BIDSconvertR

Release 0.0.1

Niklas Wulms

Dec 07, 2022

OVERVIEW:

The BIDSconvertR R-package converts MRI DICOM data into BIDS-specification.

Note: We welcome feedback on usability or new features for the BIDSconvertR, which is still in development.

CHAPTER

ONE

AIM

The goal of BIDSconvertR is to provide a workflow that can:

- open the BIDS conversion to both beginner and advanced R-users
- convert DICOM to NIfTI data using dcm2niix (https://github.com/rordenlab/dcm2niix)
- structure this data according to the BIDS specification (https://bids-specification.readthedocs.io/en/stable/)
- validate the manually entered sequence-ID's by color-coding
- enable easy access to the BIDS-Validator (Website/Docker) (https://bids-standard.github.io/bids-validator/)
- provide the papayaWidget viewer (https://github.com/muschellij2/papayaWidget) for inspecting images
- enable continuous application during data acquisition in ongoing studies

CHAPTER TWO

FEATURES

- **Creation of a 'user_settings' file that**
  - stores all settings and
  - can be used to relaunch the conversion in case of new acquired data in ongoing studies
- **Optional renaming of subject-ID's or session-ID's**
  - Extraction of the subject-ID's
  - Removal of redundant strings
- **Conversion with DCM2NIIX (Chris Rorden)**
  - Anonymization of metadata for the BIDS Folder
  - Extraction of metadata with potentially identifying information that is stored in a separate folder
- **Shiny-App for BIDS sequence editing**
  - Color-coded BIDS validation for manually entered sequence-ID's
  - Selection of 'relevant' sequences for the BIDS folder
- **Automatic start of BIDS-Validator**
  - in Docker if it is installed or
  - on the BIDS-Validator homepage
- **Shiny-App for navigating through the BIDS folder**
  - image viewing and visual quality control

When new files or sequences are added, the'sequence mapper' determines whether they are new. Then it reopens until everything is declared in accordance with BIDS. Files that have already been processed are bypassed.

CHAPTER

THREE

TECHNICAL REQUIREMENTS

- Supported and tested on Microsoft 10 and Ubuntu 22.04.
- Not tested on MacOS, but it should work. You could try it out and let me know if there are any problems.

CHAPTER

FOUR

WHAT THE USER NEEDS TO KNOW TO APPLY THE BIDSCONVERTR

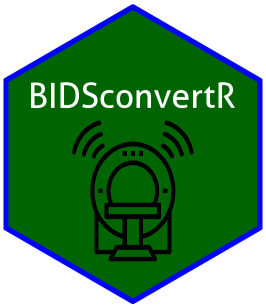
Important: What it isn't:

A fully automated workflow that takes care of everything. You must have a basic understanding of files, folders, and DICOMS, as described below.

- What exactly is a folder or file? What happened to my data?
- What is a DICOM-containing folder?
- What is the difference between a subject- and sequence-ID?
- What exactly is BIDS?
- How to create valid BIDS sequence names?

So, if you can manually rename and restructure the folders according to BIDS, you can run the tool to scale things up.

4.1 BIDSconvertR



The hexagonal sticker was made using the iconspackage and based on the MRI svg graphics provided by Flaticon and mavadee FlaticonLink.

4.1.1 Aim

BIDSconvertR aims to provide a workflow that can:

- do the task inside of the R environment
- convert DICOM data to NIfTI (with the awesome dcm2niix
- structure this data according to the BIDS specification

4.1.2 Features

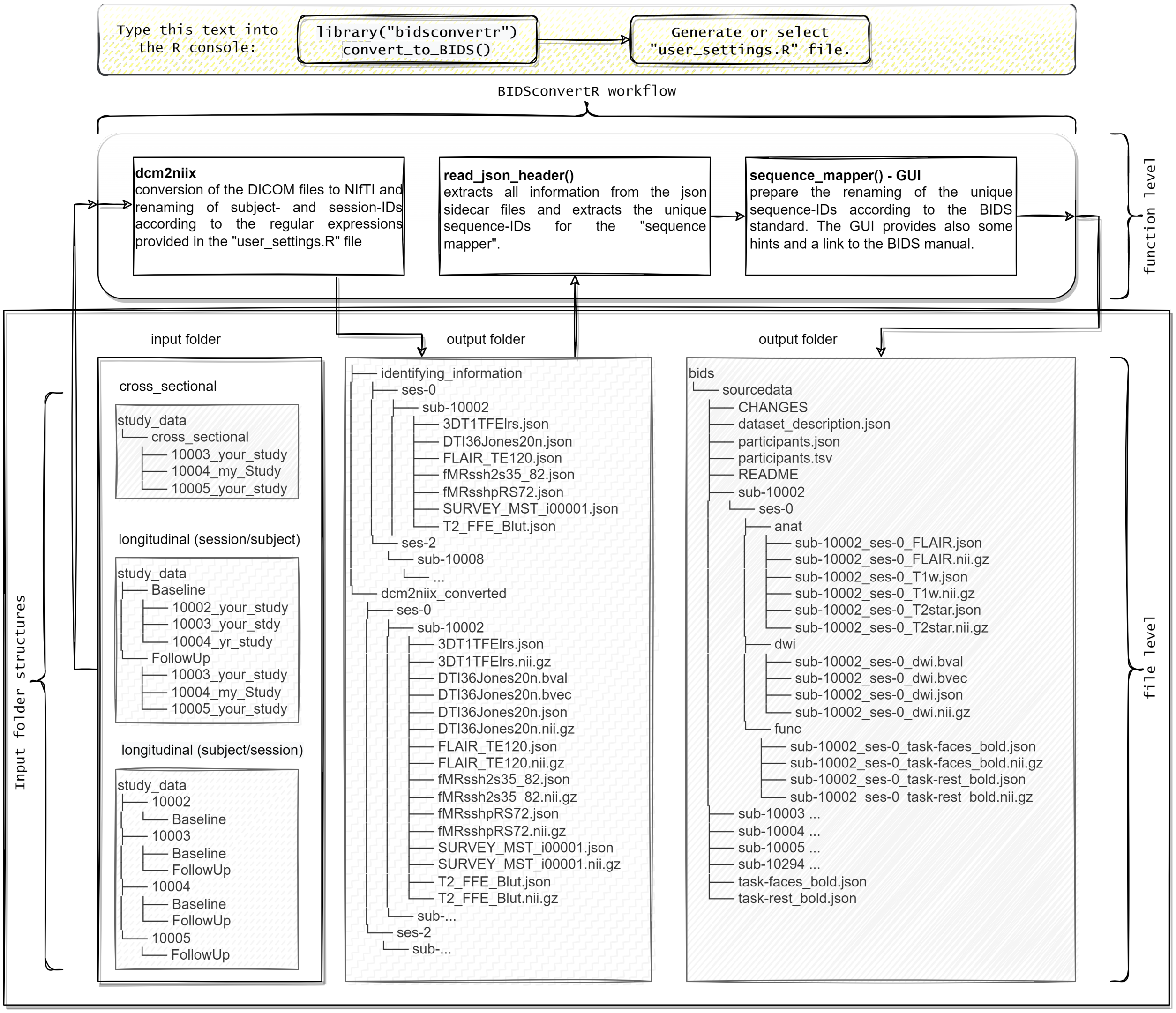
- continuous application
  - lazy processing of already existing files
  - easy application during data collection in ongoing studies
- user-friendliness (minimial terminal interaction required)
  - user dialog with message boxes guiding the users through the workflow
  - Shiny App (GUI) for sequence editing and data inspection
- file cleaning
  - Renaming of subject-IDs or session-IDs with regular expressions.
  - Renaming of session-IDs
- BIDS validation
  - verification (color-coded) of sequence-IDs (comparing the entered sequence-IDs to regular expressions according to BIDS)
  - implemented validation with BIDS-Validator (Website/Docker)
- quality control
  - user-friendly (papayaWidget image viewer) for BIDS datasets
- pseudonymized BIDS output (only metadata)
  - all potentially identifiable metadata was removed from images and JSONs

Note: Only the metadata contained within the BIDS folder is free of potentially identifiable information. Follow the legal terms when sharing your dataset and consider additional defacing, pseudonymization, or both.

4.1.3 Milestones / To-Do's

- publish the BIDSconvertR
- testing the tool on MAC systems
- create Docker container
- move image viewer into separate package
- adapt code to CRAN requirements

4.2 Workflow visualization



```
library("bidsconvertr")
convert_to_BIDS()
```

dcm2niix
conversion of the DICOM files to NIfTI and renaming of subject- and session-IDs according to the regular expressions provided in the "user_settings.R" file

read_json_header()
extracts all information from the json sidecar files and extracts the unique sequence-IDs for the "sequence mapper".

sequence_mapper() - GUI
prepare the renaming of the unique sequence-IDs according to the BIDS standard. The GUI provides also some hints and a link to the BIDS manual.

cross_sectional

```
study_data
└── cross_sectional
    ├── 10003_your_study
    ├── 10004_my_Study
    └── 10005_your_study
```

longitudinal (session/subject)

```
study_data
├── Baseline
│   ├── 10002_your_study
│   ├── 10003_your_stdy
│   └── 10004_yr_study
└── FollowUp
    ├── 10003_your_study
    ├── 10004_my_Study
    └── 10005_your_study
```

longitudinal (subject/session)

```
study_data
├── 10002
│   └── Baseline
├── 10003
│   ├── Baseline
│   └── FollowUp
├── 10004
│   ├── Baseline
│   └── FollowUp
└── 10005
    └── FollowUp
```

```
├── identifying_information
│   ├── ses-0
│   │   ├── sub-10002
│   │   │   ├── 3DT1TFElrs.json
│   │   │   ├── DTI36Jones20n.json
│   │   │   ├── FLAIR_TE120.json
│   │   │   ├── fMRssh2s35_82.json
│   │   │   ├── fMRsshpRS72.json
│   │   │   ├── SURVEY_MST_i00001.json
│   │   │   └── T2_FFE_Blut.json
│   └── ses-2
│       └── sub-10008
│           └── ...
└── dcm2niix_converted
    ├── ses-0
    │   ├── sub-10002
    │   │   ├── 3DT1TFElrs.json
    │   │   ├── 3DT1TFElrs.nii.gz
    │   │   ├── DTI36Jones20n.bval
    │   │   ├── DTI36Jones20n.bvec
    │   │   ├── DTI36Jones20n.json
    │   │   ├── DTI36Jones20n.nii.gz
    │   │   ├── FLAIR_TE120.json
    │   │   ├── FLAIR_TE120.nii.gz
    │   │   ├── fMRssh2s35_82.json
    │   │   ├── fMRssh2s35_82.nii.gz
    │   │   ├── fMRsshpRS72.json
    │   │   ├── fMRsshpRS72.nii.gz
    │   │   ├── SURVEY_MST_i00001.json
    │   │   ├── SURVEY_MST_i00001.nii.gz
    │   │   ├── T2_FFE_Blut.json
    │   │   └── T2_FFE_Blut.nii.gz
    │   └── sub-...
    └── ses-2
        └── sub-...
```

```
bids
└── sourcedata
    ├── CHANGES
    ├── dataset_description.json
    ├── participants.json
    ├── participants.tsv
    ├── README
    ├── sub-10002
    │   └── ses-0
    │       ├── anat
    │       │   ├── sub-10002_ses-0_FLAIR.json
    │       │   ├── sub-10002_ses-0_FLAIR.nii.gz
    │       │   ├── sub-10002_ses-0_T1w.json
    │       │   ├── sub-10002_ses-0_T1w.nii.gz
    │       │   ├── sub-10002_ses-0_T2star.json
    │       │   └── sub-10002_ses-0_T2star.nii.gz
    │       ├── dwi
    │       │   ├── sub-10002_ses-0_dwi.bval
    │       │   ├── sub-10002_ses-0_dwi.bvec
    │       │   ├── sub-10002_ses-0_dwi.json
    │       │   └── sub-10002_ses-0_dwi.nii.gz
    │       └── func
    │           ├── sub-10002_ses-0_task-faces_bold.json
    │           ├── sub-10002_ses-0_task-faces_bold.nii.gz
    │           ├── sub-10002_ses-0_task-rest_bold.json
    │           └── sub-10002_ses-0_task-rest_bold.nii.gz
    ├── sub-10003 ...
    ├── sub-10004 ...
    ├── sub-10005 ...
    ├── sub-10294 ...
    ├── task-faces_bold.json
    └── task-rest_bold.json
```

4.3 Citation

Wulms, Niklas. 2022. *Wulms/Bidsconvertr*. Zenodo. https://doi.org/10.5281/ZENODO.5878407.

