ONTARIO
NEURODEGENERATIVE
DISEASE RESEARCH
INITIATIVE

Complete Standards App - Reference Guide

On behalf of the Neuroinformatics and Biostatistics (NIBS) platform

August 18, 2020

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

Contents

1 Introduction and Set-up

1.1 Overview

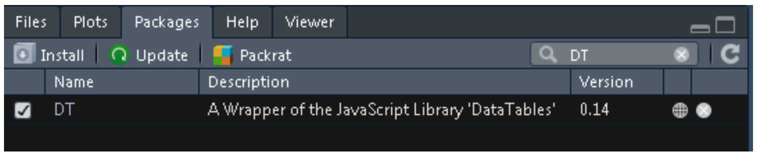
The purpose of the ONDRI Complete Standards App is to run a standards check on the required components of a data package: the DATA file, DICT file, README file, METHODS file, and any additional files (e.g., MISSING file). The app will point to the precise location of the issue in files and throughout the package, allowing for curators to quickly identify where to make the fixes and adjustments prior to formal standards checks and onto outlier analysis by the Neuroinformatics and Biostatistics (NIBS) team.

1.2 Installation

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 Git (again please choose the correct installer carefully). During the installation process, you can leave all installation options to their default and original configuration. However, you can download Git in a different folder path if you wish.
3. Open RStudio and run the following commands in the RStudio console (the bottom left pane) to install the necessary packages if you have not already done so:

```
install.packages(xml2)
install.packages(shiny)
install.packages(tools)
install.packages(shinyalert)
install.packages(V8)
install.packages(stringr)
install.packages(plyr)
install.packages(shiny)
install.packages(DT)
install.packages(data.table)
install.packages(lubridate)
install.packages(shinyalert)
install.packages(shinyWidgets)
install.packages(shinyFiles)
install.packages(shinyjs)
install.packages(dplyr)
install.packages(rowr)
install.packages(varhandle)
install.packages(lubridate)
install.packages(readr)
```

4. You will know that the packages listed above are loaded if the checkbox beside each in the Packages tab are checked. Add the checkmark to each for any that are missing from the necessary packages.



	Name	Description	Version
✓	DT	A Wrapper of the JavaScript Library 'DataTables'	0.14

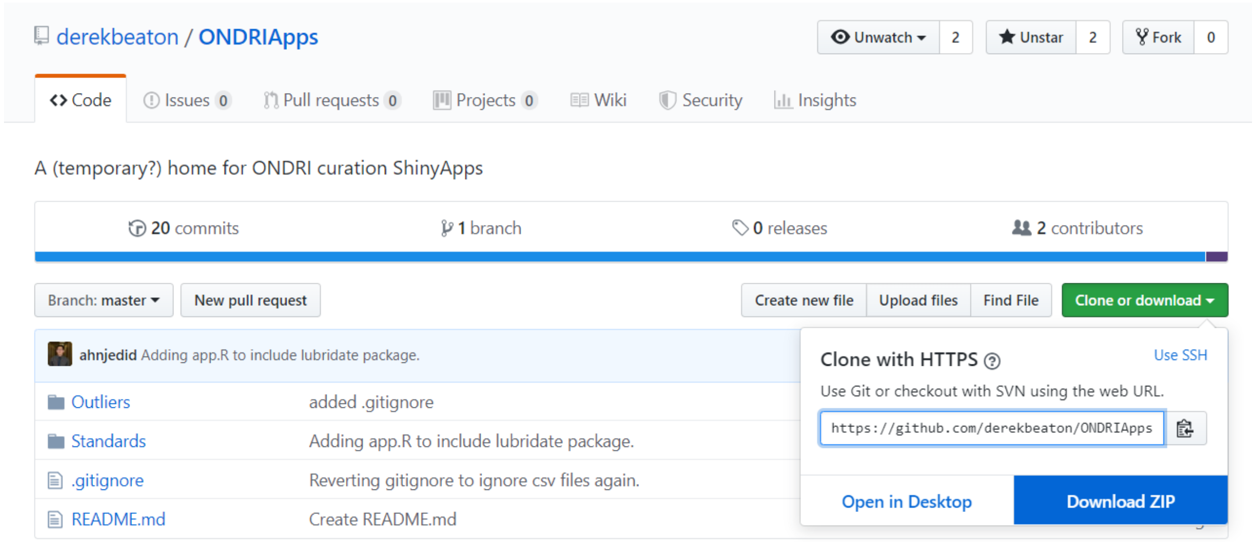
Figure 1: The "Packages" tab is located in the bottom right pane of RStudio.

