

DataPipeline

Generated by Doxygen 1.8.17

1 Usage:	1
2 Namespace Index	3
2.1 Packages	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 run_pipeline Namespace Reference	11
6.1.1 Variable Documentation	11
6.1.1.1 action	11
6.1.1.2 args	11
6.1.1.3 help	11
6.1.1.4 parser	12
6.1.1.5 str	12
6.1.1.6 type	12
6.2 scripts Namespace Reference	12
6.3 scripts.cleaner Namespace Reference	12
6.3.1 Detailed Description	13
6.3.2 Function Documentation	13
6.3.2.1 check_correlated_column()	13
6.3.2.2 check_outliers()	13
6.3.2.3 create_correlation_graphs()	14
6.3.2.4 create_mask()	14
6.3.2.5 get_correlated_descriptors()	14
6.3.2.6 get_duplicit_correlated_descriptors()	15
6.3.2.7 load_data()	15
6.3.2.8 main()	15
6.3.2.9 plot_correlation_heatmap()	15
6.3.2.10 process_config()	16
6.3.2.11 remove_duplicits()	16
6.3.2.12 remove_useless_columns()	16
6.3.2.13 set_proper_data_type()	16
6.3.2.14 unique_count()	17
6.3.3 Variable Documentation	17
6.3.3.1 args	17
6.3.3.2 input_data	17

6.3.3.3 output_data	17
6.3.3.4 project_dir	17
6.3.3.5 script_dir	17
6.4 scripts.crawlers Namespace Reference	17
6.5 scripts.crawlers.crawler_orchestrator Namespace Reference	18
6.5.1 Function Documentation	18
6.5.1.1 main()	18
6.6 scripts.crawlers.dependencies Namespace Reference	18
6.7 scripts.crawlers.dependencies.pubchem Namespace Reference	18
6.7.1 Function Documentation	18
6.7.1.1 crawl()	18
6.8 scripts.crawlers.dependencies.swissadme Namespace Reference	18
6.8.1 Function Documentation	19
6.8.1.1 crawl()	19
6.8.1.2 is_file_downloaded()	19
6.9 scripts.crawlers.dependencies.swisstarget Namespace Reference	19
6.9.1 Function Documentation	19
6.9.1.1 crawl()	19
6.9.1.2 is_file_downloaded()	19
6.10 scripts.merger Namespace Reference	20
6.10.1 Function Documentation	20
6.10.1.1 merge()	20
6.11 scripts.pipe_handler Namespace Reference	20
6.12 scripts.populate_chem Namespace Reference	20
6.12.1 Function Documentation	20
6.12.1.1 populate()	20
7 Class Documentation	21
7.1 scripts.pipe_handler.IgnoreBrokenPipe Class Reference	21
7.1.1 Constructor & Destructor Documentation	22
7.1.1.1 __init__()	22
7.1.2 Member Data Documentation	22
7.1.2.1 flush	22
7.1.2.2 stream	22
7.1.2.3 write	22
8 File Documentation	23
8.1 README.md File Reference	23
8.2 run_pipeline.py File Reference	23
8.3 scripts/__init__.py File Reference	23
8.4 scripts/crawlers/__init__.py File Reference	23
8.5 scripts/crawlers/dependencies/__init__.py File Reference	24
8.6 scripts/cleaner.py File Reference	24

8.7 scripts/crawlers/crawler_orchestrator.py File Reference	24
8.8 scripts/crawlers/dependencies/pubchem.py File Reference	25
8.9 scripts/crawlers/dependencies/swissadme.py File Reference	25
8.10 scripts/crawlers/dependencies/swisstarget.py File Reference	25
8.11 scripts/merger.py File Reference	25
8.12 scripts/pipe_handler.py File Reference	26
8.13 scripts/populate_chem.py File Reference	26
Index	27

Chapter 1

Usage:

Run the file [run_pipeline.py](#) after installing python using conda from environment.yml