4.4 Install R, dependencies and RStudio

All code was tested with R Version 4.2.1, RTools42, and Tidyverse V 1.3.2.

4.4.1 Windows

Download and install a recent R version (V 4.2.1).

Download and install Rtools42 according to your R-version (V 4.2.1). It is required to install the packages.

Download and install RStudio.

4.4.2 Debian/Ubuntu

To install R and its dependencies, enter the following commands into the terminal.

```
wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc | sudo gpg --
↪dearmor -o /usr/share/keyrings/r-project.gpg

echo "deb [signed-by=/usr/share/keyrings/r-project.gpg] https://cloud.r-project.org/bin/
↪linux/ubuntu jammy-cran40/" | sudo tee -a /etc/apt/sources.list.d/r-project.list

sudo apt update

sudo apt install --no-install-recommends r-base

sudo apt install r-base-dev

sudo apt install libcurl4-openssl-dev libssl-dev libxml2-dev libfontconfig1-dev
```

Install RStudio.

Then "Right-Click -> Open With ->Software Install" should help in the installation.

4.4.3 Mac

We need some collaborators, who try out the BIDSconvertR on a Mac system. Contact me, if you are interested.

4.5 Install the BIDSconvertR

Note: Please install R in accordance with the instructions. From here, copy & paste the following commands into RStudio's 'console' panel.

Known issues:

- If you are asked to install the package from binary source files select "yes".
- If you are asked to update packages, you can skip this step via '3' or update the packages via '1' or '2'.
- If there is a 'installation of package 'devtools' had non-zero exit status' error message, identify the package (here 'devtools') with the error message and install it manually via 'install.packages("package_name")'.

Both packages must be installed only once. The `devtools` package is required to install packages from Github.

```
install.packages("shiny")
install.packages("devtools")
```

Using the command below, you can now install the most recent development version of `BIDSconvertR`.

```
devtools::install_github(repo = "wulms/bidsconvertr")
```

4.6 Requirements

4.6.1 R-environment

- R (4.2.1)
- RStudio (recent version)
- R packages installed:
  - 'devtools'
  - 'shiny'
  - 'bidsconvertr'

4.6.2 Input data

The input data must have the following structure:

- an input folder containing all folders with DICOM data
- .../subjects/sessions/DICOM
- one folder per subject, for example: "00001", "00002"
- these folders containing the session data, each of which contains the DICOM data
- .../sessions/subjects/DICOM
- cross-sectional (at least one folder named 'crosssectional' or a custom session name)
- one folder per session, for example: "session_1", "session_2"
- these folders, which contain DICOM data in separate folders for each subject, e.g. "00001", "00002"
- if you have any additional file structures, please contact me, so that I can include them.

4.7 The installation procedure

Installing R and RStudio and downloading sample data:

- To install the *BIDSconvertR* on Windows or Linux follow the *instructions*.
- Download the BIDScoin example data : Download here. Please be aware that the data has been compressed twice using gunzip (suffix: '.gz') and tar (suffix: '.tar'). You need to unpack both. Some programs do this in one step, others need to extract it twice. The data is fine, when you can easily click down the folder structure.

4.8 The BIDSconvertR workflow

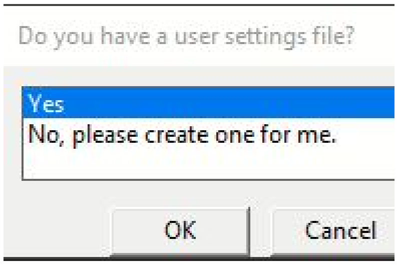
This procedure should demonstrate how to use the BIDSconvertR. We appreciate the BIDScoin team's permission to use their sample data. This tutorial only covers the basic steps. You can find a detailed description including visualizations in the "workflow" section.

This workflow covers each stage of the BIDSconvertR workflow:

- Launch RStudio and use the R console from here on.
- Execute `library("bidsconvertr", quietly = TRUE)`.
- Use the `convert_to_BIDS()`command to launch the tool.

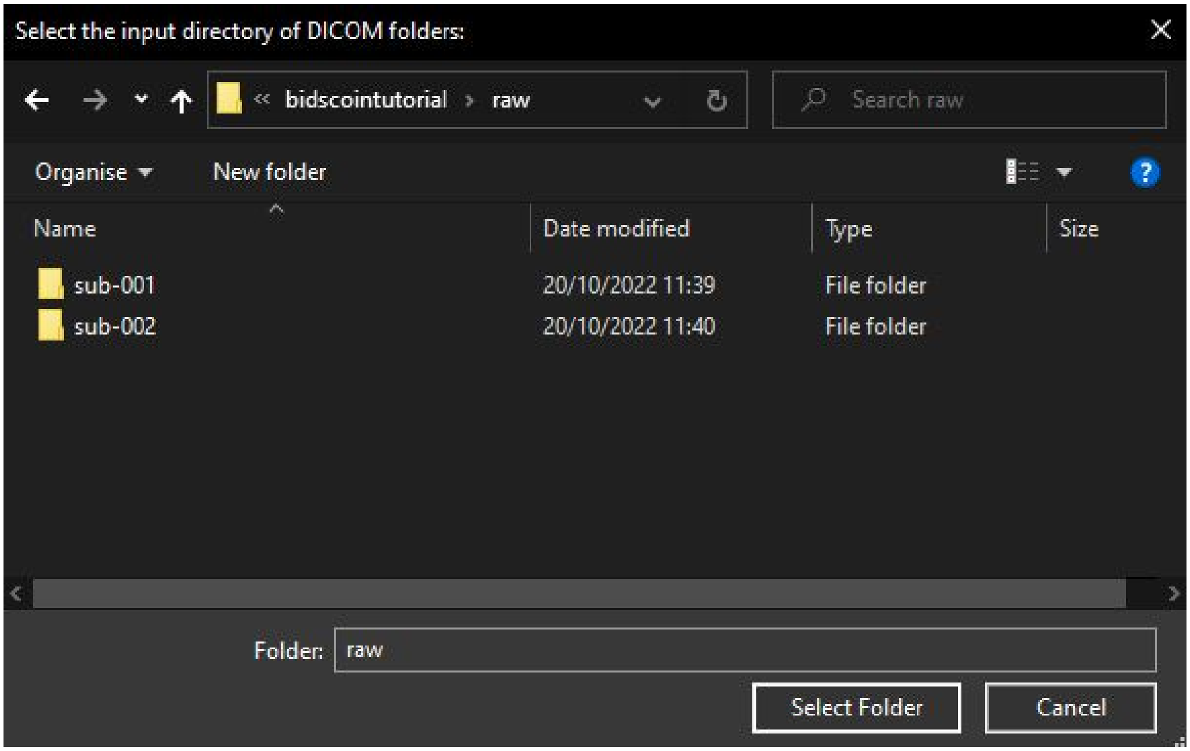
4.8.1 Creation of 'user_settings.R' file.

- Create your own 'user_settings.R' file by following the popup messages.



Do you have a user settings file?

  - Step 1: Choose the input folder that contains the DICOM's (here 'bidscointutorial\raw').

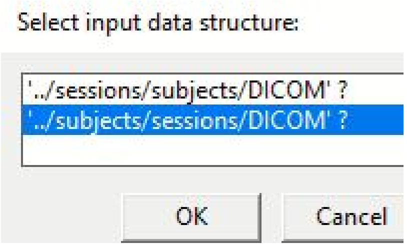


Select the input directory of DICOM folders:

Name	Date modified	Type	Size
sub-001	20/10/2022 11:39	File folder	
sub-002	20/10/2022 11:40	File folder	

Folder: raw

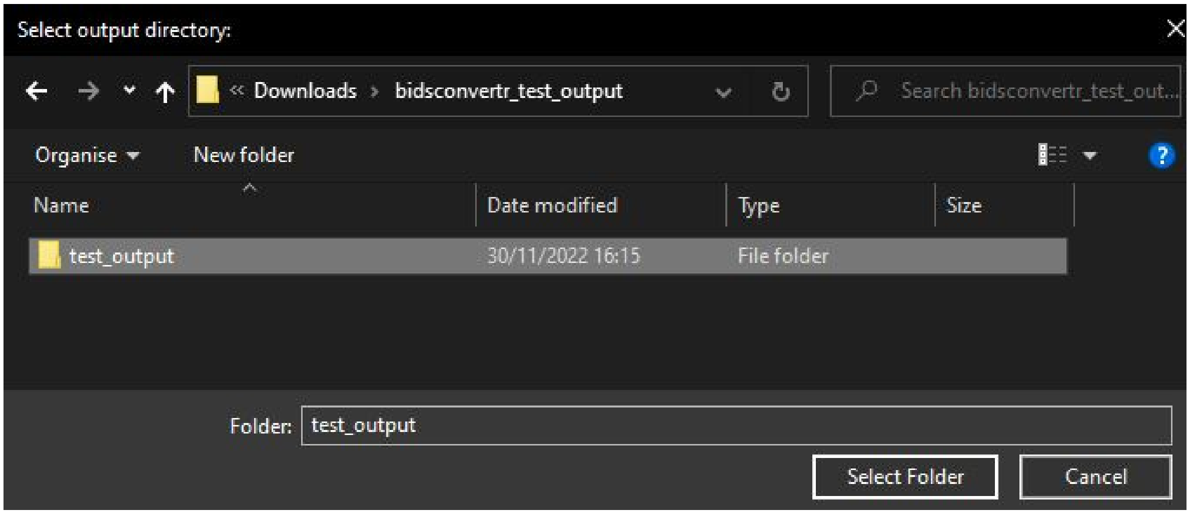
- Step 2: Select the `../subject/session/..` order of folders.