1.3 Deployement

There are 2 ways to deploy the app: the first way is through downloading the repository directly off of GitHub and the second way is through Git Bash, We describe both options below.

Option 1: Download Repository

1. Go to the following link: https://github.com/derekbeaton/ONDRIApps
2. Click the green "Clone or download" button, and then click "download ZIP".



A (temporary?) home for ONDRI curation ShinyApps

20 commits · 1 branch · 0 releases · 2 contributors

ahnjedid Adding app.R to include lubridate package.

Outliers	added .gitignore
Standards	Adding app.R to include lubridate package.
.gitignore	Reverting gitignore to ignore csv files again.
README.md	Create README.md

Use Git or checkout with SVN using the web URL.

https://github.com/derekbeaton/ONDRIApps

Figure 2: Please note that the repository is on Derek's personal GitHub profile.

3. Browse to the directory that you saved the ZIP folder in, right click on the folder, and click "Extract All". Extract into a separate folder (not in the same directory as the ZIP folder). You may delete the ZIP folder afterwards.

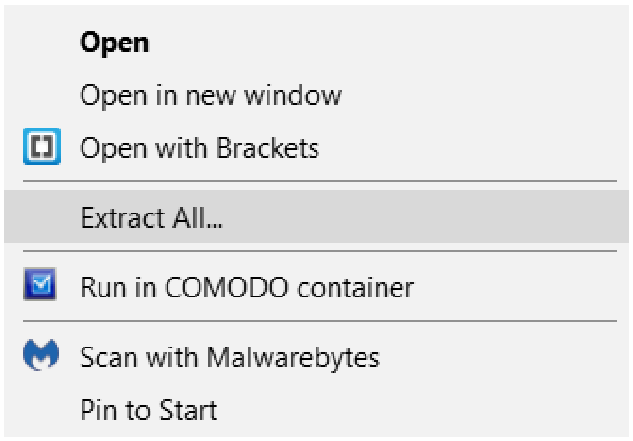


Figure 3: Please note that your drop down may look different (depending on operating system and versions of RStudio)

Option 2: Git Bash

1. Right click on the folder that you wish to store the repository/program files in, and click on "Git Bash Here". The Git Bash application will pop up.

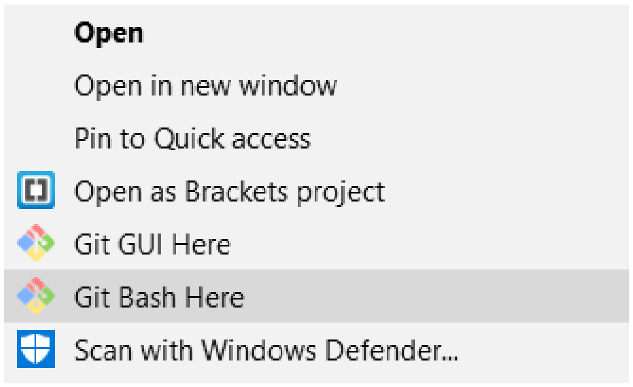
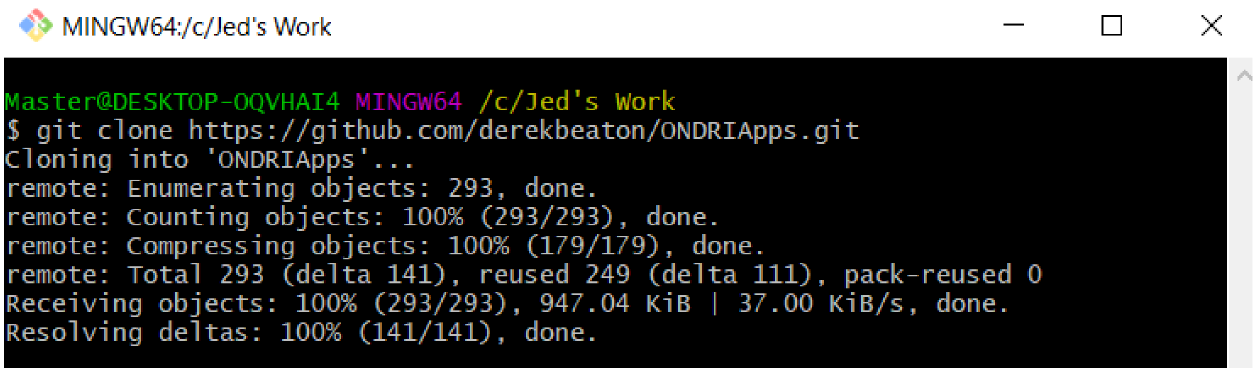


