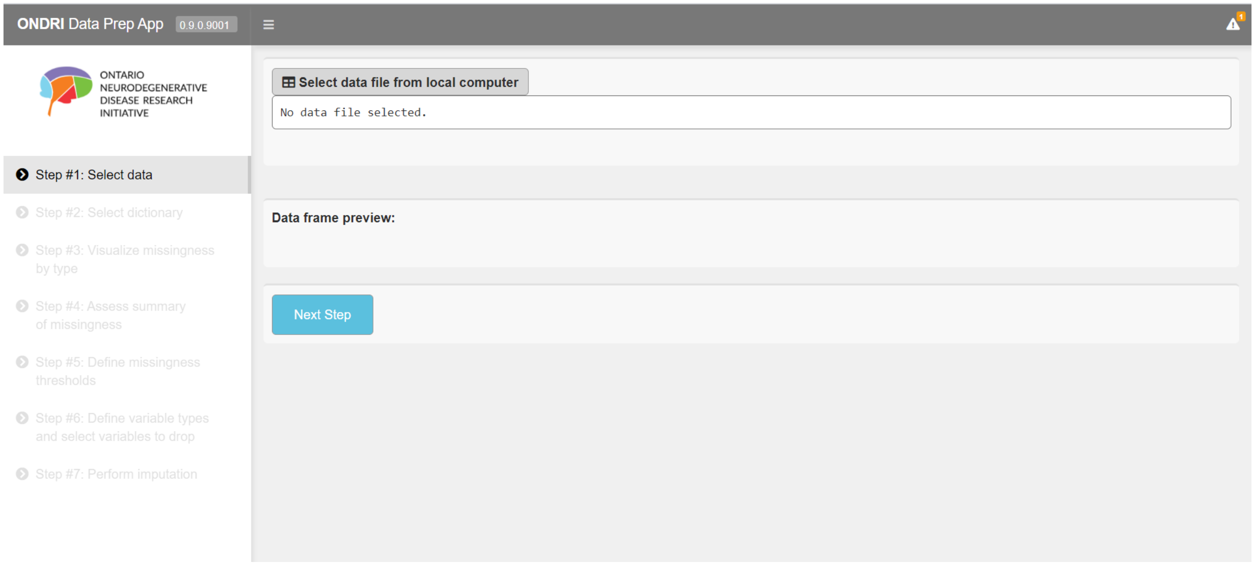
Figure 1: This message may appear multiple times during the installation process.

5. When installation is complete, type `ONDRIDataPrepApp::runApp()` to open the app.

3 Running App

IMPORTANT: Please note that the data package has to pass standards through the ONDRI Standards App before proceeding to data preparation.

The app breaks the pipeline into seven distinct and sequential steps, which are shown in a menu in the left pane of the user interface (UI). Each step is addressed below:


ONDRI Data Prep App 0.9.0.9001
ONTARIO NEURODEGENERATIVE DISEASE RESEARCH INITIATIVE
Step #1: Select data
Step #2: Select dictionary
Step #3: Visualize missingness by type
Step #4: Assess summary of missingness
Step #5: Define missingness thresholds
Step #6: Define variable types and select variables to drop
Step #7: Perform imputation
Select data file from local computer
Data frame preview:
Next Step

```
No data file selected.
```

Figure 2: The initial UI when the app is opened. The seven steps of the pipeline are shown in the menu.

3.1 Select data

To read in tabular data through the app, please select data as a csv file from your local computer.

3.2 Select dictionary

Likewise, to read in the associated data dictionary through the app, please select a csv file from your local computer.

3.3 Visualize missingness

Missingness is visualized through a heatmap, which assesses missingness by ONDRI missing codes. Each missing code is represented by a distinct colour as shown in Figure 3. It is important to note that the heatmap indicates blank values through a separate colour (white), which represents conditional missingness within the data (i.e. If answer is no, please leave blank).

3.4 Assess summary of missingness

There are a total of four distinct tables used to summarize missingness by ONDRI missing codes. Missingness can be viewed through either percentage or absolute count for both variables (columns) and participants (rows). The UI is controlled through radio buttons which allow for smooth and easy navigation between tables as shown in Figure 4. Overall, this step serves as an alternative method of understanding the missingness within each variable and participant.