[run_pipeline.py](#) supports the following arguments:

- **-inputfile "path"** to specify .csv containing [phototoxic, name] pairs, where phototoxic is either 0 or 1
- **-outputfile "path"** to specify final output for .csv file
- **-noscrrape** to skip the scraping using crawlers

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

run_pipeline	11
scripts	12
scripts.cleaner	12
scripts.crawlers	17
scripts.crawlers.crawler_orchestrator	18
scripts.crawlers.dependencies	18
scripts.crawlers.dependencies.pubchem	18
scripts.crawlers.dependencies.swissadme	18
scripts.crawlers.dependencies.swisstarget	19
scripts.merger	20
scripts.pipe_handler	20
scripts.populate_chem	20

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

object	
scripts.pipe_handler.IgnoreBrokenPipe	21

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

[scripts.pipe_handler.IgnoreBrokenPipe](#) 21

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

run_pipeline.py	23
scripts/__init__.py	23
scripts/cleaner.py	24
scripts/merger.py	25
scripts/pipe_handler.py	26
scripts/populate_chem.py	26
scripts/crawlers/__init__.py	23
scripts/crawlers/crawler_orchestrator.py	24
scripts/crawlers/dependencies/__init__.py	24
scripts/crawlers/dependencies/pubchem.py	25
scripts/crawlers/dependencies/swissadme.py	25
scripts/crawlers/dependencies/swisstarget.py	25

Chapter 6

Namespace Documentation

6.1 run_pipeline Namespace Reference

Variables

- `parser` = `argparse.ArgumentParser()`
- `type`
- `str`
- `help`
- `action`
- `args` = `parser.parse_args()`

6.1.1 Variable Documentation

6.1.1.1 action

```
run_pipeline.action
```

6.1.1.2 args

```
run_pipeline.args = parser.parse_args()
```

6.1.1.3 help

```
run_pipeline.help
```

6.1.1.4 parser

```
run_pipeline.parser = argparse.ArgumentParser()
```

6.1.1.5 str

```
run_pipeline.str
```

6.1.1.6 type

```
run_pipeline.type
```

6.2 scripts Namespace Reference

Namespaces

- [cleaner](#)
- [crawlers](#)
- [merger](#)
- [pipe_handler](#)
- [populate_chem](#)

6.3 scripts.cleaner Namespace Reference

Functions

- `pd.DataFrame` [load_data](#) (str csv_file)
- `pd.DataFrame` [set_proper_data_type](#) (pd.DataFrame df)
- [str, int] [unique_count](#) (pd.Series column)
- `pd.DataFrame` [remove_useless_columns](#) (df)
- list [get_duplicity_correlated_descriptors](#) (nx.MultiGraph graphs, exclude)
- def [plot_correlation_heatmap](#) (pd.DataFrame correlations, str fig_location, int threshold)
- def [get_correlated_descriptors](#) (pd.DataFrame df, int threshold, str fig_location)
- def [create_correlation_graphs](#) (correlations, fig_location, threshold)
- def [create_mask](#) (df, mask_name, [None, list] columns=None, [None, list] rows=None, data_dir='../data')
- `pd.DataFrame` [check_correlated_column](#) (df, threshold=0.9, remove=False, preserve_columns=[], plot_dir='../plot', data_dir='../data')
- def [remove_duplicity](#) (pd.DataFrame df, subset, keep='first')
- def [check_outliers](#) (df, threshold=4.2, remove=False)
- def [main](#) (input_file='../data/chem_output/chem_populated.csv', output_file='../data/final/phototox.csv', [project_dir](#)='../', correlation_threshold=0.95, outliers_threshold=4.2, preserve_columns=None, remove=False)
- dict [process_config](#) (config_file='../conf/cleaner.ini')