Figure 4: Please note that your drop down may look different (depending on operating system and versions of RStudio)

2. Clone the repository through Git by running the following command:

```
git clone https://github.com/derekbeaton/ONDRIApps.git
```



```
Master@DESKTOP-OQVHAI4 MINGW64 /c/Jed's Work
$ git clone https://github.com/derekbeaton/ONDRIApps.git
Cloning into 'ONDRIApps'...
remote: Enumerating objects: 293, done.
remote: Counting objects: 100% (293/293), done.
remote: Compressing objects: 100% (179/179), done.
remote: Total 293 (delta 141), reused 249 (delta 111), pack-reused 0
Receiving objects: 100% (293/293), 947.04 KiB | 37.00 KiB/s, done.
Resolving deltas: 100% (141/141), done.
```

Figure 5: The Git Bash terminal, which you can use only if you have installed Git.

Next steps after choosing Option 1 or Option 2:

1. Open RStudio, go to File -> Open Project, go to the "Joint-APP" folder within the repository, and double click code (code.Rproj).

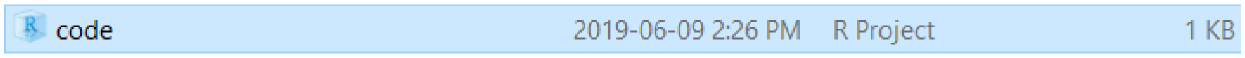


Figure 6: code.Rproj will be evident by the RStudio symbol on the left.

2. Open app.R in the RStudio file interface.
3. Click the drop down of "Run App" in the top right of the file interface and select "Run External".

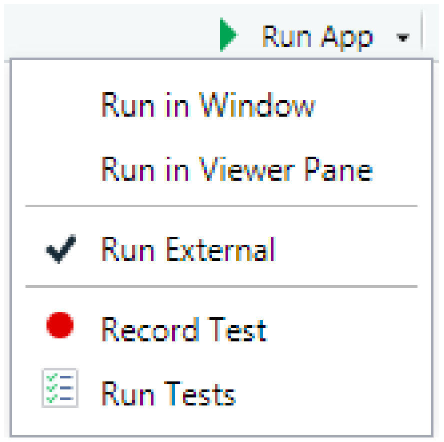


Figure 7: To open the drop down, click the small (upside down) black triangle to the right of "Run App"

4. Click "Run App" or type

```
shiny::runApp()
```

in the command line.

2 Using The App

Outlined in this section are details to prepare the app for running standards on a data package.

2.1 Preparation

1. Ensure that all files of the data package are in the same folder on your local computer. In addition, please download the participant IDs file into any folder. Ensure you have the participant ID file relating to the appropriate study for your data.
2. In the Standards App, click on the button "Click to select a folder.", and browse to the folder that you created. Then click the blue "Select" button in the bottom right corner.

