
pypmi

Release 0.2+22.g816c304.dirty

Mar 02, 2020

Table of Contents

1	Usage	3
2	Reference API	5
	Python Module Index	11
	Index	13

The PPMI is an ongoing longitudinal study that began in early 2010 with the primary goal of identifying biomarkers of Parkinson's disease (PD) progression. To date, the PPMI has collected data from over 400 individuals with de novo PD and nearly 200 age-matched healthy participants, in addition to large cohorts of individuals genetically at-risk for PD. Data, made available on the [PPMI website](#), include comprehensive clinical-behavioral assessments, biological assays, single-photon emission computed tomography (SPECT) images, and magnetic resonance imaging (MRI) scans.

While accessing this data is straightforward (researchers must simply sign a data usage agreement and provide information on the purpose of their research), the sheer amount of data made available can be quite overwhelming to work with. Thus, the primary goal of this package is to provide a Python interface to making working with the data provided by the PPMI easier.

While this project is still very much under development it is nevertheless functional. However, **please note that this project's functionality is liable to change quite dramatically until an initial release is made**—so be careful! Check out our [reference API](#) for some of the current capabilities of `pypmi` while our [user guide](#) is under construction.

1.1 Getting the data

First things first: you need to get the data! Once you have access to the [PPMI database](#), log in to the database and follow these instructions:

1. Select Download from the navigation bar at the top
2. Select Study Data from the options that appear in the navigation bar
3. Select ALL at the bottom of the left-hand navigation bar on the new page
4. Click Select ALL tabular data (csv) format and then press Download>> in the top right hand corner of the page
5. Unzip the downloaded directory and save it somewhere on your computer

Alternatively, you can use [pypmi](#) module to download the data programatically:

```
>>> import pypmi
>>> files = pypmi.fetch_studydata('all', user='username', password='password')
Fetching authentication key for data download...
Requesting 113 datasets for download...
Downloading PPMI data: 17.3MB [00:33, 519kB/s]
```

By default, the data will be downloaded to your current directory making it easy to load them in the future, but you can optionally provide a path argument to [pypmi.fetch_studydata\(\)](#) to specify where you would like the data to go. (Alternatively, you can set an environmental variable `$PPMI_PATH` to specify where they should be downloaded to; this takes precedence over the current directory.)

1.2 Loading and working with the data

Once you have the data downloaded you can use the functions to load various portions of it into [tidy](#) data frames.

For example, we can generate a number of clinical-behavioral measures:

```
>>> behavior = pypmi.load_behavior()
>>> behavior.columns
Index(['participant', 'visit', 'date', 'benton', 'epworth', 'gds',
      'hvlrt_recall', 'hvlrt_recognition', 'hvlrt_retention', 'lns', 'moca',
      'pigd', 'quip', 'rbd', 'scopa_aut', 'se_adl', 'semantic_fluency',
      'stai_state', 'stai_trait', 'symbol_digit', 'systolic_bp_drop',
      'tremor', 'updrs_i', 'updrs_ii', 'updrs_iii', 'updrs_iii_a', 'updrs_iv',
      'upsit'],
      dtype='object')
```

The call to `pypmi.load_behavior()` may take a few seconds to run—there’s a lot of data to import and wrangle!

If we want to query the data with regards to, say, subject diagnosis it might be useful to load in some demographic information:

```
>>> demographics = pypmi.load_demographics()
>>> demographics.columns
Index(['participant', 'diagnosis', 'date_birth', 'date_diagnosis',
      'date_enroll', 'status', 'family_history', 'age', 'gender', 'race',
      'site', 'handedness', 'education'],
      dtype='object')
```

Now we can perform some interesting queries! As an example, let’s just ask how many individuals with Parkinson’s disease have a baseline UPDRS III score. We’ll have to use information from both data frames to answer the question:

```
>>> import pandas as pd
>>> updrs = (behavior.query('visit == "BL" & ~updrs_iii.isna()')
...         .get(['participant', 'updrs_iii']))
>>> parkinsons = demographics.query('diagnosis == "pd"').get('participant')
>>> len(pd.merge(parkinsons, updrs, on='participant'))
423
```

And the same for healthy individuals:

```
>>> healthy = demographics.query('diagnosis == "hc"').get('participant')
>>> len(pd.merge(healthy, updrs))
195
```

There’s a lot of power gained in leveraging the pandas DataFrame objects, so take a look at the [pandas documentation](#) to see what more you can do!