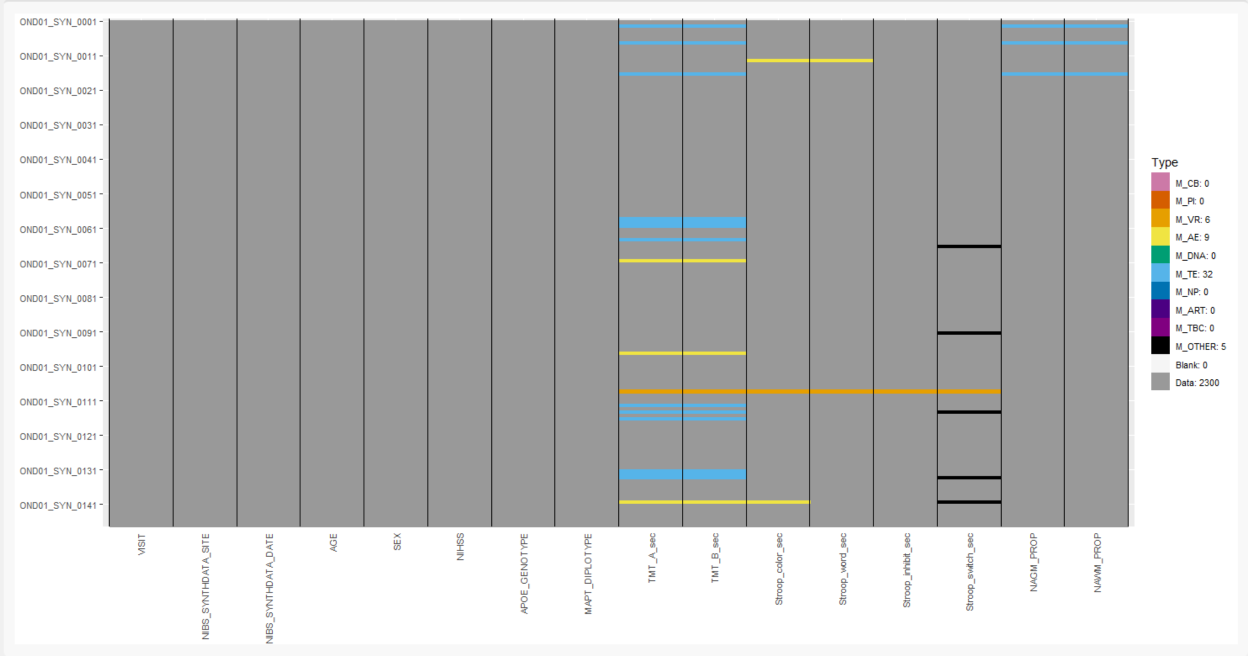
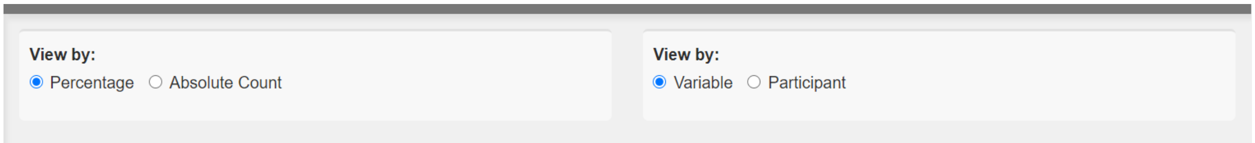


Figure 3: An example of a missingness heatmap.