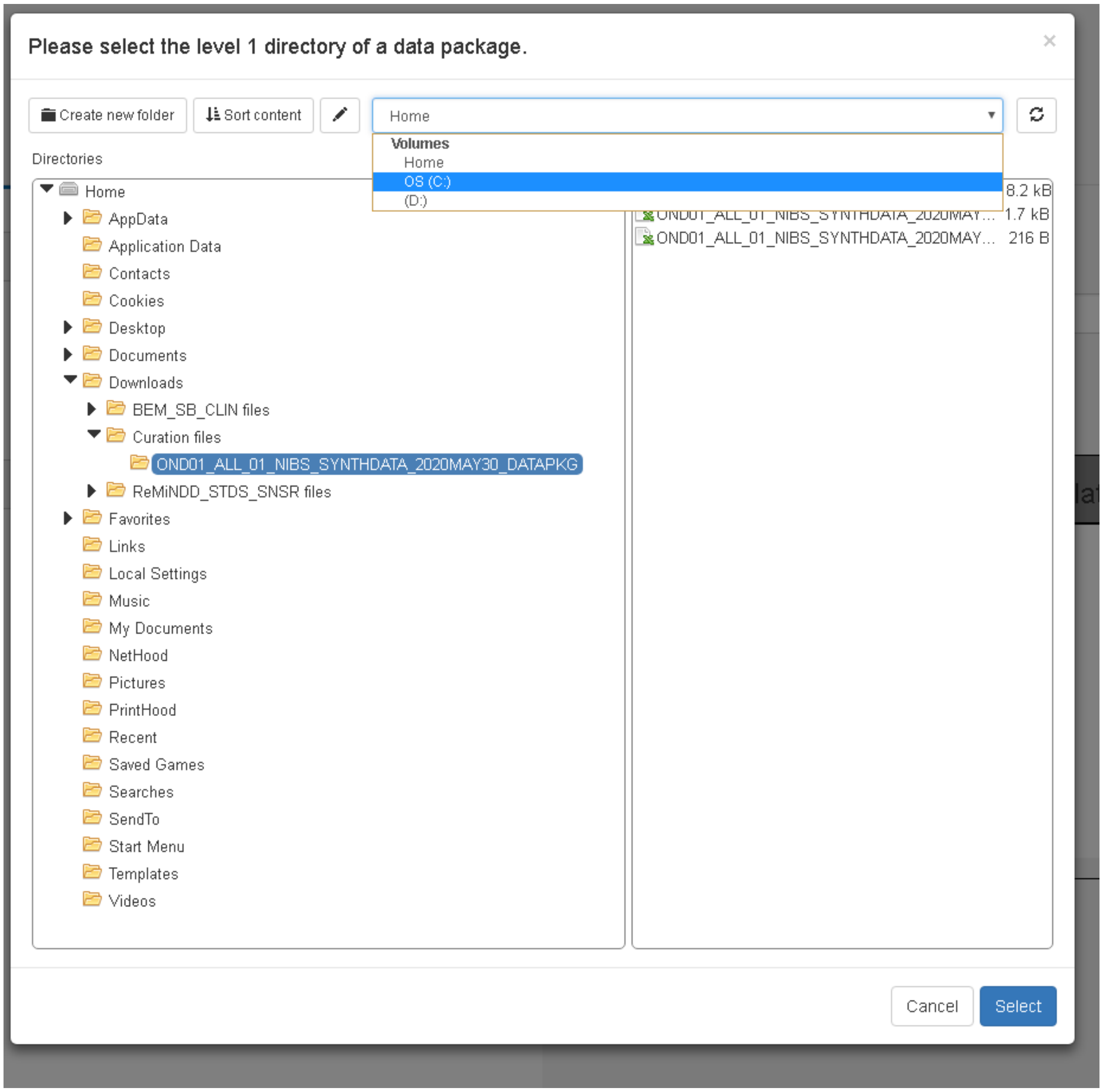
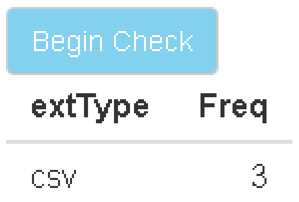


Figure 8: This window gives you access to both your home directory and your C: drive (on Windows).

3. The right-side panes will provide information about the loaded files:

NOTE: Once the package is selected you can view what file extensions and their occurence in the table which appears below the begin check button.



extType	Freq
csv	3

Figure 9: The table will show you number of occurences for the different file extensions found in your data package.

IMPORTANT: The default directory chosen is your home directory. If you wish to access the C: drive (on Windows), please use the dropdown as shown below.