Select input data structure:

'../sessions/subjects/DICOM' ?
'../subjects/sessions/DICOM' ?

- Step 3: Create and choose the output folder. All BIDSconvertR files (including the 'user_settings.R' file) are saved here.



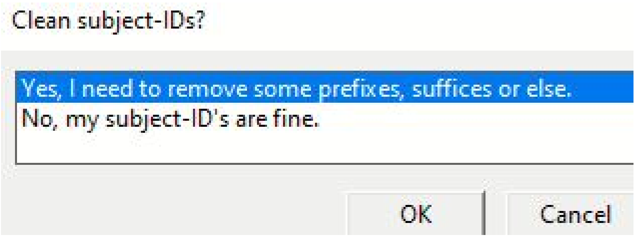
Select output directory:

« Downloads › bidsconvertr_test_output
Search bidsconvertr_test_out...
Organise
New folder

Name	Date modified	Type	Size
test_output	30/11/2022 16:15	File folder	

Folder: test_output

- Step 4: Do not use the "subject-ID" or "pattern to remove" features. You have compliant subject-IDs!

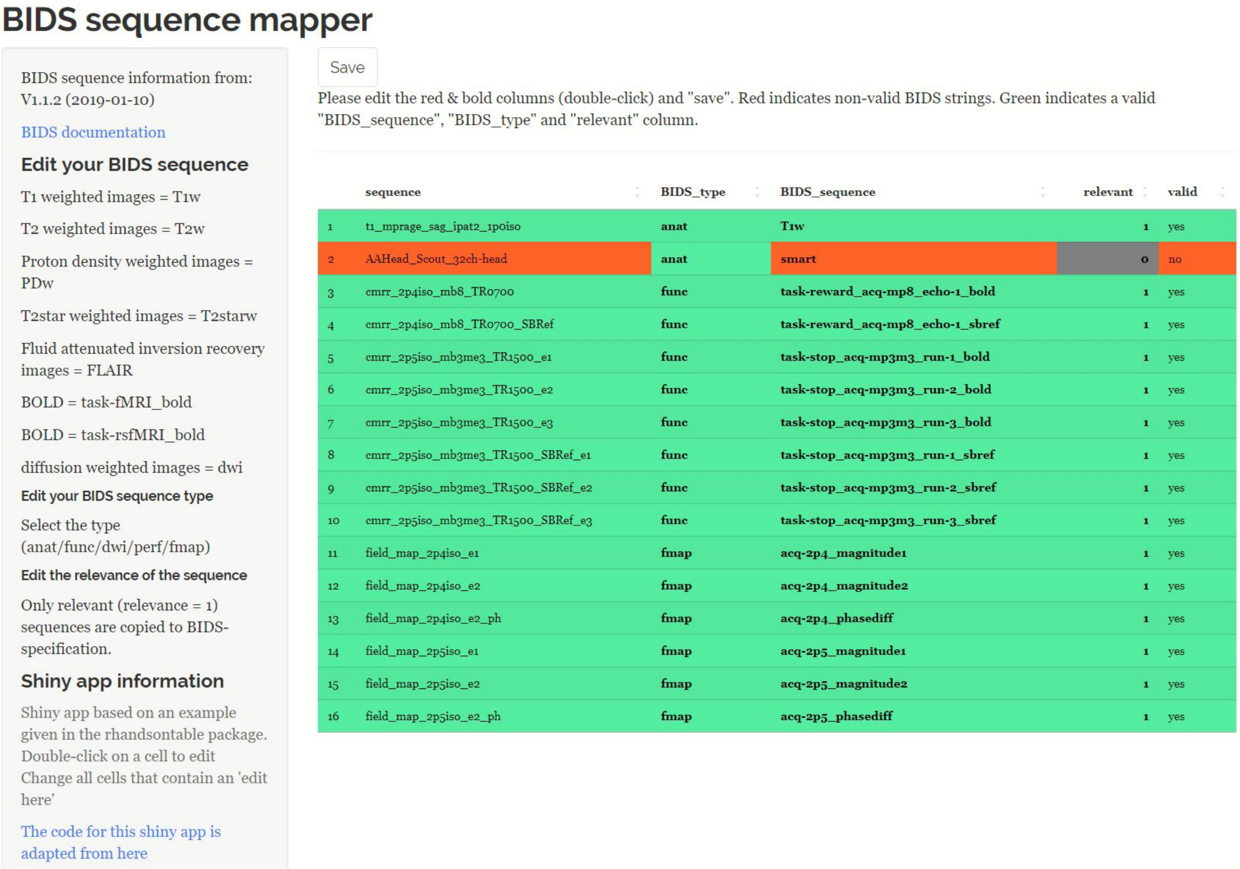


Clean subject-IDs?

Yes, I need to remove some prefixes, suffices or else.
No, my subject-ID's are fine.

4.8.2 GUI: sequence mapper

- Use the sequence mapper to rename your files in accordance with BIDS, click "save" and close the window. The workflow should automatically launch the next steps.



BIDS sequence mapper

BIDS sequence information from: V1.1.2 (2019-01-10)

BIDS documentation

Edit your BIDS sequence

T1 weighted images = T1w

T2 weighted images = T2w

Proton density weighted images = PDw

T2star weighted images = T2starw

Fluid attenuated inversion recovery images = FLAIR

BOLD = task-fMRI_bold

BOLD = task-rsfMRI_bold

diffusion weighted images = dwi

Edit your BIDS sequence type

Select the type (anat/func/dwi/perf/fmap)

Edit the relevance of the sequence

Only relevant (relevance = 1) sequences are copied to BIDS-specification.

Shiny app information

Shiny app based on an example given in the rhandsontable package. Double-click on a cell to edit Change all cells that contain an 'edit here'

The code for this shiny app is adapted from here

Please edit the red & bold columns (double-click) and "save". Red indicates non-valid BIDS strings. Green indicates a valid "BIDS_sequence", "BIDS_type" and "relevant" column.

	sequence	BIDS_type	BIDS_sequence	relevant	valid
1	t1_mprage_sag_ipat2_1p0iso	anat	T1w	1	yes
2	AAHead_Scout_32ch-head	anat	smart	0	no
3	cmrr_2p4iso_mb8_TR0700	func	task-reward_acq-mp8_echo-1_bold	1	yes
4	cmrr_2p4iso_mb8_TR0700_SBRef	func	task-reward_acq-mp8_echo-1_sbref	1	yes
5	cmrr_2p5iso_mb3me3_TR1500_e1	func	task-stop_acq-mp3m3_run-1_bold	1	yes
6	cmrr_2p5iso_mb3me3_TR1500_e2	func	task-stop_acq-mp3m3_run-2_bold	1	yes
7	cmrr_2p5iso_mb3me3_TR1500_e3	func	task-stop_acq-mp3m3_run-3_bold	1	yes
8	cmrr_2p5iso_mb3me3_TR1500_SBRef_e1	func	task-stop_acq-mp3m3_run-1_sbref	1	yes
9	cmrr_2p5iso_mb3me3_TR1500_SBRef_e2	func	task-stop_acq-mp3m3_run-2_sbref	1	yes
10	cmrr_2p5iso_mb3me3_TR1500_SBRef_e3	func	task-stop_acq-mp3m3_run-3_sbref	1	yes
11	field_map_2p4iso_e1	fmap	acq-2p4_magnitude1	1	yes
12	field_map_2p4iso_e2	fmap	acq-2p4_magnitude2	1	yes
13	field_map_2p4iso_e2_ph	fmap	acq-2p4_phasediff	1	yes
14	field_map_2p5iso_e1	fmap	acq-2p5_magnitude1	1	yes
15	field_map_2p5iso_e2	fmap	acq-2p5_magnitude2	1	yes
16	field_map_2p5iso_e2_ph	fmap	acq-2p5_phasediff	1	yes

Here you find the entries as text for copy & pasting. Alternatively, download the `sequence_map.tsv` from this link, right-click, save, and replace the file in your BIDS output folder.

sequence	BIDS_type	BIDS_sequence	relevant
t1_mprage_sag_ipat2_1p0iso	anat	T1w	1
AAHead_Scout_32ch-head	anat	smart	0
cmrr_2p4iso_mb8_TR0700	func	task-reward_acq-mp8_echo-1_bold	1
cmrr_2p4iso_mb8_TR0700_SBRef	func	task-reward_acq-mp8_echo-1_sbref	1
cmrr_2p5iso_mb3me3_TR1500_e1	func	task-stop_acq-mp3m3_run-1_bold	1
cmrr_2p5iso_mb3me3_TR1500_e2	func	task-stop_acq-mp3m3_run-2_bold	1
cmrr_2p5iso_mb3me3_TR1500_e3	func	task-stop_acq-mp3m3_run-3_bold	1
cmrr_2p5iso_mb3me3_TR1500_SBRef_e1	func	task-stop_acq-mp3m3_run-1_sbref	1
cmrr_2p5iso_mb3me3_TR1500_SBRef_e2	func	task-stop_acq-mp3m3_run-2_sbref	1
cmrr_2p5iso_mb3me3_TR1500_SBRef_e3	func	task-stop_acq-mp3m3_run-3_sbref	1
field_map_2p4iso_e1	fmap	acq-2p4_magnitude1	1
field_map_2p4iso_e2	fmap	acq-2p4_magnitude2	1
field_map_2p4iso_e2_ph	fmap	acq-2p4_phasediff	1
field_map_2p5iso_e1	fmap	acq-2p5_magnitude1	1
field_map_2p5iso_e2	fmap	acq-2p5_magnitude2	1
field_map_2p5iso_e2_ph	fmap	acq-2p5_phasediff	1

Now the data is automatically saved into BIDS, the BIDS validator is started, and the `Shiny BIDS viewer` starts.

4.9 Starting the workflow

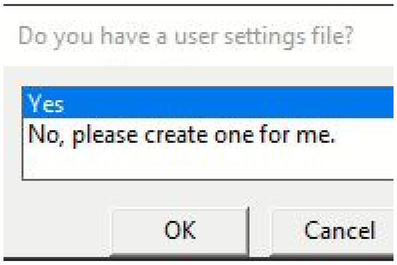
```
# Load the library.
# The 'quietly' argument turns off the messages about loading other dependencies.
library(bidsconvertr, quietly = TRUE)

# Start the workflow.
convert_to_BIDS()
```

4.9.1 The ‘user settings.R’ file - selection or creation

A ‘user settings.R’ file is required when using the ‘convert to BIDS()’ function. You are able to create a new file or select an already created one from a further conversion process.

Note: The ‘user settings.R’ file stores the settings and variables you’ve chosen (folders, filename convention, dcm2niix string, regular expressions). It is saved to the output directory and can be used again for future conversion processes. A user dialogue will walk you through the creation and selection process.