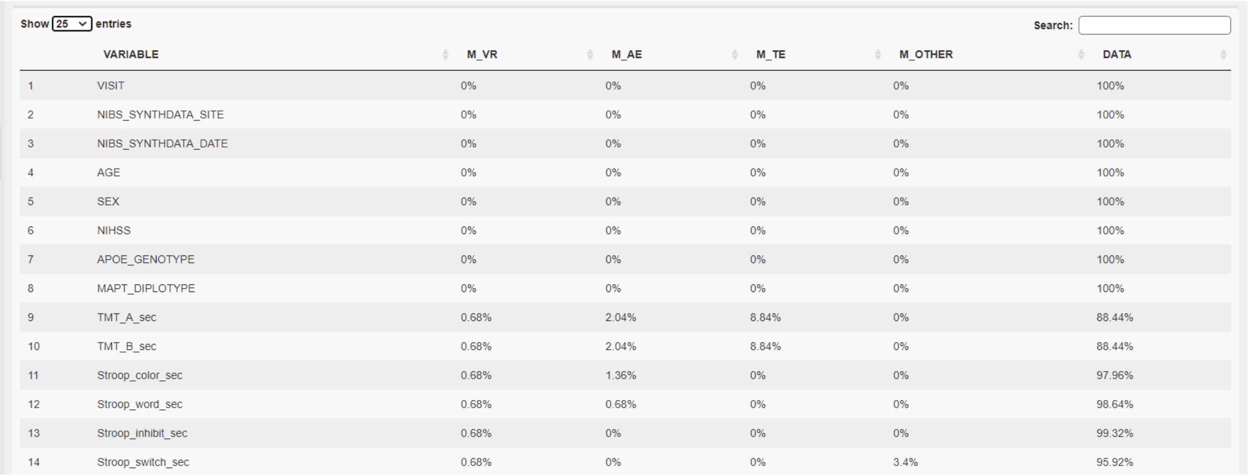
View by:

● Percentage ○ Absolute Count

View by:

● Variable ○ Participant

Figure 4: The radio buttons used to control missingness table visualization.



Show 25 entries

Search:

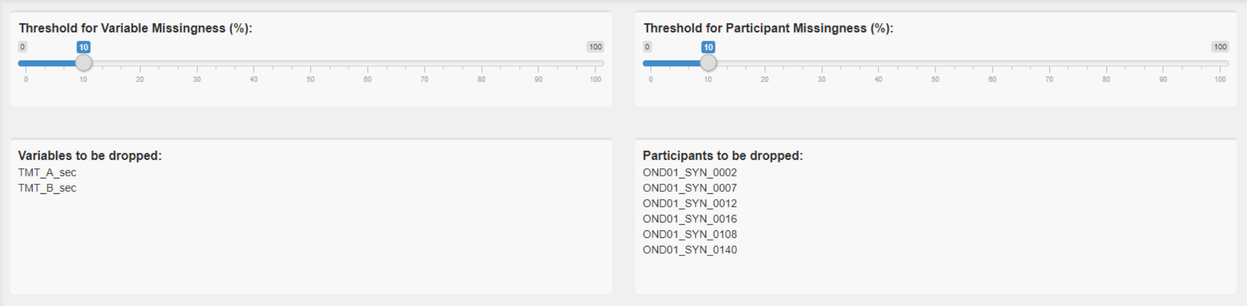
	VARIABLE	M_VR	M_AE	M_TE	M_OTHER	DATA
1	VISIT	0%	0%	0%	0%	100%
2	NIBS_SYNTHDATA_SITE	0%	0%	0%	0%	100%
3	NIBS_SYNTHDATA_DATE	0%	0%	0%	0%	100%
4	AGE	0%	0%	0%	0%	100%
5	SEX	0%	0%	0%	0%	100%
6	NIHSS	0%	0%	0%	0%	100%
7	APOE_GENOTYPE	0%	0%	0%	0%	100%
8	MAPT_DIPLOTYPE	0%	0%	0%	0%	100%
9	TMT_A_sec	0.68%	2.04%	8.84%	0%	88.44%
10	TMT_B_sec	0.68%	2.04%	8.84%	0%	88.44%
11	Stroop_color_sec	0.68%	1.36%	0%	0%	97.96%
12	Stroop_word_sec	0.68%	0.68%	0%	0%	98.64%
13	Stroop_inhibit_sec	0.68%	0%	0%	0%	99.32%
14	Stroop_switch_sec	0.68%	0%	0%	3.4%	95.92%

Figure 5: An example of a table summarizing missingness in variables by percentage.

3.5 Define missingness thresholds

The two sliders allow for dropping of variables and participants respectively depending on their percentage of missing values. The default threshold is 10%, which drops any variables and participants containing greater than 10% missingness (in other words, less than 90% completeness). Although the sliders allow for a percentage anywhere in the range of 0% to 100%, they should ideally be somewhere between 0% and 10% for optimal downstream analyses.