4. Click on the button "Click to select an ID file.", and browse to the folder that contains the ID file. Select the ID file and then click the blue "Select" button in the bottom right corner.
5. Once you have chosen the files you want to run through the app you need to chose the appropriate study (under "Study Selection") corresponding to the data you will be checking. If the data you are running through the app requires different code checks from the quick select options, you can chose the "Custom" option under Select Study and add the codes you wish to check in the data package. If you would like to check your data package for non-specific items (such as blank cells, whitespace...) you can select the "Package only checks".

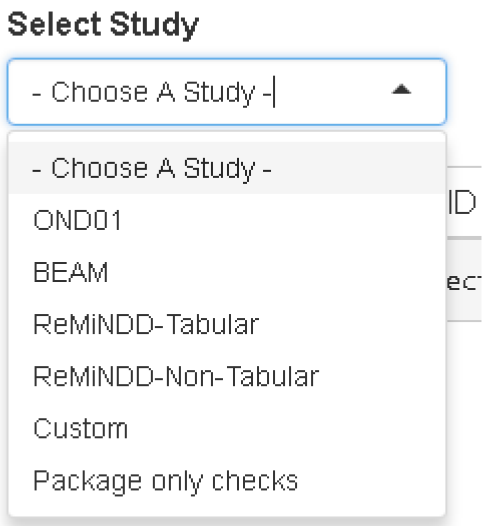


Figure 10: Study selection drop down menu.

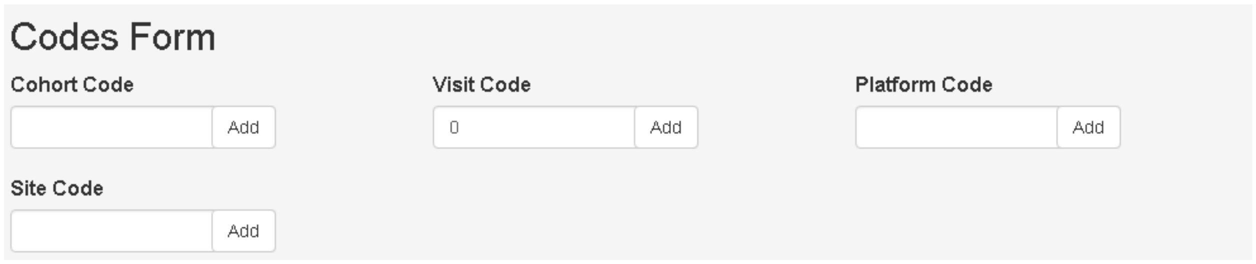
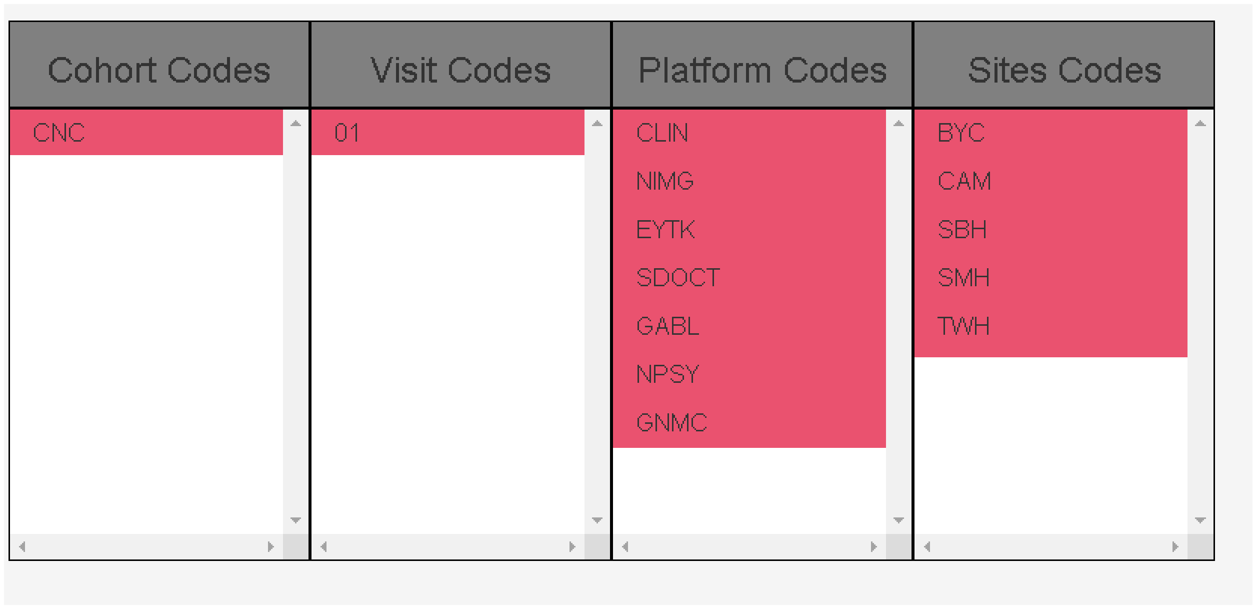