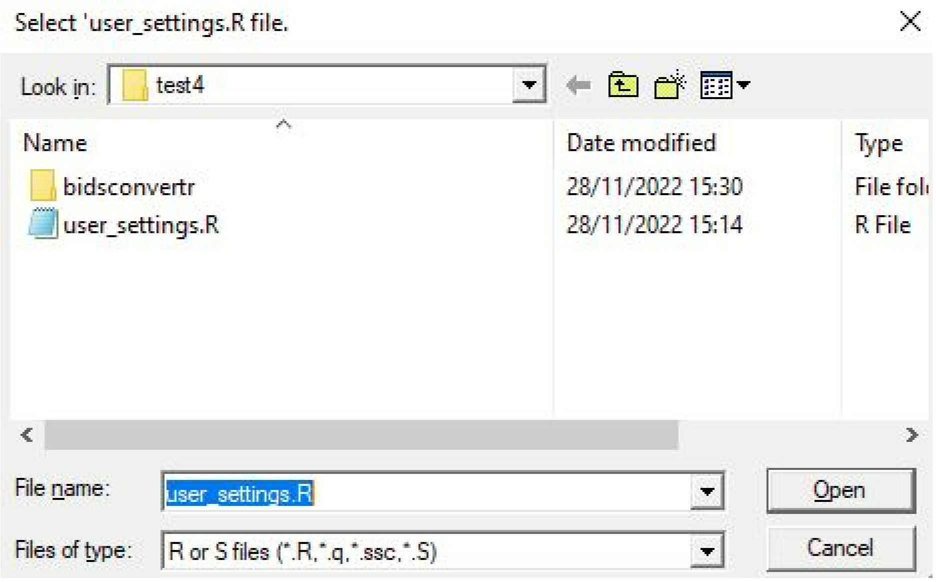
4.9.2 Do you have a `user settings` file?



Do you have a user settings file?

Option	What happens?
Yes	You are able to select your already existing file. It is located inside of a output directory. The user dialog is skipped.
No	Starts a selection process and creates your user settings file in the output folder.

If you clicked "Yes", you have to select an already existing "user_settings.R" file.



Select 'user_settings.R file.

Look in: test4

Name	Date modified	Type
bidsconvertr	28/11/2022 15:30	File fol
user_settings.R	28/11/2022 15:14	R File

File name: user_settings.R

Files of type: R or S files (*.R,*.q,*.ssc,*.S)

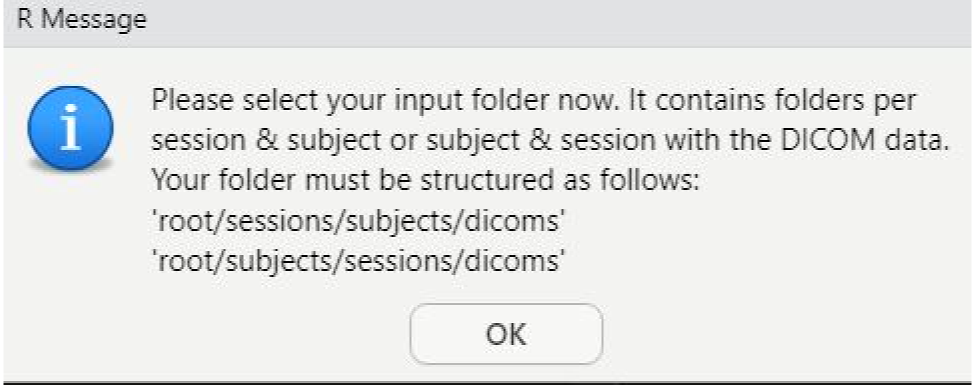
4.10 Creation of 'user settings' file.

Note: You entered `convert_to_BIDS()` and selected in the popup window that you want to create the `user_settings.R` file. You have to know where your data is and should be saved to (select the according folders) and set some options based on your input data (session/subject or subject/session) folder order. The questions from each popup window are described below in their order.

You need RStudio for popup support (user-friendly solution). Otherwise the questions will be asked inside the terminal (for the advanced user).

4.10.1 Choose input directories (DICOM)

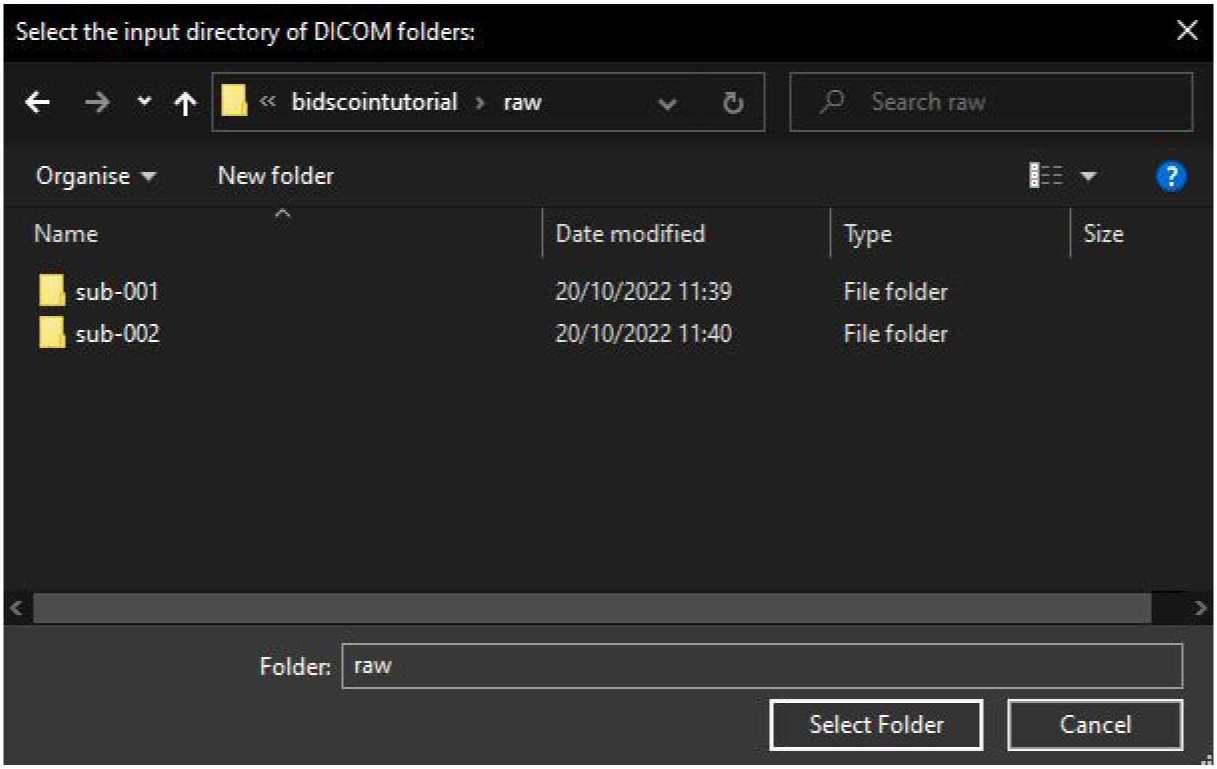
Select the root folder, which contains all session/subject or subject/session folders. If you just have one session, keep your data in a folder named, say, ‘session-0’.



R Message

Please select your input folder now. It contains folders per session & subject or subject & session with the DICOM data. Your folder must be structured as follows:
'root/sessions/subjects/dicoms'
'root/subjects/sessions/dicoms'

Please select the root directory of all DICOM images (your input folder, as described above.)



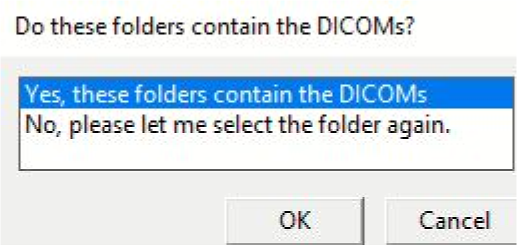
Select the input directory of DICOM folders:

Name	Date modified	Type	Size
sub-001	20/10/2022 11:39	File folder	
sub-002	20/10/2022 11:40	File folder	

Folder: raw

Several folders are listed in the terminal. These should include the DICOMs.

Do these folders contain the DICOM images?

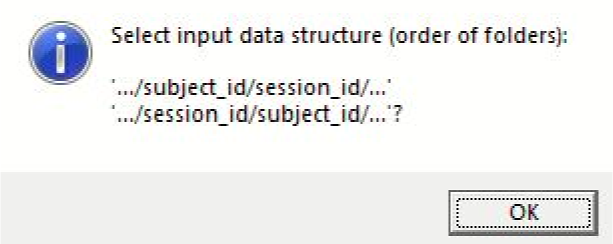


Do these folders contain the DICOMs?

Yes, these folders contain the DICOMs
No, please let me select the folder again.

Option to select	What happens?
Yes	next step
No	You are able to select the folder again

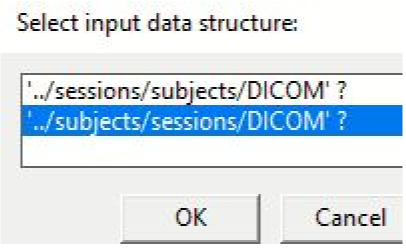
4.10.2 Select folder hierarchy (subject/session or session/subject)



Select input data structure (order of folders):

'.../subject_id/session_id/...'
'.../session_id/subject_id/...'?

Is your DICOM data structured as 'session/subject' or 'subject/session'.



Select input data structure:

'../sessions/subjects/DICOM' ?
'../subjects/sessions/DICOM' ?

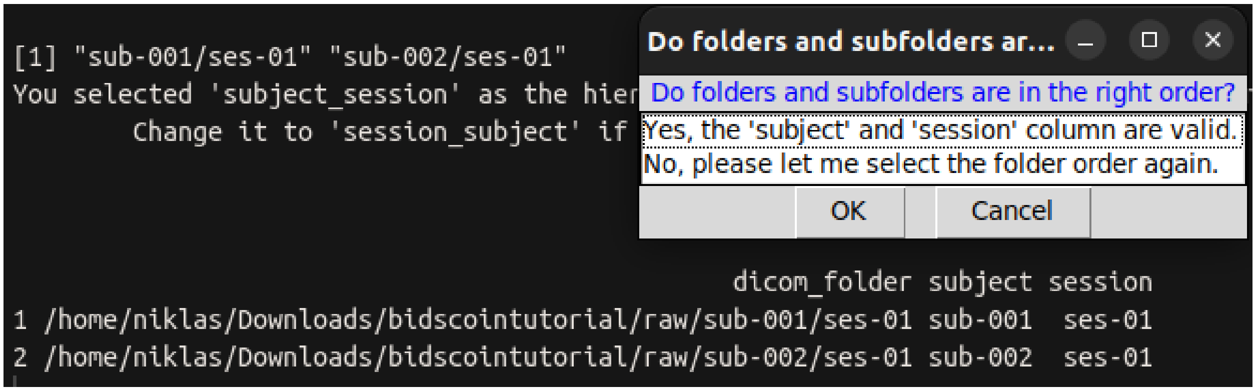
Based on this selection, the tool extracts the subject- and session-ID's. The folders are displayed on the console.

Please note: Any subject- or session-ID is possible! Without "sub-" or "ses-" as well.