This is the primary reference of `pypmi`. Please refer to the [user guide](#) for more information on how to best implement these functions in your own workflows.

2.1 `pypmi` - Dataset fetchers and loaders

Functions for listing and downloading datasets from the PPMI database:

<code>fetchable_studydata()</code>	Lists study data available to download from the PPMI
<code>fetchable_genetics(projects)</code>	Lists genetics data available to download from the PPMI
<code>fetch_studydata(*datasets, path, user, ...)</code>	Downloads specified study data <i>datasets</i> from the PPMI database
<code>fetch_genetics(*datasets, path, user, ...)</code>	Downloads specified genetics data <i>datasets</i> from the PPMI database

2.1.1 `pypmi.fetchable_studydata`

`pypmi.fetchable_studydata()` → List[str]
 Lists study data available to download from the PPMI

Returns `available` – List of available data files

Return type list

See also:

`pypmi.fetch_studydata()`

2.1.2 pypmi.fetchable_genetics

`pypmi.fetchable_genetics` (*projects: bool = False*) → List[str]

Lists genetics data available to download from the PPMI

Parameters `projects` (*bool, optional*) – List available projects instead of individual data files available for download. Due to the size of genetic data, many datasets are split up into multiple files associated with a single project or analysis; you can specify these projects when downloading data with `pypmi.datasets.fetch_genetics()` and all associated files will be fetched.

Returns `available` – List of available data files

Return type list

See also:

`pypmi.fetch_genetics()`

2.1.3 pypmi.fetch_studydata

`pypmi.fetch_studydata` (**datasets, path: str = None, user: str = None, password: str = None, overwrite: bool = False, verbose: bool = True*) → List[str]

Downloads specified study data *datasets* from the PPMI database

Parameters

- ***datasets** (*str*) – Datasets to download. Can provide as many as desired, but they should be listed in `pypmi.fetchable_studydata()`. Alternatively, if any of the provided values are ‘all’, then all available datasets will be fetched.
- **path** (*str, optional*) – Filepath where downloaded data should be saved. If data files already exist at *path* they will be overwritten unless *overwrite=False*. If not supplied the current directory is used. Default: None
- **user** (*str, optional*) – Email for user authentication to the LONI IDA database. If not supplied will look for \$PPMI_USER variable in environment. Default: None
- **password** (*str, optional*) – Password for user authentication to the LONI IDA database. If not supplied will look for \$PPMI_PASSWORD variable in environment. Default: None
- **overwrite** (*bool, optional*) – Whether to overwrite existing PPMI data files at *path* if they already exist. Default: False
- **verbose** (*bool, optional*) – Whether to print progress bar as download occurs. Default: True

Returns `downloaded` – Filepath(s) to downloaded datasets

Return type list

See also:

`pypmi.fetchable_studydata()`

2.1.4 pypmi.fetch_genetics

`pypmi.fetch_genetics` (**datasets, path: str = None, user: str = None, password: str = None, overwrite: bool = False, verbose: bool = True*) → List[str]