Please note that the algorithm in this step is an iterative procedure that removes variables first followed by participants until the threshold in both sliders are satisfied. If the sliders get updated, the app will automatically update a list of the variables and participants that would be dropped in realtime. Once percentages are confirmed, a preview of the data frame will be provided for user viewing. To update the sliders after confirmation, please click the reset button.



Threshold for Variable Missingness (%): 10

Threshold for Participant Missingness (%): 10

Variables to be dropped:
TMT_A_sec
TMT_B_sec

Participants to be dropped:
OND01_SYN_0002
OND01_SYN_0007
OND01_SYN_0012
OND01_SYN_0016
OND01_SYN_0108
OND01_SYN_0140

Figure 6: The sliders used to control the missingness threshold and the output of variables and participants that would be dropped.

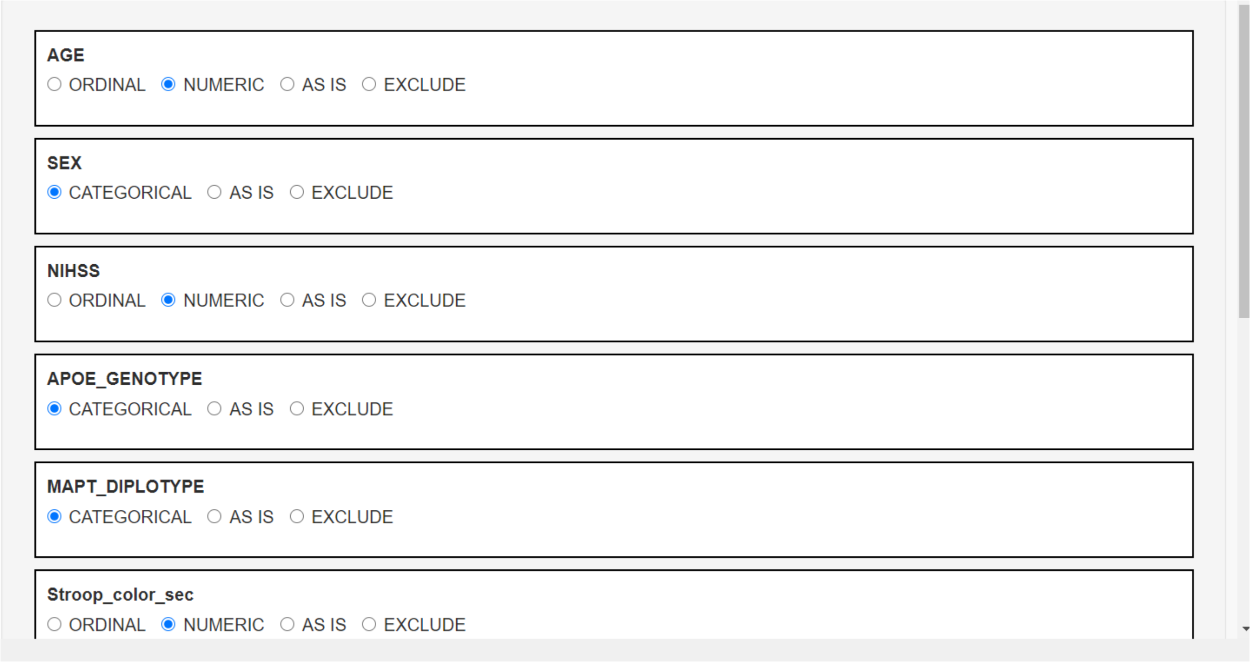
3.6 Define variable types and select variables to drop

Using the data dictionary as a reference, the app will predetermine the appropriate data types corresponding to each variable. The value in the TYPE column in the data dictionary will represent the default value, except in the following scenarios:

1. If at least one blank cell is detected in the column, the variable will automatically be put into the **EXCLUDE** category, with no other options being allowed.
2. If only one unique value exists in the column (excluding ONDRI missing codes), the only options provided will be **AS IS** and **EXCLUDE**.
3. If a column is of type **MIXED**, **TEXT**, or **DATE** as indicated in the data dictionary, the only options provided will be **AS IS** and **EXCLUDE**.
4. If a column contains non-numeric values (excluding ONDRI missing codes), **ORDINAL** and **NUMERIC** will not be provided as options (regardless of the value provided in the TYPE column in the data dictionary).