Figure 11: In the top right side of the screen you can add custom study codes to be checked.

NOTE: You can view the codes that will be checked during the checking process in the table on the right side of the screen once a study has been chosen.



Cohort Codes	Visit Codes	Platform Codes	Sites Codes
CNC	01	CLIN	BYC
		NIMG	CAM
		EYTK	SBH
		SDOCT	SMH
		GABL	TWH
		NPSY	
		GNMC	

Figure 12: The codes input table is found on the right side of the screen and it will let you preivew the codes that will be checked during the process.

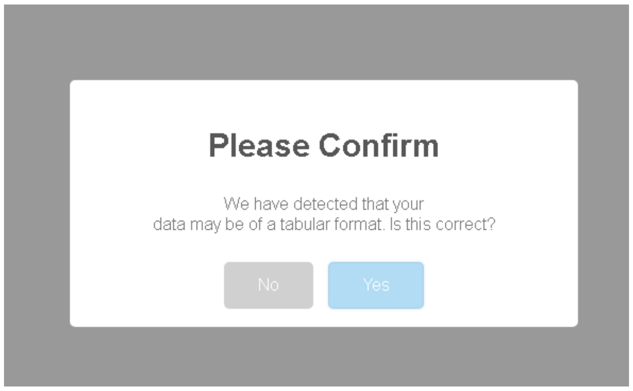
IMPORTANT: The app needs to have at least one item in all of the codes fields in the table to be able to run.

Congratulations, you are ready to start running standards on your data package!

2.2 Running Standards

1. Now that you are ready to check your data package click the “Begin Check” button.
2. When the button is pressed a pop up will appear asking you to confirm that your data is tabular or non-tabular. If you have been directed to the wrong type (tabular/non-tabular) press “NO” and you

will be redirected to the select page (at this point you will need to see if you have accidently selected the wrong package or if your tabular package contains sub directories causing the app to identify it as a non-tabular format). If it is the correct type, select “YES” to continue.


Please Confirm

We have detected that your data may be of a tabular format. Is this correct?

Figure 13: Example of a pop up when the program detects a tabular package.

3. Select the Standards you wish to check on the left hand side of the screen (By default, all standards are checked). Then press the appropriate buttons at the bottom of the screen to check the different files within the data package.
4. For any checks that provide location of errors (such as blank cells), the precise locations are given through:

- Column titles and subject IDs in the DATA & MISSING file
- Column titles and column labels in the DICT file
- Column titles and file names in the README file

The dropdown next to the green progress bar displays the standard that has been violated and the column title where this violation occurs. The subject IDs, column labels, or file names that are listed in the right pane are associated with the violated values for this particular column.