Variables

- `script_dir` = `os.path.dirname(os.path.realpath(__file__))`
- `project_dir` = `os.path.dirname(script_dir)`
- dict `args` = `process_config(os.path.join(project_dir, 'conf/cleaner.ini'))`
- `input_data` = `os.path.join(project_dir, 'data/chem_output/chem_populated.csv')`
- `output_data` = `os.path.join(project_dir, 'data/chem_output/phototox.csv')`

6.3.1 Detailed Description

Clean input data. Deletes unuseful columns. Correlations columns and outliers are ether deleted or a mask is c

6.3.2 Function Documentation

6.3.2.1 `check_correlated_column()`

```
pd.DataFrame scripts.cleaner.check_correlated_column (
    df,
    threshold = 0.9,
    remove = False,
    preserve_columns = [],
    plot_dir = '../plot',
    data_dir = '../data' )
```

Check weather some columns are not correlated over given threshold. If yes, remove them or create mask.

```
:param df: input DataFrame
:param threshold: correlation threshold
:param remove: if True, remove these columns, otherwise make mask
:param preserve_columns: columns, taht are always perseved
:param plot_dir: dir to save generated correlation plot
:param data_dir: dir to store masks
:return: output DataFrame
```

6.3.2.2 `check_outliers()`

```
def scripts.cleaner.check_outliers (
    df,
    threshold = 4.2,
    remove = False )
```

Check weather some outliners over given threshold (standart deviation). If yes, remove them, otherwise create

```
:param df: input DtaFrame
:param threshold: standart deviation
:param remove: if True, remove these columns, otherwise make mask
:return: output DataFrame
```

6.3.2.3 create_correlation_graphs()

```
def scripts.cleaner.create_correlation_graphs (
    correlations,
    fig_location,
    threshold )

:param correlations:
:param fig_location:
:param threshold:
:return:
```

6.3.2.4 create_mask()

```
def scripts.cleaner.create_mask (
    df,
    mask_name,
    [None, list] columns = None,
    [None, list] rows = None,
    data_dir = '../data' )
```

Creates mask from columns or rows.

```
:param df: input Dataframe
:param mask_name: mask name
:param columns: input columns to mask
:param rows: input rows to mask
:param data_dir: dir to store mask
```

6.3.2.5 get_correlated_descriptors()

```
def scripts.cleaner.get_correlated_descriptors (
    pd.DataFrame df,
    int threshold,
    str fig_location )
```

This function creates correlation between descriptors

```
:param df: input DataFrame
:param threshold: threshold for minimal correlation
:param fig_location: location of correlation figure
:return: created correlations
```

6.3.2.6 get_duplicit_correlated_descriptors()

```
list scripts.cleaner.get_duplicit_correlated_descriptors (
    nx.MultiGraph graphs,
    exclude )
```

Fnc extract corelated descriptors (column names) from graphs. From each correlation group is excluded one item or if some items from exclude are in correlation, these items are excluded instead.

:param *graphs*: input graphs
:param *exclude*: list of descriptors to exclude from final result
:return: list of correlated descriptors without excluded

6.3.2.7 load_data()

```
pd.DataFrame scripts.cleaner.load_data (
    str csv_file )
```

Load data from csv to pandas DataFrame
:param *csv_file*: csv file name
:return: loaded DataFrame

6.3.2.8 main()

```
def scripts.cleaner.main (
    input_file = '../data/chem_output/chem_populated.csv',
    output_file = '../data/final/phototox.csv',
    project_dir = '../',
    correlation_threshold = 0.95,
    outliers_threshold = 4.2,
    preserve_columns = None,
    remove = False )
```

6.3.2.9 plot_correlation_heatmap()

```
def scripts.cleaner.plot_correlation_heatmap (
    pd.DataFrame correlations,
    str fig_location,
    int threshold )
```

Plot heatmap of correlated descriptors
:param *correlations*: input DataFrame with correlations
:param *fig_location*: path to store plot
:param *threshold*: used threshold to generate correlations

6.3.2.10 process_config()

```
dict scripts.cleaner.process_config (
    config_file = '../conf/cleaner.ini' )
```