Once variable types have been identified and confirmed, transformation of the data is performed automatically (in the background) so that different variable types can be recoded into a disjunctive format. A preview of the data frame after transformations will be outputted for user viewing. If a variable is of type **AS IS**, they will not be recoded and will simply be added into the prepared data for output. On the other hand, variables of type **EXCLUDE** will be removed from the prepared data. To update the variable data types after confirmation, please click the reset button.

NOTE: For ordinal variables, please ensure that they are coded as a numerical based ordinal scale. If the variable values are character based, please convert them to numeric and ordered values manually before running the app. Otherwise, please treat these variables as categorical.



AGE
○ ORDINAL ● NUMERIC ○ AS IS ○ EXCLUDE

SEX
● CATEGORICAL ○ AS IS ○ EXCLUDE

NIHSS
○ ORDINAL ● NUMERIC ○ AS IS ○ EXCLUDE

APOE_GENOTYPE
● CATEGORICAL ○ AS IS ○ EXCLUDE

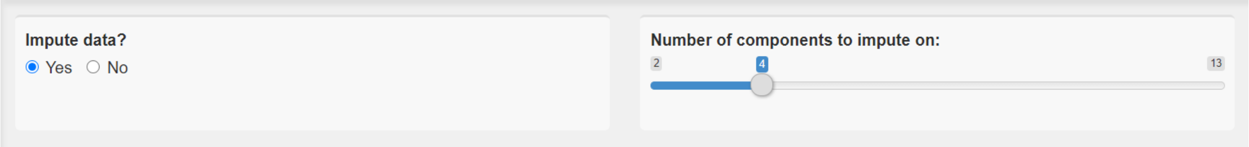
MAPT_DIPLOTYPE
● CATEGORICAL ○ AS IS ○ EXCLUDE

Stroop_color_sec
○ ORDINAL ● NUMERIC ○ AS IS ○ EXCLUDE

Figure 7: The app will predetermine which data types are valid for each variable.

3.7 Perform imputation

Imputation represents the process of handling missing data with substituted values, and it is the last step in the pipeline and is optional. Imputation will differ depending on whether the data is strictly continuous or not strictly continuous (which is a mix of numeric, categorical and/or ordinal variables), which the app will automatically determine and implement. The default value of the slider indicates the optimal number of components to impute on. Once the inputs are confirmed, a preview of the data frame after imputation is outputted for user viewing. To update the data type after confirmation, please click the reset button.



Impute data?
● Yes ○ No

Number of components to impute on:
2 4 13

Figure 8: This particular example shows 4 as the optimal number of components to impute on, with 2 and 13 representing the lower and upper bound thresholds respectively. These values are precalculated by the app.

3.8 Export into global environment

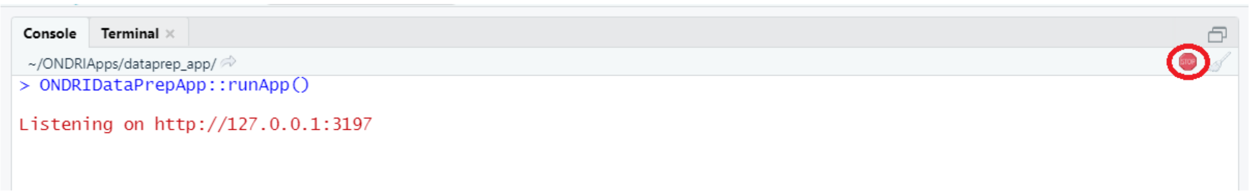
Once imputation is complete, please click the "Export into global environment" button, which will export up to four data frames into the RStudio global environment. They are the following:

1. **DF__DROPPED__ROWS:** This data consists of the rows/participants that were dropped when defining missingness thresholds (in step 5 of the pipeline). If no rows/participants were dropped, this data frame won't be exported.
2. **DF__DROPPED__COLS:** This data consists of the columns/variables that were dropped when defining missingness thresholds (in step 5 of the pipeline). If no columns/variables were dropped, this data frame won't be exported.