Folder order of your files	Selection
sub-0001/ses-01	subject/session
ses-01/sub-0001	session/subject

Choose the option that best fits your data, as displayed in the terminal.

Were subject-ID’s and session-ID’s extracted correctly?



```
[1] "sub-001/ses-01" "sub-002/ses-01"
You selected 'subject_session' as the hier
        Change it to 'session_subject' if

                                               dicom_folder subject session
1 /home/niklas/Downloads/bidscointutorial/raw/sub-001/ses-01 sub-001  ses-01
2 /home/niklas/Downloads/bidscointutorial/raw/sub-002/ses-01 sub-002  ses-01
```

Do folders and subfolders ar...

Do folders and subfolders are in the right order?

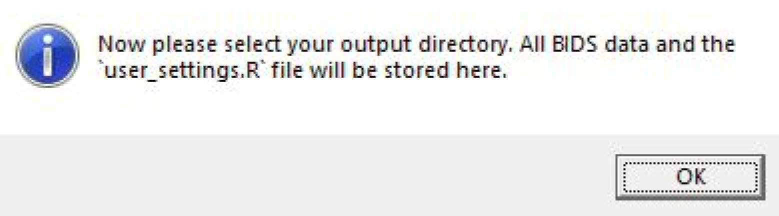
Yes, the 'subject' and 'session' column are valid.

No, please let me select the folder order again.

A table with the columns “subject” and “session” is displayed on the terminal. Do they appear to be valid?

Option to select	What happens?
Yes	Next step is started.
No	Change the folder order again.

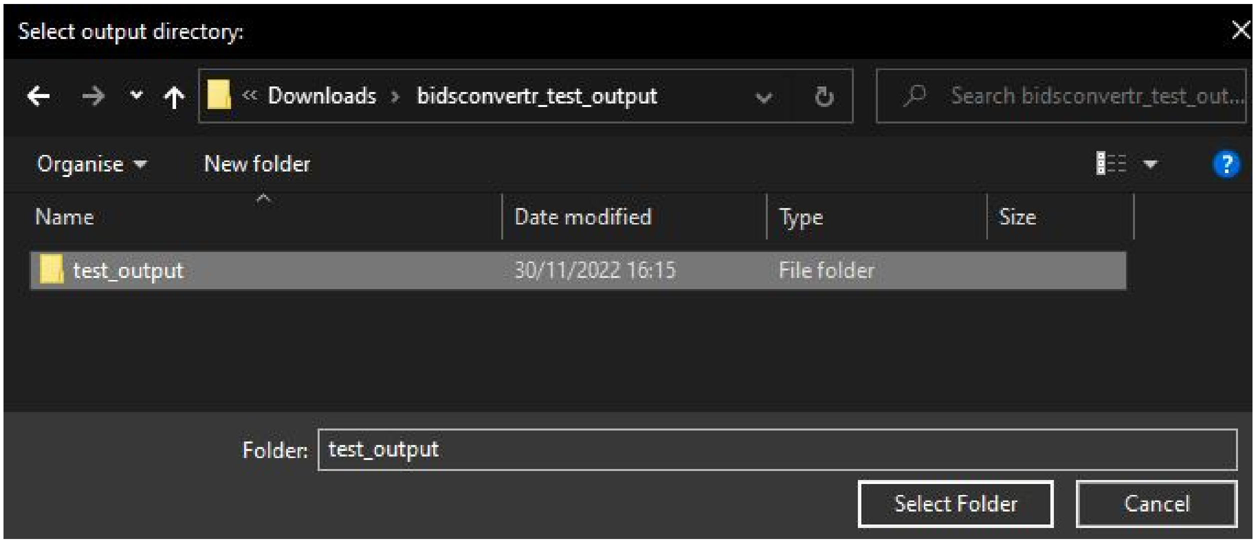
4.10.3 Choose output directory (BIDS)



Now please select your output directory. All BIDS data and the `user_settings.R` file will be stored here.

The output directory will contain:

- temporary files and metadata of the dcm2niix conversion,
- ‘user_settings.R’ file and
- BIDS output.



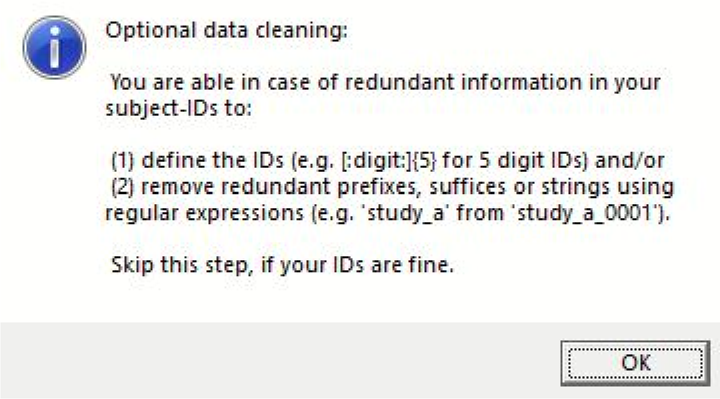
Select output directory:

Folder: test_output

You can choose any folder on your hard drive. As a result, raw data can be stored somewhere other than where processed data is.

4.11 Advanced features

Note: This part is optional. You should read this if your subject-IDs are messy (have a redundant prefix, suffix, or string, for example), or if you want to rename your session-IDs.



Optional data cleaning:

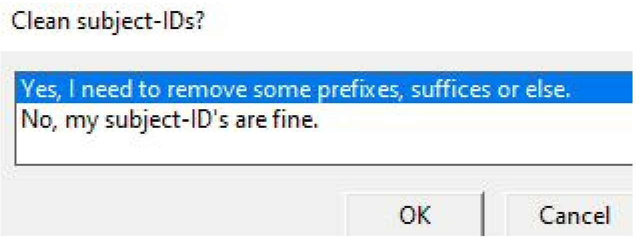
You are able in case of redundant information in your subject-IDs to:

(1) define the IDs (e.g. [:digit:]{5} for 5 digit IDs) and/or
(2) remove redundant prefixes, suffices or strings using regular expressions (e.g. 'study_a' from 'study_a_0001').

Skip this step, if your IDs are fine.

This section covers the optional cleaning or extraction of subject-ID's and renaming of sessions.

You are prompted to decide if you wish to edit subject- and session-ID's during the user dialog. You can skip this step if your data was already collected with clear subject- and session-ID's:



Clean subject-IDs?

Yes, I need to remove some prefixes, suffices or else.
No, my subject-ID's are fine.

Option to select	What happens?
No, my subject-ID's are fine.	The paths are created, subject- and session-ID's get a "sub-" and "ses-" prefix, if it isn't already there.
Yes	Set a subject-ID regular expression or set a string, prefix, suffix or regular expression, which is then removed from each subject-ID.

4.11.1 Resources for regular expressions

For more information on regular expressions (regex) please see the stringR cheat sheet or RegexOne.

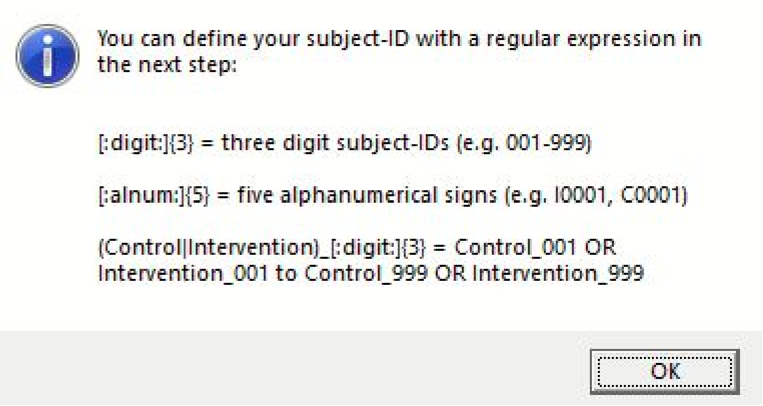
Each regex set here should match to your data. Contact me via email or the issues in this repository if you run across any troubles.

You should have enough freedom using these two routines to clean up your filenames and gradually change and improve your regular expression.

4.11.2 subject-ID cleaning

The input folder name serves as the subject-ID. In the absence of a regular expression, the subject-ID is unaltered.

Regular expression: subject-ID



You can define your subject-ID with a regular expression in the next step:

[:digit:]{3} = three digit subject-IDs (e.g. 001-999)

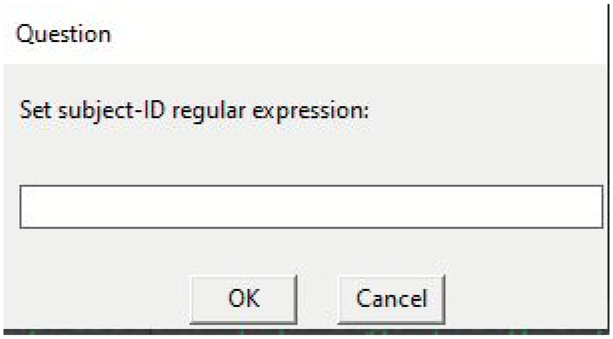
[:alnum:]{5} = five alphanumerical signs (e.g. I0001, C0001)

(Control|Intervention)_[:digit:]{3} = Control_001 OR Intervention_001 to Control_999 OR Intervention_999

The subject-ID is extracted from the input string using this regular expression. If all of your files had a well established naming convention, you could use this.

subject-ID	regular expression	described in words	output subject-ID
01234	[:digit:]{5}	5 digits	sub-01234
Control2132	(Control|Patient)[:digit:]{4}	"Control" OR "Patient" followed by 4 digits	sub-Control2132
Patient0123_test	(Control|Patient)[:digit:]{4}		sub-Patient0213
abcd0123	[:alpha:]{4}[:digit:]{4}	4 letters and 4 digits	sub-abcd0123
pilot_sdfjd3222	[:alpha:]{4}[:digit:]{4}		sub-sdfjd3222
adc932d	[:alnum:]{5,7}	between 5 to 7 alphanumeric (letters, digits)	sub-adc932d
23d49	[:alnum:]{5,7}		sub-23d49

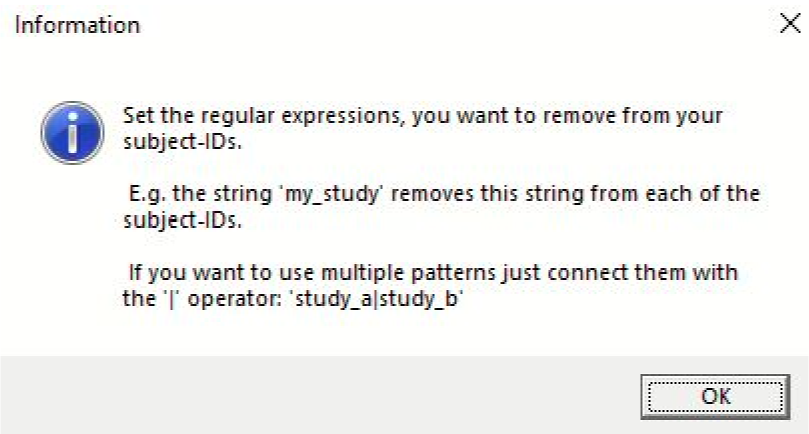
Examples of subject-ID regular expressions



Question

Set subject-ID regular expression:

Regular expression: pattern to remove



Information

Set the regular expressions, you want to remove from your subject-IDs.