Process input config and extract
:param config_file: config file name location
:return: configuration

6.3.2.11 remove_duplicits()

```
def scripts.cleaner.remove_duplicits (
    pd.DataFrame df,
    subset,
    keep = 'first' )
```

Removes duplicit rows
:param df: nput DataFrame
:param subset: columns where to check duplicity
:param keep: keep argument
:return: output DataFrame

6.3.2.12 remove_useless_columns()

```
pd.DataFrame scripts.cleaner.remove_useless_columns (
    df )
```

Remove columns from DataFrame, that are duplicated from multiple sources or otherwise unuseful
:param df: input DataFrame
:return: reduced DataFrame

6.3.2.13 set_proper_data_type()

```
pd.DataFrame scripts.cleaner.set_proper_data_type (
    pd.DataFrame df )
```

Set data types to df columns.
:param df: input DataFrame
:return: df with changed types

6.3.2.14 unique_count()

```
[str, int] scripts.cleaner.unique_count (
    pd.Series column )
```

Get number of unique values in Series
:param column: input Series
:return: columns name and number of unique values

6.3.3 Variable Documentation

6.3.3.1 args

```
dict scripts.cleaner.args = process_config(os.path.join(project_dir, 'conf/cleaner.ini'))
```

6.3.3.2 input_data

```
scripts.cleaner.input_data = os.path.join(project_dir, 'data/chem_output/chem_populated.csv')
```

6.3.3.3 output_data

```
scripts.cleaner.output_data = os.path.join(project_dir, 'data/chem_output/phototox.csv')
```

6.3.3.4 project_dir

```
scripts.cleaner.project_dir = os.path.dirname(script_dir)
```

6.3.3.5 script_dir

```
scripts.cleaner.script_dir = os.path.dirname(os.path.realpath(__file__))
```

6.4 scripts.crawlers Namespace Reference

Namespaces

- [crawler_orchestrator](#)
- [dependencies](#)

6.5 scripts.crawlers.crawler_orchestrator Namespace Reference

Functions

- def [main](#) (file)

6.5.1 Function Documentation

6.5.1.1 main()

```
def scripts.crawlers.crawler_orchestrator.main (  
    file )
```

6.6 scripts.crawlers.dependencies Namespace Reference

Namespaces

- [pubchem](#)
- [swissadme](#)
- [swisstarget](#)

6.7 scripts.crawlers.dependencies.pubchem Namespace Reference

Functions

- def [crawl](#) (prefs)

6.7.1 Function Documentation

6.7.1.1 crawl()

```
def scripts.crawlers.dependencies.pubchem.crawl (  
    prefs )
```

6.8 scripts.crawlers.dependencies.swissadme Namespace Reference

Functions

- def [is_file_downloaded](#) (filename, timeout=5)
- def [crawl](#) (prefs)

6.8.1 Function Documentation

6.8.1.1 crawl()

```
def scripts.crawlers.dependencies.swissadme.crawl (  
    prefs )
```

6.8.1.2 is_file_downloaded()

```
def scripts.crawlers.dependencies.swissadme.is_file_downloaded (  
    filename,  
    timeout = 5 )
```

6.9 scripts.crawlers.dependencies.swisstarget Namespace Reference

Functions

- def [is_file_downloaded](#) (filename, timeout=5)
- def [crawl](#) (prefs)

6.9.1 Function Documentation

6.9.1.1 crawl()

```
def scripts.crawlers.dependencies.swisstarget.crawl (  
    prefs )
```

6.9.1.2 is_file_downloaded()

```
def scripts.crawlers.dependencies.swisstarget.is_file_downloaded (  
    filename,  
    timeout = 5 )
```

6.10 scripts.merger Namespace Reference

Functions

- def [merge](#) ()

6.10.1 Function Documentation

6.10.1.1 merge()

```
def scripts.merger.merge ( )
```

6.11 scripts.pipe_handler Namespace Reference

Classes

- class [IgnoreBrokenPipe](#)

