

Using R with the Flywheel SDK

Shelby Bachman

Terminology

2020_Shelby_HRVB_LCintensity Emotion Cognition (fw//emocog2020_Shelby_HRVB_LCintensity) ? S

Description Sessions Information Analyses Data Views : Download Project

Actions Advanced Filters [List Icon] [User Icon]

Acquisitions Session Subject Project Analyses Provenance

☐ Subject Label Session Label Timestamp

☐ sub-5001 subject

☐ ses-pre session

☐ ses-post

2 sub-5002

2 sub-5003

2 sub-5004

2 sub-5005

2 sub-5006

2 sub-5007

1 sub-5008

ses-pre No timestamp BIDS View [Toggle] Run Gear [Menu Icon]

☐ Description Classification All Types

☐ T1w acquisition

sub-5001_ses-pre_T1w.nii.gz file MR: Structural, T1 nifti

.manifest.json N/A source code

sub-5001_ses-pre_T1w_resamp.nii.gz N/A nifti

☐ acq-FSE_T1w

sub-5001_ses-pre_acq-FSE_T1w.nii.gz MR: Structural, T1 nifti

.manifest.json N/A source code

sub-5001_ses-pre_acq-FSE_T1w_resamp.nii.gz N/A nifti

Default filenames on flywheel

<div><input type="checkbox"/> t1_mprage_short_32channel 2019-12-04 16:04</div>				<div></div>	
<div><input type="checkbox"/></div>	1.3.12.2.1107.5.2.43.67025.2019120416000098459528061.0.0.0.dicom.zip	MR: Structural, T1, T2*, MPAGE	dicom		
<div><input checked="" type="checkbox"/></div>	1.3.12.2.1107.5.2.43.67025.2019120416000098459528061.0.0.0.nii.gz	MR: Structural, T1, T2*, MPAGE	nifti		

<div><input type="checkbox"/> localizer_SBRef 2019-12-04 16:27</div>				<div></div>	
<div><input type="checkbox"/></div>	1.3.12.2.1107.5.2.43.67025.2019120416263566274397203.0.0.0.dicom.zip	MR: Localizer, T2, SBRef	dicom		
<div><input checked="" type="checkbox"/></div>	1.3.12.2.1107.5.2.43.67025.2019120416263566274397203.0.0.0.nii.gz	MR: Localizer, T2, SBRef	nifti		

Running analyses (manually)

Run Analysis Gear

Select an Analysis Gear from the following list:

ANTs: RegistrationSyN

0.0.4 (latest)

Runs antsRegistrationSyN from ANTs toolkit

Inputs

Configuration

Information

Select Input

fixed

No input chosen (Required)

Choose Input

mask1

No input chosen (Optional)

Choose Input

mask2

No input chosen (Optional)

Choose Input

mask3

No input chosen (Optional)

Choose Input

moving

No input chosen (Required)

Choose Input

Analysis Label

ants-antsregistrationsyn 12/14/2020 14:24:14

Tags

Close

Run Gear

Emotion Cognition > 2020_Shelby_HRVB_LCintensity	
SUBJECTS	
sub-5052	>
sub-5003	>
sub-6018	>
sub-8003	>
sub-8017	>
sub-5024	>
sub-5039	>
sub-6019	>
sub-7021	>
sub-5033	>
sub-5015	>
sub-5027	>
sub-6025	>
sub-5021	>
sub-5020	>
sub-6011	>
sub-5060	>
sub-8009	>

> = Suggested file

Select File

Tasks for which I am (so far) using Flywheel

- Assessing which data exist
- Downloading files
- Uploading files
- Running analyses

Reasons to use the SDK for these tasks (instead of the UI)

- The UI is prone to manual errors
- The UI takes time
- The UI limits reproducibility
- **The SDK reduces opportunities for errors, is efficient, allows for reproducibility at each step**

Overview of the Flywheel SDK

- The Flywheel SDK is a tool which allows you to programmatically access Flywheel
- Two options for using the SDK:
 - Flywheel SDK in python ([link](#))
 - Flywheel SDK in MATLAB ([link](#))
- Want free & open source but don't know python? **Solution: You can [use R](#) to call the python SDK.**

Dependencies for using the python SDK with R

Required

- R
- Python 3
- Python library `flywheel-sdk`
- R package `reticulate` (`object.function()` in python becomes `object$function()` in R)

Not required, but helpful

- RStudio
- R packages for data manipulation: `dplyr`, `stringr`, etc.

Connecting to Flywheel

First, you need to import the `flywheel-sdk` python module in R:

```
library(reticulate)
use_python(Sys.which('python3'))
flywheel <- import('flywheel')
```

Next, you need to make an API call to Flywheel:
(*Remember:* never hard code your API key in a script!)

```
my_key <- readline(prompt = 'Enter flywheel API key: ')
fw <- flywheel$Client(my_key)
rm(my_key)
```

Use these last three lines of code in R scripts *only*.
Knitting an R markdown document is more complicated, as `readline()` will not work.
Instead, you can use parameterized reports (described [here](#) and [here](#)).