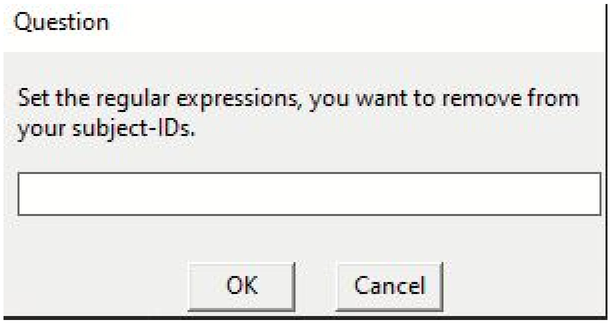
E.g. the string 'my_study' removes this string from each of the subject-IDs.

If you want to use multiple patterns just connect them with the '|' operator: 'study_a|study_b'

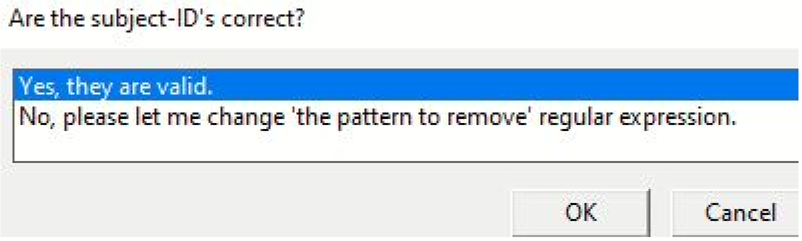
The `pattern_to_remove` regular expression simply removes the regular expression(s) from the subject-ID.

subject-ID	regular expression	described in words	output subject-ID
02313_bidirect	_(bidirect|BiDirect|Bidiect)	"_" followed by "bidirect", "BiDirect" or "BiDiect"	sub-02313
03211_BiDirect	_(bidirect|BiDirect|Bidiect)		sub-03211
02111_Bidiect	_(bidirect|BiDirect|Bidiect)		sub-02111
test0111	test|study_a_	"test" or "study_a_"	sub-0111
study_a_1111	test|study_a_		sub-1111
pre9222post	pre|post|suffix|prefix	as in the cell above	sub-9222
suffix223prefix	pre|post|suffix|prefix		sub-223

Examples of 'patterns to remove' regular expressions

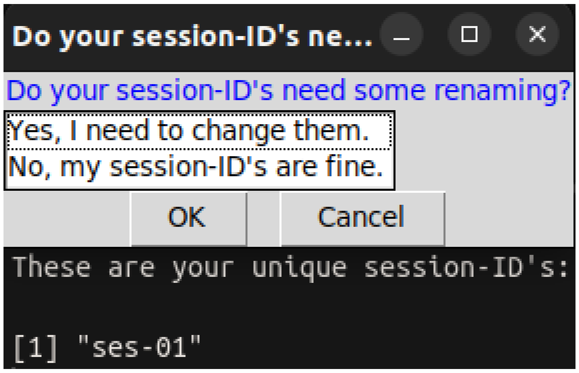

Question

Set the regular expressions, you want to remove from your subject-IDs.


Are the subject-ID's correct?

Yes, they are valid.
No, please let me change 'the pattern to remove' regular expression.

4.11.3 session-ID cleaning


Do your session-ID's need some renaming?

Yes, I need to change them.
No, my session-ID's are fine.

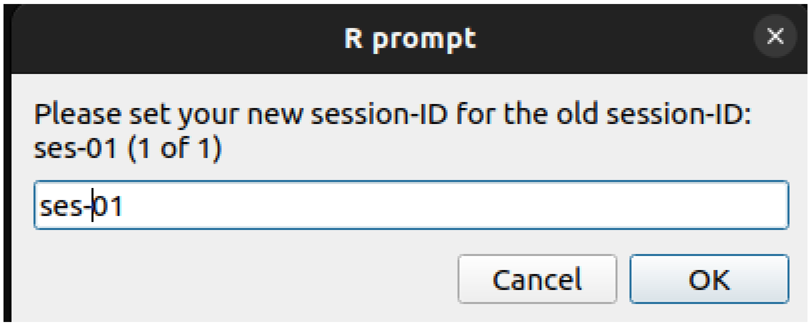
```
These are your unique session-ID's:

[1] "ses-01"
```

You can choose to keep them or rename them.

Option to select	What happens?
Yes, I need to change them	Each session is opened separately and you can enter the new session-ID.
No, my session-ID’s are fine.	Nothing is edited, you keep your session-IDs

Yes:

“Yes, I need to change them” can result in output like this. Your session-IDs can now be edited. The user could also choose to use “followup” or something else.



Please set your new session-ID for the old session-ID: ses-01 (1 of 1)

ses-01

session-ID (old)	session-ID (user input)	session-BIDS
baseline	1	ses-1
follow_up	2	ses-2

No:

"No, my session-ID's are fine"

session-ID (old)	session-BIDS
baseline	ses-baseline
follow_up	ses-follow_up

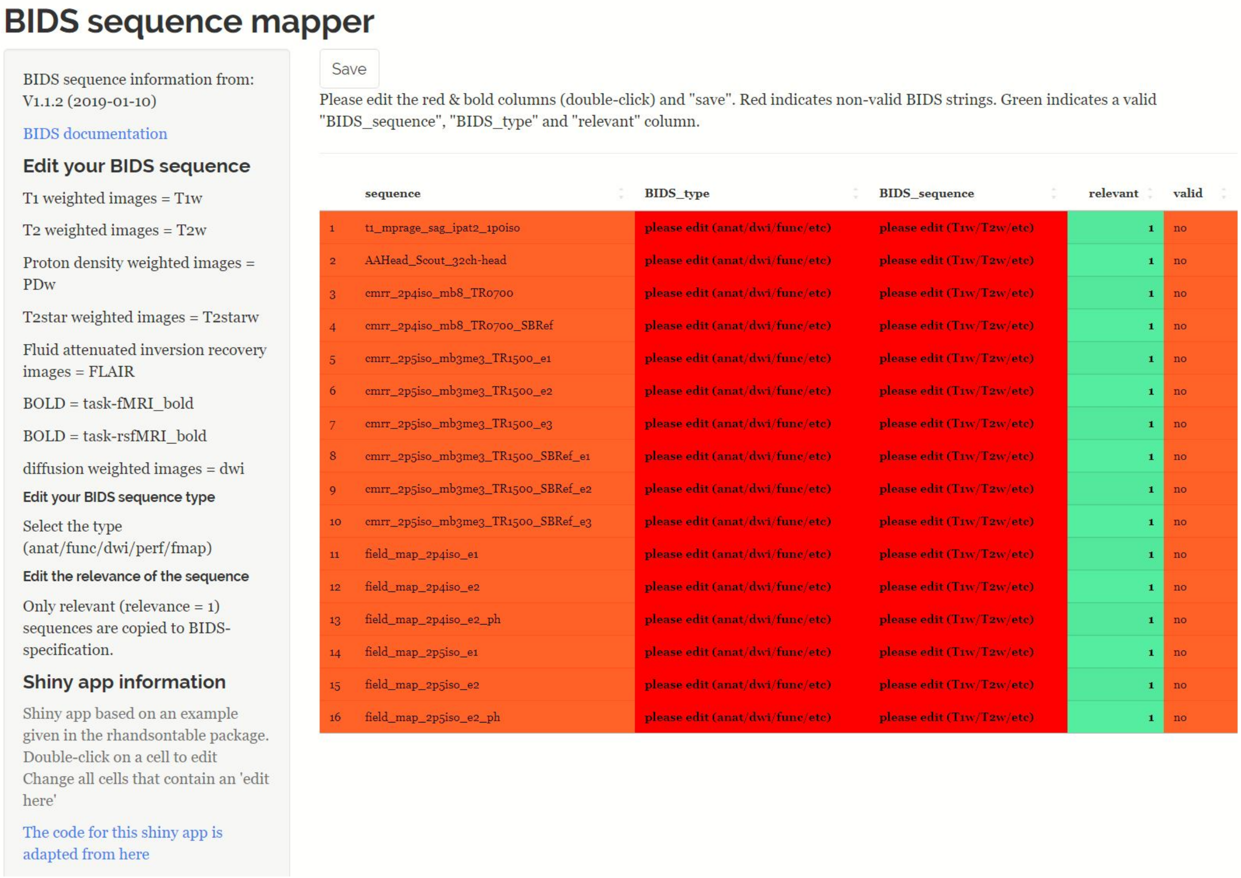
4.12 dcm2niix conversion

Your data is converted into NifTI format using dcm2niix based on the inputs. All potentially identifiable metadata was removed with dcm2niix from the NII + JSON files.

Each JSON header is read out and combined into one file for further inspection.

4.13 GUI: The sequence mapper

After extracting the distinct sequence-IDs, the `Sequence Mapper`, a Shiny application is started. You should be able to rename your sequences in accordance with the BIDS specification. A double-click on a cell will open it for editing. The `Sequence Mapper` should now begin displaying the interface as follows:



BIDS sequence mapper

BIDS sequence information from: V1.1.2 (2019-01-10)

BIDS documentation

Edit your BIDS sequence

T1 weighted images = T1w

T2 weighted images = T2w

Proton density weighted images = PDw

T2star weighted images = T2starw

Fluid attenuated inversion recovery images = FLAIR

BOLD = task-fMRI_bold

BOLD = task-rsfMRI_bold

diffusion weighted images = dwi

Edit your BIDS sequence type

Select the type (anat/func/dwi/perf/fmap)

Edit the relevance of the sequence

Only relevant (relevance = 1) sequences are copied to BIDS-specification.

Shiny app information

Shiny app based on an example given in the rhandsontable package. Double-click on a cell to edit Change all cells that contain an 'edit here'

The code for this shiny app is adapted from here

Please edit the red & bold columns (double-click) and "save". Red indicates non-valid BIDS strings. Green indicates a valid "BIDS_sequence", "BIDS_type" and "relevant" column.

	sequence	BIDS_type	BIDS_sequence	relevant	valid
1	t1_mprage_sag_ipat2_1p0iso	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
2	AAHead_Scout_32ch-head	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
3	cmrr_2p4iso_mb8_TR0700	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
4	cmrr_2p4iso_mb8_TR0700_SBRef	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
5	cmrr_2p5iso_mb3me3_TR1500_e1	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
6	cmrr_2p5iso_mb3me3_TR1500_e2	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
7	cmrr_2p5iso_mb3me3_TR1500_e3	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
8	cmrr_2p5iso_mb3me3_TR1500_SBRef_e1	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
9	cmrr_2p5iso_mb3me3_TR1500_SBRef_e2	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
10	cmrr_2p5iso_mb3me3_TR1500_SBRef_e3	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
11	field_map_2p4iso_e1	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
12	field_map_2p4iso_e2	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
13	field_map_2p4iso_e2_ph	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
14	field_map_2p5iso_e1	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
15	field_map_2p5iso_e2	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no
16	field_map_2p5iso_e2_ph	please edit (anat/dwi/func/etc)	please edit (T1w/T2w/etc)	1	no

You have to edit each entry according to the BIDS specification. Some tips can be found on the left panel and hyperlinks to the BIDS specification. Then you click "save" and close the 'sequence mapper'.