3. **DF__EXCLUDED__COLS:** This data consists of the columns/variables that were excluded when defining variable types (in step 6 of the pipeline). If no columns/variables were excluded, this data frame won't be exported.
4. **DF__PREPPED__DATA:** This data consists of the final output after the original data has gone through the entire pipeline including missingness drops, transformations, and imputation. This data frame will always be exported.

All data frames will contain the four standard ONDRI columns/variables, which are SUBJECT, VISIT, *_SITE, and *_DATE.

IMPORTANT: Once you receive a success message, please close the app and click the stop button in the RStudio console as shown in Figure 9. You should see the new data frames in your global environment in the top right pane.



```
~/ONDRIApps/dataprep_app/
> ONDRIDataPrepApp::runApp()

Listening on http://127.0.0.1:3197
```

Figure 9: Please click the stop button that is circled in red once you close the app.

3.9 Extra notes and navigation

The left menu allows for previous steps in the pipeline to be accessed in a user-friendly manner. However, once you return to a particular step, subsequent steps may be inaccessible from the menu as changes made in the current step would affect the remainder of the pipeline. If this is the case, the pipeline from the current step onward would be reset.

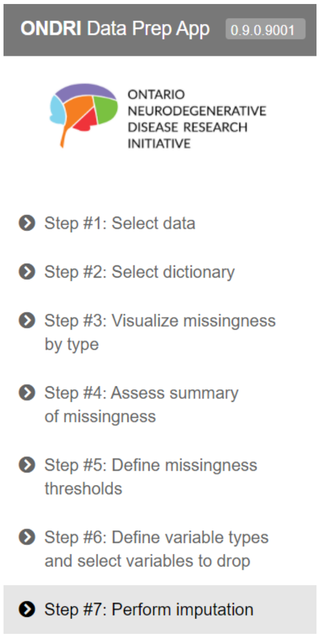

ONDRI Data Prep App 0.9.0.9001
ONTARIO NEURODEGENERATIVE DISEASE RESEARCH INITIATIVE
Step #1: Select data
Step #2: Select dictionary
Step #3: Visualize missingness by type
Step #4: Assess summary of missingness
Step #5: Define missingness thresholds
Step #6: Define variable types and select variables to drop
Step #7: Perform imputation

Figure 10: This menu is used to return to previous steps in the pipeline.

4 Program Limitations

4.1 Non-numeric ordinal variables

As mentioned previously, the app will only accept ordinal variables that are coded as a numerical based ordinal scale. If the variable values are character based, they will have to be converted to numeric and ordered values manually before running the app.

4.2 OuRS package version discrepancy

The app requires the latest version of OuRS (https://github.com/derekbeaton/OuRS). Otherwise, it will crash at the transformation step of the pipeline.

4.3 Runtime on large datasets

The app will run slower when feeding in large datasets such as NIMG DTI. Please be patient as the contents of the user interface will take up to thirty seconds to load in each step of the pipeline.

5 Development, issues, and updates

As the app is currently undergoing beta testing, patches will be pushed to GitHub as bugs and suggestions come in. Curators, RAs, and students will be notified when major updates become available. The version number will be updated accordingly in the app, as shown in Figure 11.

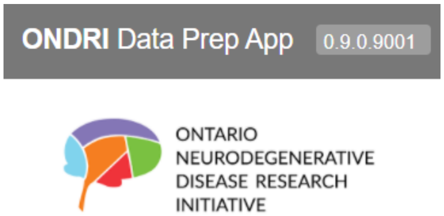


Figure 11: The version number will indicate whether you are using the latest version of the app.

For any bug reports or enhancement suggestions, please add an issue to the GitHub repository (https://github.com/ondri-nibs/dataprep_app/issues) so they can be tracked. Please assign Jedid Ahn, select no milestone, and label it as either a bug (red) or an enhancement (cyan).

If you require any clarification or have technical difficulties with the app, please contact Jedid Ahn at jahn@research.baycrest.org and cc Derek at dbeaton@research.baycrest.org.