6.12 scripts.populate_chem Namespace Reference

Functions

- def [populate](#) ()

6.12.1 Function Documentation

6.12.1.1 populate()

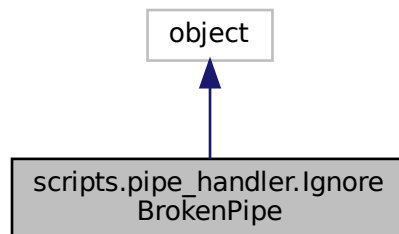
```
def scripts.populate_chem.populate ( )
```

Chapter 7

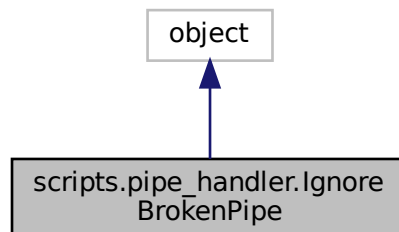
Class Documentation

7.1 `scripts.pipe_handler.IgnoreBrokenPipe` Class Reference

Inheritance diagram for `scripts.pipe_handler.IgnoreBrokenPipe`:



Collaboration diagram for `scripts.pipe_handler.IgnoreBrokenPipe`:



Public Member Functions

- `def __init__(self, stream)`

Public Attributes

- [stream](#)
- [write](#)
- [flush](#)

7.1.1 Constructor & Destructor Documentation

7.1.1.1 `__init__()`

```
def scripts.pipe_handler.IgnoreBrokenPipe.__init__ (
    self,
    stream )
```

7.1.2 Member Data Documentation

7.1.2.1 `flush`

```
scripts.pipe_handler.IgnoreBrokenPipe.flush
```

7.1.2.2 `stream`

```
scripts.pipe_handler.IgnoreBrokenPipe.stream
```

7.1.2.3 `write`

```
scripts.pipe_handler.IgnoreBrokenPipe.write
```

The documentation for this class was generated from the following file:

- [scripts/pipe_handler.py](#)

Chapter 8

File Documentation

8.1 README.md File Reference

8.2 run_pipeline.py File Reference

Namespaces

- [run_pipeline](#)

Variables

- [run_pipeline.parser](#) = argparse.ArgumentParser()
- [run_pipeline.type](#)
- [run_pipeline.str](#)
- [run_pipeline.help](#)
- [run_pipeline.action](#)
- [run_pipeline.args](#) = parser.parse_args()

8.3 scripts/__init__.py File Reference

Namespaces

- [scripts](#)

8.4 scripts/crawlers/__init__.py File Reference

Namespaces

- [scripts.crawlers](#)

8.5 scripts/crawlers/dependencies/__init__.py File Reference

Namespaces

- [scripts.crawlers.dependencies](#)

8.6 scripts/cleaner.py File Reference

Namespaces

- [scripts.cleaner](#)

Functions

- `pd.DataFrame` [scripts.cleaner.load_data](#) (str csv_file)
- `pd.DataFrame` [scripts.cleaner.set_proper_data_type](#) (pd.DataFrame df)
- [str, int] [scripts.cleaner.unique_count](#) (pd.Series column)
- `pd.DataFrame` [scripts.cleaner.remove_useless_columns](#) (df)
- list [scripts.cleaner.get_duplicat_correlated_descriptors](#) (nx.MultiGraph graphs, exclude)
- def [scripts.cleaner.plot_correlation_heatmap](#) (pd.DataFrame correlations, str fig_location, int threshold)
- def [scripts.cleaner.get_correlated_descriptors](#) (pd.DataFrame df, int threshold, str fig_location)
- def [scripts.cleaner.create_correlation_graphs](#) (correlations, fig_location, threshold)
- def [scripts.cleaner.create_mask](#) (df, mask_name, [None, list] columns=None, [None, list] rows=None, data_dir='../data')
- `pd.DataFrame` [scripts.cleaner.check_correlated_column](#) (df, threshold=0.9, remove=False, preserve_columns=[], plot_dir='../plot', data_dir='../data')
- def [scripts.cleaner.remove_duplicits](#) (pd.DataFrame df, subset, keep='first')
- def [scripts.cleaner.check_outliers](#) (df, threshold=4.2, remove=False)
- def [scripts.cleaner.main](#) (input_file='../data/chem_output/chem_populated.csv', output_file='../data/final/phototox.csv', project_dir='../', correlation_threshold=0.95, outliers_threshold=4.2, preserve_columns=None, remove=False)
- dict [scripts.cleaner.process_config](#) (config_file='../conf/cleaner.ini')