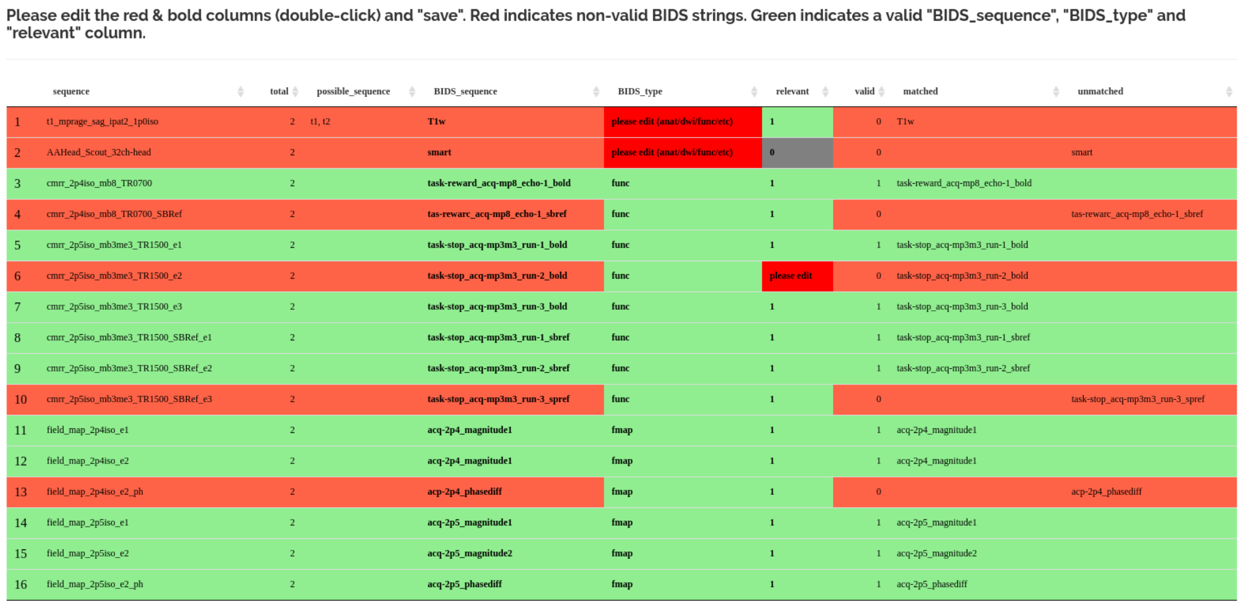
1) You must edit every cell with the phrase "please edit."

2) Using regular expressions based on BIDS, each 'BIDS_sequence' and 'BIDS_type' entry is validated in the backend. There is a good possibility that a row will provide a valid BIDS output if it is coloured "green."

3) Note that you are not restricted in how you name files by us. If you mark a non-valid BIDS string as relevant, it will be copied to BIDS and you can save it.

4) Red rows with "irrelevant" flagged cells can be disregarded. There is no export of these to BIDS. However, you need to take the "please edit" out of them. This is required in order for the algorithm to recognize that the user has changed each cell.

5) After clicking "save", please exit the app. The workflow continues after the closing.



Please edit the red & bold columns (double-click) and "save". Red indicates non-valid BIDS strings. Green indicates a valid "BIDS_sequence", "BIDS_type" and "relevant" column.

	sequence	total	possible_sequence	BIDS_sequence	BIDS_type	relevant	valid	matched	unmatched
1	t1_mprage_sag_ipat2_1p0iso	2	t1, t2	**T1w**	**please edit (anat/dwi/func/etc)**	**1**	0	T1w	
2	AAHead_Scout_32ch-head	2		**smart**	**please edit (anat/dwi/func/etc)**	**0**	0		smart
3	cmrr_2p4iso_mb8_TR0700	2		**task-reward_acq-mp8_echo-1_bold**	**func**	**1**	1	task-reward_acq-mp8_echo-1_bold	
4	cmrr_2p4iso_mb8_TR0700_SBRef	2		**tas-rewarc_acq-mp8_echo-1_sbref**	**func**	**1**	0		tas-rewarc_acq-mp8_echo-1_sbref
5	cmrr_2p5iso_mb3me3_TR1500_e1	2		**task-stop_acq-mp3m3_run-1_bold**	**func**	**1**	1	task-stop_acq-mp3m3_run-1_bold	
6	cmrr_2p5iso_mb3me3_TR1500_e2	2		**task-stop_acq-mp3m3_run-2_bold**	**func**	**please edit**	0	task-stop_acq-mp3m3_run-2_bold	
7	cmrr_2p5iso_mb3me3_TR1500_e3	2		**task-stop_acq-mp3m3_run-3_bold**	**func**	**1**	1	task-stop_acq-mp3m3_run-3_bold	
8	cmrr_2p5iso_mb3me3_TR1500_SBRef_e1	2		**task-stop_acq-mp3m3_run-1_sbref**	**func**	**1**	1	task-stop_acq-mp3m3_run-1_sbref	
9	cmrr_2p5iso_mb3me3_TR1500_SBRef_e2	2		**task-stop_acq-mp3m3_run-2_sbref**	**func**	**1**	1	task-stop_acq-mp3m3_run-2_sbref	
10	cmrr_2p5iso_mb3me3_TR1500_SBRef_e3	2		**task-stop_acq-mp3m3_run-3_spref**	**func**	**1**	0		task-stop_acq-mp3m3_run-3_spref
11	field_map_2p4iso_e1	2		**acq-2p4_magnitude1**	**fmap**	**1**	1	acq-2p4_magnitude1	
12	field_map_2p4iso_e2	2		**acq-2p4_magnitude1**	**fmap**	**1**	1	acq-2p4_magnitude1	
13	field_map_2p4iso_e2_ph	2		**acp-2p4_phasediff**	**fmap**	**1**	0		acp-2p4_phasediff
14	field_map_2p5iso_e1	2		**acq-2p5_magnitude1**	**fmap**	**1**	1	acq-2p5_magnitude1	
15	field_map_2p5iso_e2	2		**acq-2p5_magnitude2**	**fmap**	**1**	1	acq-2p5_magnitude2	
16	field_map_2p5iso_e2_ph	2		**acq-2p5_phasediff**	**fmap**	**1**	1	acq-2p5_phasediff	

When a sequence is recognised as BIDS compatible, it is indicated in the "matched" column. Investigate each letter of the filename if your sequence appears in the "unmatched" column.

4.14 BIDS validation

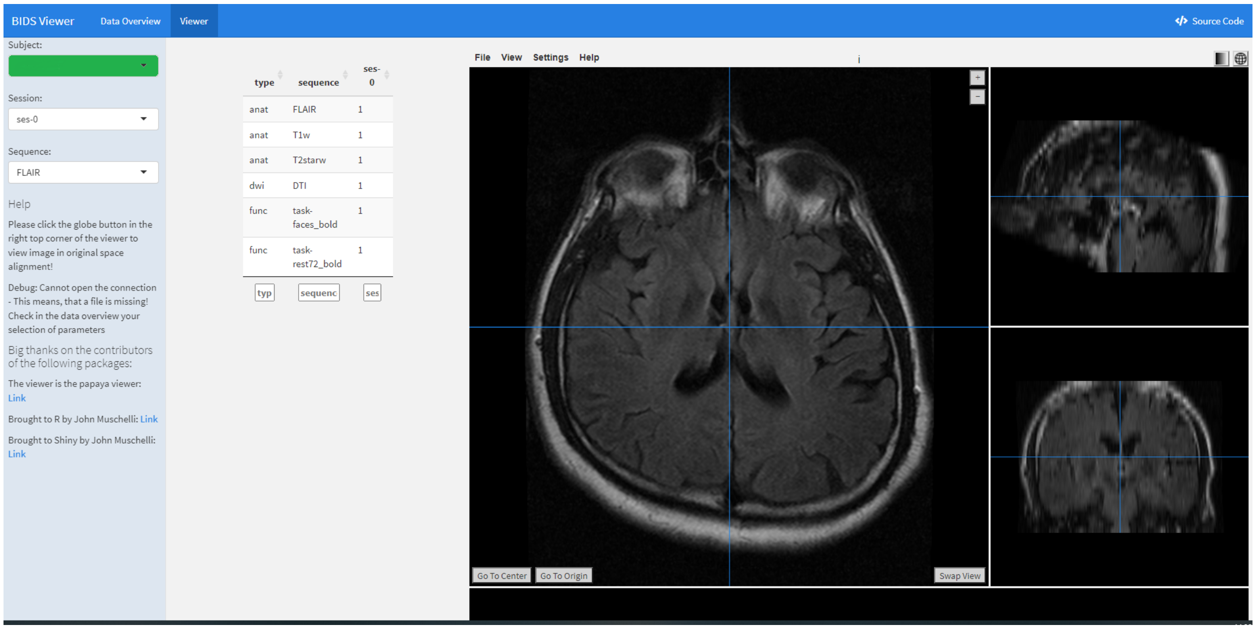
Note: Only the metadata contained within the BIDS folder is free of potentially identifiable information. Follow the legal terms when sharing your dataset and consider additional defacing, pseudonymization, or both.

When everything is in order:

1. The files are copied to BIDS.
2. The BIDS validation process begins. If Docker is already installed on your machine, it is started automatically. If not, the online-version is launched, and you must choose your folder manually. Please note: Files are never uploaded to the BIDS-Validator.
3. You are prompted, if you want to remove temporary images from your hard disk. Avoid performing this by hand! Only do this after validating your data and gathering all of your data.

4.15 GUI: BIDS viewer

A Shiny viewer is launched to examine the images visually.



Subject:

Session: ses-0

Sequence: FLAIR

Help

Please click the globe button in the right top corner of the viewer to view image in original space alignment!

Debug: Cannot open the connection - This means, that a file is missing! Check in the data overview your selection of parameters

Big thanks on the contributors of the following packages:

The viewer is the papaya viewer: Link

Brought to R by John Muschelli: Link

Brought to Shiny by John Muschelli: Link

type	sequence	ses-0
anat	FLAIR	1
anat	T1w	1
anat	T2starw	1
dwi	DTI	1
func	task-faces_bold	1
func	task-rest72_bold	1

4.16 What is in my 'user_settings.R' file?

You can also edit the file "user_settings.R" manually to your needs.