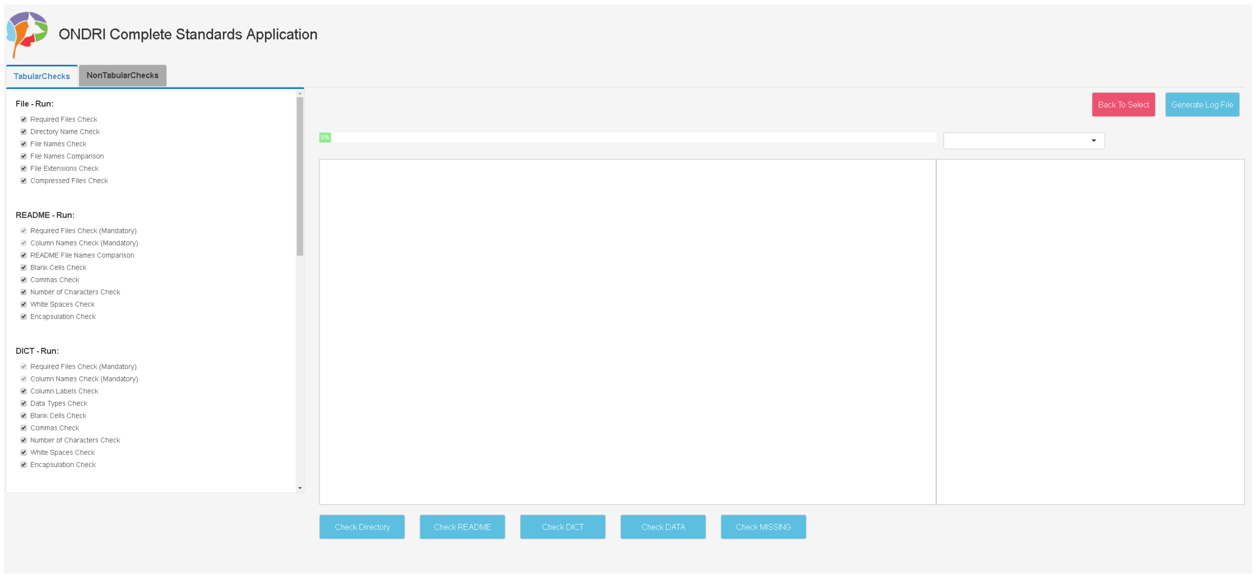
5. For Non-Tabular data you now press on the different tabs “Level1Checks”, “Level2Checks and”Level3Checks" (found at the top of the screen) and repeat the same process described in step 3 and 4. This will check the sub folders of your data package.

6. Once you have performed all the checks you wanted to run. Press the “Generate Log File” button to generate a log file of all the checks that were run during the process. Choose the location where this file is to be saved.

NOTE: At any point during the process you can select the “Back To Select” button to return to the select screen. This will reset the app and all previous selections will be reset.

2.3 Program Limitations

Please note that a data package can pass standards despite errors when running the app. The following are common errors that get flagged and should be ignored in most cases:



File - Run:

- ✓ Required Files Check
- ✓ Directory Name Check
- ✓ File Names Check
- ✓ File Names Comparison
- ✓ File Extensions Check
- ✓ Compressed Files Check

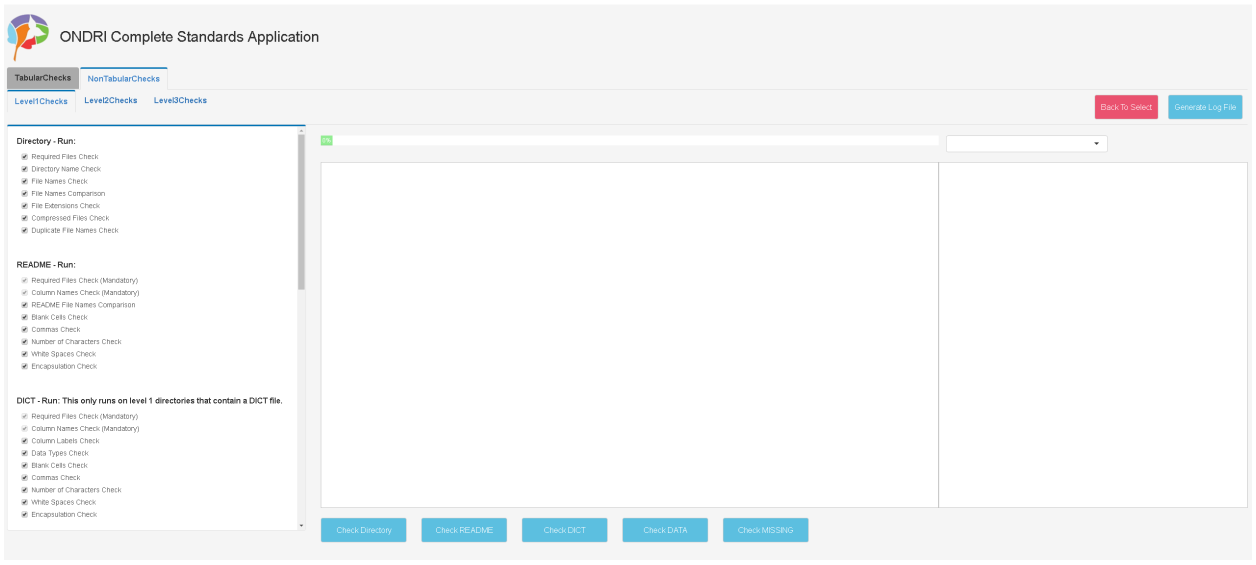
README - Run:

- ✓ Required Files Check (Mandatory)
- ✓ Column Names Check (Mandatory)
- ✓ README File Names Comparison
- ✓ Blank Cells Check
- ✓ Commas Check
- ✓ Number of Characters Check
- ✓ White Spaces Check
- ✓ Encapsulation Check

DICT - Run:

- ✓ Required Files Check (Mandatory)
- ✓ Column Names Check (Mandatory)
- ✓ Column Labels Check
- ✓ Data Types Check
- ✓ Blank Cells Check
- ✓ Commas Check
- ✓ Number of Characters Check
- ✓ White Spaces Check
- ✓ Encapsulation Check

Figure 14: Interface for tabular packages.



Directory - Run:

- ✓ Required Files Check
- ✓ Directory Name Check
- ✓ File Names Check
- ✓ File Names Comparison
- ✓ File Extensions Check
- ✓ Compressed Files Check
- ✓ Duplicate File Names Check

README - Run:

- ✓ Required Files Check (Mandatory)
- ✓ Column Names Check (Mandatory)
- ✓ README File Names Comparison
- ✓ Blank Cells Check
- ✓ Commas Check
- ✓ Number of Characters Check
- ✓ White Spaces Check
- ✓ Encapsulation Check

DICT - Run: This only runs on level 1 directories that contain a DICT file.

- ✓ Required Files Check (Mandatory)
- ✓ Column Names Check (Mandatory)
- ✓ Column Labels Check
- ✓ Data Types Check
- ✓ Blank Cells Check
- ✓ Commas Check
- ✓ Number of Characters Check
- ✓ White Spaces Check
- ✓ Encapsulation Check

Figure 15: Interface for non-tabular packages.

1. Blank Cells: Flagging for blank cells always appears in the DICT file, as the VALUES column for SUBJECT and DATE column labels are usually blank. In addition, blank cells are allowed when a column is dependent on the values of another column. (Ex: If yes, explain why. If no, leave as blank.)
2. Encapsulation: Single quotes are allowed when used as an apostrophe.

The following are errors that pop up often, but should not be ignored. Rather, they are flagged to remind the curator and/or double check any values.

1. Missing codes M_TBC and M_OTHER are flagged as requiring special permission from Neuroinformatics. Please ensure that you receive this permission.
2. Data type MIXED is flagged as requiring special permission from Neuroinformatics. Please ensure that you receive this permission.
3. Date range: Date range is flagged to verify out of window dates. Once the dates have been verified, these messages can be ignored.
4. Number of characters: Any cells with over 200 characters are flagged and this is most common in the DICT file. While this is not a strict rule, please ensure that sentences are concise. After double checking, it is okay to ignore these messages.
5. IMPORTANT: Commas are flagged often, and should not be ignored. All commas need to be removed and/or replaced with semicolons, as the file itself is comma separated.

Adding on, the app cannot check for spelling or grammatical mistakes, nor can it flag for dataset dependent errors such as adding up all values in columns and matching with a TOTAL_SUM column.

IMPORTANT: Please note that a data package always has to be submitted to standards on LabKey. This is because the Neuroinformatics team does additional checks that the app cannot check for (such as variable naming).

3 Future Developement

Potential features to look forward to:

- The ability to upload codes spread sheet which will auto fill the codes input table.
- The ability to save codes inputed in the codes input table.