Finding your group & project

```
group_name <- 'emocog'
project_name <- '2020_Shelby_HRVB_LCintensity'

# find your project within your group
project <- fw$lookup(paste(group_name, project_name, sep = '/'))

# examine the project object
project
```

```
## {'analyses': [{ 'created': datetime.datetime(2020, 10, 20, 4, 15, 9, 347000, tzinfo=tzutc()),  
##                  'description': None,  
##                  'files': [{ 'classification': {},  
##                             'created': datetime.datetime(2020, 10, 21, 14, 41, 46, 678000, tzinfo=tzutc()),  
##                             'deid_log_id': None,  
##                             'hash': '',  
##                             'id': '40b318d3-d7f9-40ff-80a8-c2f6506a355e',  
##                             'info': {},  
##                             'info_exists': None,  
##                             'mimetype': 'application/json',  
##                             'modality': None,  
##                             'modified': datetime.datetime(2020, 10, 21, 14, 41, 46, 678000, tzinfo=tzutc()),  
##                             'name': '.manifest.json',  
##                             'origin': { 'id': '5f8e644da5eb893b984a2dc0',  
##                                       'method': None,  
##                                       'name': None,  
##                                       'type': 'job',  
##                                       'via': None},
```

Description **Sessions** Information Analyses Data Views 


 Download Project


Actions  Advanced Filters   


Acquisitions Session Subject Project Analyses Provenance


<input type="checkbox"/>	Subject Label	Session Label	Timestamp
<input type="checkbox"/>	sub-5001		
<input type="checkbox"/>	ses-pre		
<input type="checkbox"/>	ses-post		
<input type="checkbox"/>	sub-5002		
<input type="checkbox"/>	sub-5003		
<input type="checkbox"/>	sub-5004		
<input type="checkbox"/>	sub-5005		
<input type="checkbox"/>	sub-5006		
<input type="checkbox"/>	sub-5007		
<input type="checkbox"/>	sub-5008		


ses-pre
No timestamp




BIDS View ☐ Run Gear 


☐  Description




☐  Classification

☐  All Types

☐ T1w 

 sub-5001_ses-pre_T1w.nii.gz	MR: Structural, T1	nifti
 .manifest.json	N/A	source code
 sub-5001_ses-pre_T1w_resamp.nii.gz	N/A	nifti

☐ acq-FSE_T1w 

 sub-5001_ses-pre_acq-FSE_T1w.nii.gz	MR: Structural, T1	nifti
 .manifest.json	N/A	source code
 sub-5001_ses-pre_acq-FSE_T1w_resamp.nii.gz	N/A	nifti

Session metadata

```
sessions <- project$sessions() # list of all sessions in your project  
length(sessions)
```

```
## [1] 325
```

```
sessions[[1]]$id # ID for first session
```

```
## [1] "5fce953899edcf4badbc9a9c"
```

```
sessions[[1]]$label # label for first session
```

```
## [1] "ses-pre"
```

```
sessions[[1]]$subject$label # label for subject for first session
```

```
## [1] "sub-5052"
```

By looping over sessions and using these commands, we can generate a dataframe with relevant session metadata:

label_subject	label_session	id_session	n_acquisitions
sub-5001	ses-post	5fce9f4b944e4f56afbc9cf5	2
sub-5001	ses-pre	5fce9f41944e4f56afbc9cf1	2
sub-5002	ses-post	5fce9fbd944e4f56afbc9d11	2
sub-5002	ses-pre	5fce9fb31871674588bc9abe	2
sub-5003	ses-post	5fce9557d063a87285bc9a55	2
sub-5003	ses-pre	5fce954d7415a11fd3bc9a31	2

Description **Sessions** Information Analyses Data Views 

 Download Project

Actions 

Advanced Filters 



Acquisitions

Session

Subject

Project

Analyses

Provenance















<input type="checkbox"/>	Subject Label	Session Label	Timestamp
2	sub-5001		
<input type="checkbox"/>	ses-pre		
<input type="checkbox"/>	ses-post		
2	sub-5002		
2	sub-5003		
2	sub-5004		
2	sub-5005		
2	sub-5006		
2	sub-5007		
1	sub-5008		

ses-pre

No timestamp

BIDS View 

Run Gear 

<input type="checkbox"/>		 Description		 Classification		All Types 
<input type="checkbox"/>		T1w				
		sub-5001_ses-pre_T1w.nii.gz		MR: Structural, T1		nifti
		.manifest.json		N/A		source code
		sub-5001_ses-pre_T1w_resamp.nii.gz		N/A		nifti
<input type="checkbox"/>		acq-FSE_T1w				
		sub-5001_ses-pre_acq-FSE_T1w.nii.gz		MR: Structural, T1		nifti
		.manifest.json		N/A		source code
		sub-5001_ses-pre_acq-FSE_T1w_resamp.nii.gz		N/A		nifti

Acquisitions within a session

```
# make the previously-referenced session an object
this_session <- sessions[[1]]

# alternative: this_session <- fw$get(sessions[[1]]$id)
```

```
acquisitions <- this_session$acquisitions() # list of acquisitions in session
length(acquisitions)
```

```
## [1] 2
```

```
acquisitions[[1]]$id # id for first acquisition
```

```
## [1] "5fce953899edcf4badbc9a9e"
```

```
acquisitions[[1]]$label # label for first acquisition
```

```
## [1] "acq-FSE_T1w"
```

By looping over sessions and acquisitions, and using these commands, we can generate a dataframe with relevant acquisition metadata.