Variables

- [scripts.cleaner.script_dir](#) = os.path.dirname(os.path.realpath(__file__))
- [scripts.cleaner.project_dir](#) = os.path.dirname(script_dir)
- dict [scripts.cleaner.args](#) = process_config(os.path.join(project_dir, 'conf/cleaner.ini'))
- [scripts.cleaner.input_data](#) = os.path.join(project_dir, 'data/chem_output/chem_populated.csv')
- [scripts.cleaner.output_data](#) = os.path.join(project_dir, 'data/chem_output/phototox.csv')

8.7 scripts/crawlers/crawler_orchestrator.py File Reference

Namespaces

- [scripts.crawlers.crawler_orchestrator](#)

Functions

- def [scripts.crawlers.crawler_orchestrator.main](#) (file)

8.8 scripts/crawlers/dependencies/pubchem.py File Reference

Namespaces

- [scripts.crawlers.dependencies.pubchem](#)

Functions

- def [scripts.crawlers.dependencies.pubchem.crawl](#) (prefs)

8.9 scripts/crawlers/dependencies/swissadme.py File Reference

Namespaces

- [scripts.crawlers.dependencies.swissadme](#)

Functions

- def [scripts.crawlers.dependencies.swissadme.is_file_downloaded](#) (filename, timeout=5)
- def [scripts.crawlers.dependencies.swissadme.crawl](#) (prefs)

8.10 scripts/crawlers/dependencies/swisstarget.py File Reference

Namespaces

- [scripts.crawlers.dependencies.swisstarget](#)

Functions

- def [scripts.crawlers.dependencies.swisstarget.is_file_downloaded](#) (filename, timeout=5)
- def [scripts.crawlers.dependencies.swisstarget.crawl](#) (prefs)

8.11 scripts/merger.py File Reference

Namespaces

- [scripts.merger](#)

Functions

- def [scripts.merger.merge](#) ()

8.12 scripts/pipe_handler.py File Reference

Classes

- class [scripts.pipe_handler.IgnoreBrokenPipe](#)

Namespaces

- [scripts.pipe_handler](#)

8.13 scripts/populate_chem.py File Reference

Namespaces

- [scripts.populate_chem](#)

Functions

- def [scripts.populate_chem.populate](#) ()