Downloads specified genetics data *datasets* from the PPMI database

Parameters

- ***datasets** (*str*) – Datasets to download. Can provide as many as desired, but they should be listed in `pypmi.fetchable_genetics()`. Alternatively, if any of the provided values are ‘all’, then all available datasets will be fetched.
- **path** (*str*, *optional*) – Filepath where downloaded data should be saved. If data files already exist at *path* they will be overwritten unless *overwrite=False*. If not supplied the current directory is used. Default: None
- **user** (*str*, *optional*) – Email for user authentication to the LONI IDA database. If not supplied will look for \$PPMI_USER variable in environment. Default: None
- **password** (*str*, *optional*) – Password for user authentication to the LONI IDA database. If not supplied will look for \$PPMI_PASSWORD variable in environment. Default: None
- **overwrite** (*bool*, *optional*) – Whether to overwrite existing PPMI data files at *path* if they already exist. Default: False
- **verbose** (*bool*, *optional*) – Whether to print progress bar as download occurs. Default: True

Returns **downloaded** – Filepath(s) to downloaded datasets

Return type list

See also:

`pypmi.fetchable_genetics()`

Functions for loading data from PPMI database into tidy dataframes:

<code>load_behavior(path, measures)</code>	Loads clinical-behavioral data into tidy dataframe
<code>load_biospecimen(path, measures)</code>	Loads biospecimen data into tidy dataframe
<code>load_datscan(path, measures)</code>	Loads DaT scan data into tidy dataframe
<code>load_demographics(path, measures)</code>	Loads demographic data into tidy dataframe

2.1.5 pypmi.load_behavior

`pypmi.load_behavior(path: str = None, measures: List[str] = None) → pandas.core.frame.DataFrame`
 Loads clinical-behavioral data into tidy dataframe

Parameters

- **path** (*str*, *optional*) – Filepath to directory containing PPMI data files. If not specified this function will, in order, look (1) for an environmental variable \$PPMI_PATH and (2) in the current directory. Default: None
- **measures** (*list*, *optional*) – Which measures to keep in the final dataframe. If not specified all measures are retained; available behavioral measures can be viewed with `pypmi.available_behavior()`. Default: None

Returns **df** – Tidy DataFrame with all clinical-behavioral assessments

Return type `pandas.DataFrame`

See also:

`pypmi.available_behavior()`

2.1.6 pypmi.load_biospecimen

`pypmi.load_biospecimen` (*path*: *str* = *None*, *measures*: *List[str]* = *None*) → `pandas.core.frame.DataFrame`
Loads biospecimen data into tidy dataframe

Parameters

- **path** (*str*, *optional*) – Filepath to directory containing PPMI data files. If not specified this function will, in order, look (1) for an environmental variable \$PPMI_PATH and (2) in the current directory. Default: *None*
- **measures** (*list*, *optional*) – Which measures to keep in the final dataframe. There are a number of biospecimen measures that are missing for large numbers of subjects, so if not specified only those that are present in at least 80% of entries are kept. Specifying *measures='all'* will retain everything, but this will significantly increase load time. It is highly recommended to specify which measures to keep; available biospecimen measures can be viewed with `pypmi.available_biospecimen()`. Default: *None*

Returns data – Biospecimen data

Return type `pandas.DataFrame`

See also:

`pypmi.available_biospecimen()`

2.1.7 pypmi.load_datscan

`pypmi.load_datscan` (*path*: *str* = *None*, *measures*: *List[str]* = *None*) → `pandas.core.frame.DataFrame`
Loads DaT scan data into tidy dataframe

Parameters

- **path** (*str*, *optional*) – Filepath to directory containing PPMI data files. If not specified this function will, in order, look (1) for an environmental variable \$PPMI_PATH and (2) in the current directory. Default: *None*
- **measures** (*list*, *optional*) – Which measures to keep in the final dataframe. If not specified all measures are retained; available DaT scan measures can be viewed with `pypmi.available_datscan()`. Default: *None*

Returns data – DaTScan data

Return type `pandas.DataFrame`

See also:

`pypmi.available_datscan()`

2.1.8 pypmi.load_demographics

`pypmi.load_demographics` (*path*: *str* = *None*, *measures*: *List[str]* = *None*) → `pandas.core.frame.DataFrame`
Loads demographic data into tidy dataframe

Parameters

- **path** (*str*, *optional*) – Filepath to directory containing PPMI data files. If not specified this function will, in order, look (1) for an environmental variable \$PPMI_PATH and (2) in the current directory. Default: *None*

- **measures** (*list, optional*) – Which measures to keep in the final dataframe. If not specified all measures are retained; available demographics measures can be viewed with `pypmi.available_demographics()`. Default: None

Returns **demographics** – Tidy data frame containing demographic information for PPMI subjects

Return type `pandas.DataFrame`

See also:

`pypmi.available_demographics()`

Functions for listing measures available from relevant `pypmi.load_X()` commands:

<code>available_behavior(path)</code>	Lists measures available in <code>pypmi.load_behavior()</code>
<code>available_biospecimen(path)</code>	Lists measures available in <code>pypmi.load_biospecimen()</code>
<code>available_datscan(path)</code>	Lists measures available in <code>pypmi.load_datscan()</code>
<code>available_demographics(path)</code>	Lists measures available in <code>pypmi.load_demographics()</code>

2.1.9 pypmi.available_behavior

`pypmi.available_behavior(path: str = None) → List[str]`

Lists measures available in `pypmi.load_behavior()`

Parameters **path** (*str, optional*) – Filepath to directory containing PPMI data files. If not specified this function will, in order, look (1) for an environmental variable \$PPMI_PATH and (2) in the current directory. Default: None

Returns **measures** – Available behavioral measures

Return type `list`

See also:

`pypmi.load_behavior()`

2.1.10 pypmi.available_biospecimen

`pypmi.available_biospecimen(path: str = None) → List[str]`

Lists measures available in `pypmi.load_biospecimen()`

Parameters **path** (*str, optional*) – Filepath to directory containing PPMI data files. If not specified this function will, in order, look (1) for an environmental variable \$PPMI_PATH and (2) in the current directory. Default: None

Returns **measures** – Available biospecimen measures (i.e., “tests”)

Return type `list`

See also:

`pypmi.load_biospecimen()`

2.1.11 pypmi.available_datscan

`pypmi.available_datscan` (*path*: *str* = *None*) → List[str]

Lists measures available in `pypmi.load_datscan()`

Parameters `path` (*str*, *optional*) – Filepath to directory containing PPMI data files. If not specified this function will, in order, look (1) for an environmental variable \$PPMI_PATH and (2) in the current directory. Default: None

Returns `measures` – Available DaT scan measures

Return type list

See also:

`pypmi.load_datscan()`

2.1.12 pypmi.available_demographics

`pypmi.available_demographics` (*path*: *str* = *None*) → List[str]

Lists measures available in `pypmi.load_demographics()`

Parameters `path` (*str*, *optional*) – Filepath to directory containing PPMI data files. If not specified this function will, in order, look (1) for an environmental variable \$PPMI_PATH and (2) in the current directory. Default: None

Returns `measures` – Available demographic measures

Return type list

See also:

`pypmi.load_demographics()`

p

`pypmi`, 5

A

`available_behavior()` (*in module pypmi*), 9
`available_biospecimen()` (*in module pypmi*), 9
`available_datscan()` (*in module pypmi*), 10
`available_demographics()` (*in module pypmi*), 10

F

`fetch_genetics()` (*in module pypmi*), 6
`fetch_studydata()` (*in module pypmi*), 6
`fetchable_genetics()` (*in module pypmi*), 6
`fetchable_studydata()` (*in module pypmi*), 5

L

`load_behavior()` (*in module pypmi*), 7
`load_biospecimen()` (*in module pypmi*), 8
`load_datscan()` (*in module pypmi*), 8
`load_demographics()` (*in module pypmi*), 8

P

`pypmi` (*module*), 5