Below, I looped over sessions & stored metadata only for those acquisitions with my label of interest, **acq-FSE_T1w**:

label_subject	label_session	id_session	label_acq	id_acq
sub-5001	ses-post	5fce9f4b944e4f56afbc9cf5	acq-FSE_T1w	5fce9f4b60eb8063a7bc9b39
sub-5001	ses-pre	5fce9f41944e4f56afbc9cf1	acq-FSE_T1w	5fce9f485dadee3cd2bc9bda
sub-5002	ses-post	5fce9fbd944e4f56afbc9d11	acq-FSE_T1w	5fce9fc4944e4f56afbc9d14
sub-5002	ses-pre	5fce9fb31871674588bc9abe	acq-FSE_T1w	5fce9fba1871674588bc9abf
sub-5003	ses-post	5fce9557d063a87285bc9a55	acq-FSE_T1w	5fce955799edcf4badbc9aa5
sub-5003	ses-pre	5fce954d7415a11fd3bc9a31	acq-FSE_T1w	5fce954e1871674588bc9a46

Files within an acquisition

```
# make the previously-referenced acquisition an object
this_acq <- acquisitions[[1]]

# alternative: this_session <- fw$get(acquisitions[[1]]$id)
```

```
files <- this_acq$files # list of files in acquisition
length(files)
```

```
## [1] 3
```

```
files[[1]]$name # name for first file
```

```
## [1] "sub-6015_ses-post_T1w.nii.gz"
```

```
files[[1]]$type # type for first file
```

```
## [1] "nifti"
```


Now we could extend our dataframe and include filenames and types as well:

As before, below I stored metadata only for `nifti` files, with name ending in `*_acq-FSE_T1w.nii.gz` within acquisitions with my label of interest, `acq-FSE_T1w`:

label_subject	label_session	label_acq	id_acq	name_file
sub-5001	ses-post	acq-FSE_T1w	5fce9f4b60eb8063a7bc9b39	sub-5001_ses-post_acq-FSE_T1w.nii.gz
sub-5001	ses-pre	acq-FSE_T1w	5fce9f485dadee3cd2bc9bda	sub-5001_ses-pre_acq-FSE_T1w.nii.gz
sub-5002	ses-post	acq-FSE_T1w	5fce9fc4944e4f56afbc9d14	sub-5002_ses-post_acq-FSE_T1w.nii.gz
sub-5002	ses-pre	acq-FSE_T1w	5fce9fba1871674588bc9abf	sub-5002_ses-pre_acq-FSE_T1w.nii.gz
sub-5003	ses-post	acq-FSE_T1w	5fce955799edcf4badbc9aa5	sub-5003_ses-post_acq-FSE_T1w.nii.gz
sub-5003	ses-pre	acq-FSE_T1w	5fce954e1871674588bc9a46	sub-5003_ses-pre_acq-FSE_T1w.nii.gz

Downloading a file

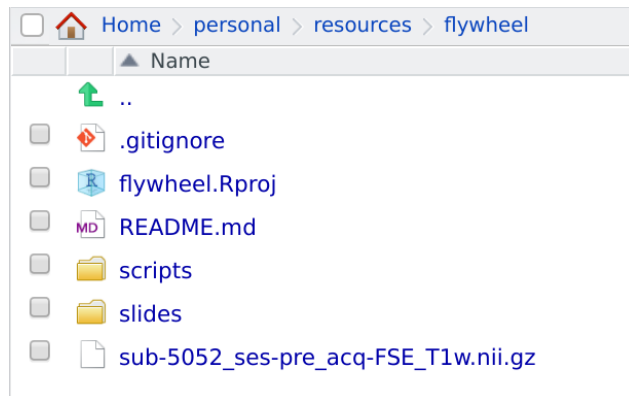
Once we've identified a file within an acquisition of interest, we can download it:

```
file_to_download <- files[[1]]$name # set name of file to download
files[[1]]$name
```

```
## [1] "sub-5052_ses-pre_acq-FSE_T1w.nii.gz"
```

```
path_for_download <- here(file_to_download) # set path to download file
```

```
files[[1]]$download(dest_file = path_for_download) # download the file
```



Links

Slides and scripts: <https://github.com/shelbybachman/flywheel>