Index

- `__init__`
 - `scripts.pipe_handler.IgnoreBrokenPipe`, 22
- `action`
 - `run_pipeline`, 11
- `args`
 - `run_pipeline`, 11
 - `scripts.cleaner`, 17
- `check_correlated_column`
 - `scripts.cleaner`, 13
- `check_outliers`
 - `scripts.cleaner`, 13
- `crawl`
 - `scripts.crawlers.dependencies.pubchem`, 18
 - `scripts.crawlers.dependencies.swissadme`, 19
 - `scripts.crawlers.dependencies.swisstarget`, 19
- `create_correlation_graphs`
 - `scripts.cleaner`, 13
- `create_mask`
 - `scripts.cleaner`, 14
- `flush`
 - `scripts.pipe_handler.IgnoreBrokenPipe`, 22
- `get_correlated_descriptors`
 - `scripts.cleaner`, 14
- `get_duplicit_correlated_descriptors`
 - `scripts.cleaner`, 14
- `help`
 - `run_pipeline`, 11
- `input_data`
 - `scripts.cleaner`, 17
- `is_file_downloaded`
 - `scripts.crawlers.dependencies.swissadme`, 19
 - `scripts.crawlers.dependencies.swisstarget`, 19
- `load_data`
 - `scripts.cleaner`, 15
- `main`
 - `scripts.cleaner`, 15
 - `scripts.crawlers.crawler_orchestrator`, 18
- `merge`
 - `scripts.merger`, 20
- `output_data`
 - `scripts.cleaner`, 17
- `parser`
 - `run_pipeline`, 11
- `plot_correlation_heatmap`
 - `scripts.cleaner`, 15
- `populate`
 - `scripts.populate_chem`, 20
- `process_config`
 - `scripts.cleaner`, 15
- `project_dir`
 - `scripts.cleaner`, 17
- `README.md`, 23
- `remove_duplicits`
 - `scripts.cleaner`, 16
- `remove_useless_columns`
 - `scripts.cleaner`, 16
- `run_pipeline`, 11
 - `action`, 11
 - `args`, 11
 - `help`, 11
 - `parser`, 11
 - `str`, 12
 - `type`, 12
- `run_pipeline.py`, 23
- `script_dir`
 - `scripts.cleaner`, 17
- `scripts`, 12
 - `scripts.cleaner`, 12
 - `args`, 17
 - `check_correlated_column`, 13
 - `check_outliers`, 13
 - `create_correlation_graphs`, 13
 - `create_mask`, 14
 - `get_correlated_descriptors`, 14
 - `get_duplicit_correlated_descriptors`, 14
 - `input_data`, 17
 - `load_data`, 15
 - `main`, 15
 - `output_data`, 17
 - `plot_correlation_heatmap`, 15
 - `process_config`, 15
 - `project_dir`, 17
 - `remove_duplicits`, 16
 - `remove_useless_columns`, 16
 - `script_dir`, 17
 - `set_proper_data_type`, 16
 - `unique_count`, 16
 - `scripts.crawlers`, 17
 - `scripts.crawlers.crawler_orchestrator`, 18
 - `main`, 18

- scripts.crawlers.dependencies, [18](#)
- scripts.crawlers.dependencies.pubchem, [18](#)
 - crawl, [18](#)
- scripts.crawlers.dependencies.swissadme, [18](#)
 - crawl, [19](#)
 - is_file_downloaded, [19](#)
- scripts.crawlers.dependencies.swisstarget, [19](#)
 - crawl, [19](#)
 - is_file_downloaded, [19](#)
- scripts.merger, [20](#)
 - merge, [20](#)
- scripts.pipe_handler, [20](#)
- scripts.pipe_handler.IgnoreBrokenPipe, [21](#)
 - __init__, [22](#)
 - flush, [22](#)
 - stream, [22](#)
 - write, [22](#)
- scripts.populate_chem, [20](#)
 - populate, [20](#)
- scripts/__init__.py, [23](#)
- scripts/cleaner.py, [24](#)
- scripts/crawlers/__init__.py, [23](#)
- scripts/crawlers/crawler_orchestrator.py, [24](#)
- scripts/crawlers/dependencies/__init__.py, [24](#)
- scripts/crawlers/dependencies/pubchem.py, [25](#)
- scripts/crawlers/dependencies/swissadme.py, [25](#)
- scripts/crawlers/dependencies/swisstarget.py, [25](#)
- scripts/merger.py, [25](#)
- scripts/pipe_handler.py, [26](#)
- scripts/populate_chem.py, [26](#)
- set_proper_data_type
 - scripts.cleaner, [16](#)
- str
 - run_pipeline, [12](#)
- stream
 - scripts.pipe_handler.IgnoreBrokenPipe, [22](#)
- type
 - run_pipeline, [12](#)
- unique_count
 - scripts.cleaner, [16](#)
- write
 - scripts.pipe_handler.IgnoreBrokenPipe, [22](#)