Variable	Example	Description
path_input_dicom	"/media/niklas/BIDS_data/dicom/"	Input path, where your DICOM folders are inside of session folders
path_output	"/media/niklas/BIDS_data/BIDSconvertR/"	A path, where all the output of the converter should be written to.
study_name	"BiDirect Study"	Your study name, only needed for the dashboard rendering.
regex_subject_id	"[:digit:]{5}"	Regex defining your unique subject ID's.
regex_group_id	"[:digit:]{1}(?=[:digit:]{4})"	Regex defining the group ID (if present).
regex_remove_pattern	"[:punct:]{1}|[:blank:]{1}|((b|d)i(d|b)	These regex will be removed from the file names.
sessions_id_old	c("Baseline", "FollowUp", "FollowUp2", "FollowUp3")	The folder (and session) names before conversion
sessions_id_new	c("0", "2", "4", "6")	The folder (and session) names after conversion. These can be identical to "sessions_id_old". But note, that in BIDS a number is the way to go.
mri_sequences	"T1|T2|DTI|fmr|rest|rs|func|FLAIR|sm	These are regular expressions, which should be matched to your MRI sequence ID's.
dcm2niix_argument_string	"-ba y -f %d -z y -w 0 -i y"	string of the arguments, which are used for dcm2niix, can be changed.

Table 1: Information in the user settings file

4.17 dcm2niix customisation

4.17.1 installation of other versions

The “convert_to_BIDS()” function automatically uses the tested “v1.0.20211006” of dcm2niix. Other versions can be installed by changing the version number and running the script before running “convert_to_BIDS()” the first time.

Otherwise, go to your output folder, delete the dcm2niix files in it, and run the “install_dcm2niix()” version with your version number.

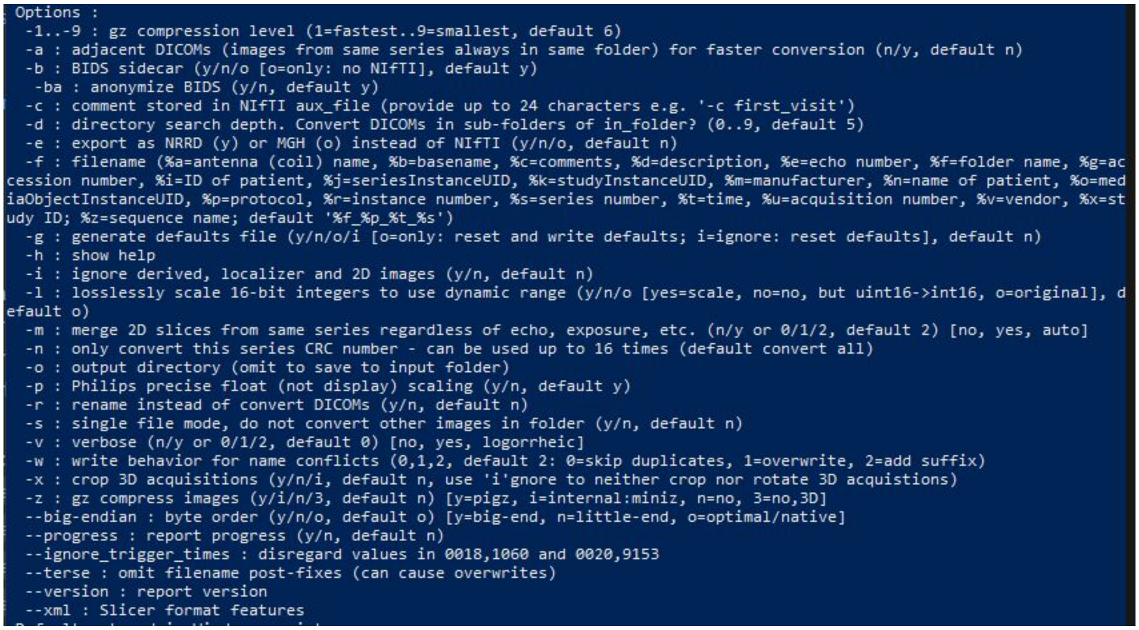
https://github.com/rordenlab/dcm2niix/releases

```
install_dcm2niix("v1.0.20181125") # if you want to install the specific version v1.0.
↪20181125
```

4.17.2 using other arguments

You can edit the dcm2niix_argument_string in the “user_settings.R” file according to your needs.

Just read here or inspect the possible arguments from this image:



```
 Options :
  -1..-9 : gz compression level (1=fastest..9=smallest, default 6)
  -a : adjacent DICOMs (images from same series always in same folder) for faster conversion (n/y, default n)
  -b : BIDS sidecar (y/n/o [o=only: no NIfTI], default y)
   -ba : anonymize BIDS (y/n, default y)
  -c : comment stored in NIfTI aux_file (provide up to 24 characters e.g. '-c first_visit')
  -d : directory search depth. Convert DICOMs in sub-folders of in_folder? (0..9, default 5)
  -e : export as NRRD (y) or MGH (o) instead of NIfTI (y/n/o, default n)
  -f : filename (%a=antenna (coil) name, %b=basename, %c=comments, %d=description, %e=echo number, %f=folder name, %g=ac
cession number, %i=ID of patient, %j=seriesInstanceUID, %k=studyInstanceUID, %m=manufacturer, %n=name of patient, %o=med
iaObjectInstanceUID, %p=protocol, %r=instance number, %s=series number, %t=time, %u=acquisition number, %v=vendor, %x=st
udy ID; %z=sequence name; default '%f_%p_%t_%s')
  -g : generate defaults file (y/n/o/i [o=only: reset and write defaults; i=ignore: reset defaults], default n)
  -h : show help
  -i : ignore derived, localizer and 2D images (y/n, default n)
  -l : losslessly scale 16-bit integers to use dynamic range (y/n/o [yes=scale, no=no, but uint16->int16, o=original], d
efault o)
  -m : merge 2D slices from same series regardless of echo, exposure, etc. (n/y or 0/1/2, default 2) [no, yes, auto]
  -n : only convert this series CRC number - can be used up to 16 times (default convert all)
  -o : output directory (omit to save to input folder)
  -p : Philips precise float (not display) scaling (y/n, default y)
  -r : rename instead of convert DICOMs (y/n, default n)
  -s : single file mode, do not convert other images in folder (y/n, default n)
  -v : verbose (n/y or 0/1/2, default 0) [no, yes, logorrheic]
  -w : write behavior for name conflicts (0,1,2, default 2: 0=skip duplicates, 1=overwrite, 2=add suffix)
  -x : crop 3D acquisitions (y/n/i, default n, use 'i'gnore to neither crop nor rotate 3D acquistions)
  -z : gz compress images (y/i/n/3, default n) [y=pigz, i=internal:miniz, n=no, 3=no,3D]
  --big-endian : byte order (y/n/o, default o) [y=big-end, n=little-end, o=optimal/native]
  --progress : report progress (y/n, default n)
  --ignore_trigger_times : disregard values in 0018,1060 and 0020,9153
  --terse : omit filename post-fixes (can cause overwrites)
  --version : report version
  --xml : Slicer format features
```

dcm2niix_argument_string <- `-ba y -f %d -z y -w 0 -i y`

Argument	Setting	Behaviour
-ba	y (yes)	bids anonymisation of JSON sidecar
-f	%d	string for the filename (do not change this one)
-z	y (yes)	compress the output (nii.gz instead of nii)
-w	0	in case of duplicate filename -> skip
-i	y (yes)	ignore derived, localizer and 2d images

Used arguments for conversion

Please edit these, if the conversion went wrong.

4.18 Table of functions

function	description	output
convert_to_BIDS()	Wrapper function for the whole workflow described below. Performs the setup of the 'user_settings.R' file, DICOM conversion, reads the json sidecar files, starts the 'sequence mapper', copies the file into BIDS standard, creates a diagnostic dashboard and runs the 'Shiny BIDS' viewer.	All the outputs are described below.
select_user_settings_file()	Select a 'user_settings.r' file or create one with a point-and-click workflow.	The 'user_settings.R' file and the path to it.
prepare_environment()	Uses the input from the 'user_settings.R' file to create all environment variables and runs some checks on the data.	Creates environment variables and dataframes.
install_dcm2niix()	Downloads and unpacks 'dcm2niix' cite{Li2016} to the output folder.	Downloads dcm2niix to the output folder.
dcm2nii_converter_anon()	Converts the DICOM images to NIfTI and removes all sensitive information from the header and the json sidecar files.	NIfTI (anonymized header), json sidecar (anonymized)
dcm2nii_converter_json()	Extracts only the json sidecar files with the sensitive information from the DICOM images.	json sidecar
read_json_headers()	Reads all json sidecars, builds a dataframe containing all this information and saves it.	sequence_overview.tsv, json_metadata.tsv
sequence_mapper()	Starts the 'sequence mapper' shiny app to edit all unique sequence filenames to BIDS.	sequence_map.tsv
check_sequence_map()	Checks if all entries of the 'sequence mapper' were edited.	
copy2BIDS()	Copies and renames the files from the temporary folder to a BIDS sourcedata folder. Creates other required BIDS files.	copy2BIDS.tsv, CHANGES, README, dataset_description.json, participants.json, participants.tsv
start_bids_validator()	Starts the BIDS-Validator in Docker (in Docker is installed) on the BIDS folder, otherwise the BIDS-Validator website is launched.	Diagnostic output in the terminal about the BIDS validity of the dataset.
run_shiny_BIDS()	Starts the 'Shiny BIDS' MR viewer app. This function can also be used on other datasets, when giving a BIDS path as argument.	
delete_temp_files()	Asks the user, if the temporary files should be deleted. Only recommended, when all data is converted and BIDS validity is ensured.	

4.19 Table of diagnostic files

filename	description	variables
dicom_paths.tsv	Output of the ‘prepare_environment()’ function. Shows the extracted dicom folders, input and output paths, and the applied regular expressions.	Take care, that your “subject/group/session_id” matches the regular expressions. Inspect the ‘output_path’ variables to find out, where the data will be saved after converting by dcm2niix cite{Li2016}. The ‘rest_string’ variables contain the removed information from the subject names.
json_metadata.tsv	Output of the ‘read_JSON_headers()’ function. The ‘create_dashboard()’ uses the information from this file.	Subject, group, session and sequence variables for identification and all physical MR parameters from the JSON sidecar.
sequence_overview.tsv	The ‘sequence_mapper()’ creates this file. It gives an overview of the number of images per unique sequence per session.	The ‘sequence’ column is the identifier column, ‘possible sequence’ is based on the regular expression from the ‘user_settings.R’ file. Per session one column is created containing the number of files. The ‘subject_session_merge’ column contains each subject with this sequence, up to a number of 30 observations.
sequence_map.tsv	The ‘sequence_mapper()’ creates and updates this file with each unique sequence ID from the ‘json_metadata.tsv’. Based on this file the ‘copy2BIDS’ output paths are created.	The variables show the total number of sequences with the unique ID; the sequence type is identified by the regular expression from the ‘user_settings.R’ file. ‘BIDS_sequence’ and ‘BIDS_type’ need to be manually matched with the sequence_mapper() to the BIDS specification. The ‘relevant’ column is used for sequence selection. It is coded with ‘0’ for “no export to BIDS” and ‘1’ for “copy this file into BIDS”.
copy2bids.tsv	Output of the copy2BIDS() function. Shows the input and output file paths of each file.	Variables per subject, session and sequence ID. Main information is in the ‘input_file_paths’ and ‘output_file_path’ columns.