
Drug2Ways Documentation

Release 0.0.13-dev

Daniel Rivas Barragan and Daniel Domingo-Fernandez

Jun 13, 2022

CONTENTS:

| | | |
|----------|-------------------------------|-----------|
| 1 | Installation | 3 |
| 2 | Formats | 5 |
| 3 | Drug2ways API | 7 |
| 4 | Command Line Interface | 9 |
| 4.1 | drug2ways | 9 |
| 5 | Algorithm | 15 |
| 6 | Constants | 17 |
| 7 | Performance | 19 |
| 8 | Disclaimer | 21 |
| | Python Module Index | 23 |
| | Index | 25 |

Drug2ways supports generic network formats such as JSON, CSV, GraphML, or GML. Check out drug2ways's documentation here. Ideally, the network should contain three different types of nodes representing drugs, proteins, and indications/phenotypes. The hypothesis underlying this software is that by reasoning over a multitude of possible paths between a given drug and indication, the drug regulates the indication in the direction of the signs of the most frequently occurring paths (i.e., majority rule). In other words, we assume that a drug has a greater likelihood of interacting with its target, and its target with intermediate nodes, to modulate a pathological phenotype as the number of possible paths connecting a drug to the phenotype increases. Based on this hypothesis, this software can be applied for different applications outlined in the next section.

Installation is as easy as getting the code from [PyPI](#) with `python3 -m pip install drug2ways`. See the [installation](#) documentation.

See also:

- Documented on [Read the Docs](#)
- Versioned on [GitHub](#)
- Tested on [Travis CI](#)
- Distributed by [PyPI](#)

INSTALLATION

The latest stable code can be installed from [PyPI](#) with:

```
$ python3 -m pip install drug2ways
```

The most recent code can be installed from the source on [GitHub](#) with:

```
$ python3 -m pip install git+https://github.com/drug2ways/drug2ways.git
```

For developers, the repository can be cloned from [GitHub](#) and installed in editable mode with:

```
$ git clone https://github.com/drug2ways/drug2ways.git  
$ cd drug2ways  
$ python3 -m pip install -e .
```


FORMATS

Drug2ways handles the following formats:

- CSV (.csv)
- TSV (.tsv)
- GraphML (.graphml or .xml)
- GML (.gml or .xml)
- BEL (.bel)
- Pickle (.pickle). BELGraph object from [PyBEL 0.13.2](#)
- Edge list (.lst)
- Node-Link JSON (.json)

The GraphML, GML, EdgeList and Node-Link JSON readers of drug2ways rely on NetworkX. BEL and Pickle files rely on PyBEL. The two easiest format to work with drug2ways are a triple-based file represented as the file below (tsv/csv). The file must contain three columns: source, relation and target (order is not relevant) and the only condition is that the relation column contains 1 and -1 to indicate the direction of the relation (increase/decrease).

| source | target | relation |
|----------|----------|----------|
| Drug1 | Protein1 | -1 |
| Protein1 | Protein2 | 1 |
| Protein2 | Protein3 | -1 |
| Protein3 | Disease1 | 1 |

DRUG2WAYS API

Drug2ways implements an API with multiple methods that facilitate its programmatic usage. The majority of this methods are similar to the ones in the CLI. They are located in `cli_helper.py`.

```
from drug2ways.cli_helper import wrapper_explore
from networkx import DiGraph

# Initialize a directed graph
directed_graph = nx.DiGraph()
directed_graph.add_edges_from([(1, 2), (1, 3)])

results, time_cache = wrapper_explore(
    graph=directed_graph, # directed graph
    source_nodes=[1], # list of source nodes
    target_nodes=[3], # list of target nodes
    lmax=2, # max length of the path
    simple_paths=True, # with or without cycles (True=no cycles are allowed)
)
```


COMMAND LINE INTERFACE

Drug2Ways Command Line Interface

4.1 drug2ways

drug2ways

```
drug2ways [OPTIONS] COMMAND [ARGS] ...
```

4.1.1 combine

Run drug2ways for a given network and get sources that optimize given targets.

```
drug2ways combine [OPTIONS]
```

Options

- l, --log**
Activate debug mode
- o, --output <output>**
Output directory
- c, --combination-length <combination_length>**
Required Combination length. Number of drugs in each combination.
- a, --activation-threshold <activation_threshold>**
Activation threshold
- simple**
Count only simple paths, i.e. paths without cycles.
- l, --lmax <lmax>**
Required Maximum length of paths
- t, --targets <targets>**
Path to file with source nodes
- s, --sources <sources>**
Path to file with source nodes

-f, --fmt <fmt>

Graph fmt

Options

csv | tsv | graphml | bel | json | pickle

-g, --graph <graph>

Required Path to the network

4.1.2 explore

Run drug2ways for a given network.

```
drug2ways explore [OPTIONS]
```

Options

-t, --time

Export time measurements

-l, --log

Activate debug mode

-d, --drug-search-bel

Drug search for BEL graphs

-n, --name <name>

Name for output file

-o, --output <output>

Output directory

--simple

Count only simple paths, i.e. paths without cycles.

-l, --lmax <lmax>

Required Maximum length of paths

-t, --targets <targets>

Path to file with source nodes

-s, --sources <sources>

Path to file with source nodes

-f, --fmt <fmt>

Graph fmt

Options

csv | tsv | graphml | bel | json | pickle

-g, --graph <graph>

Required Path to the network

4.1.3 optimize

Run drug2ways for a given network and get sources that optimize given targets.

```
drug2ways optimize [OPTIONS]
```

Options

- l, --log**
Activate debug mode
- o, --output <output>**
Output directory
- a, --activation-threshold <activation_threshold>**
Activation threshold
- simple**
Count only simple paths, i.e. paths without cycles.
- l, --lmax <lmax>**
Required Maximum length of paths
- t, --targets <targets>**
Path to file with source nodes
- s, --sources <sources>**
Path to file with source nodes
- f, --fmt <fmt>**
Graph fmt
Options
csv | tsv | graphml | bel | json | pickle
- g, --graph <graph>**
Required Path to the network

4.1.4 pathway-analysis

Run drug2ways pathway enrichment on the paths.

```
drug2ways pathway-analysis [OPTIONS]
```

Options

- l, --log**
Activate debug mode
- o, --output <output>**
Output directory

--simple

Count only simple paths, i.e. paths without cycles.

-l, --lmax <lmax>

Required Maximum length of paths

-t, --targets <targets>

Path to file with source nodes

-s, --sources <sources>

Path to file with source nodes

-f, --fmt <fmt>

Graph fmt

Options

csv | tsv | graphml | bel | json | pickle

-g, --graph <graph>

Required Path to the network

4.1.5 validate-with-data

Run validation with RCR for a given network and experimental data.

```
drug2ways validate-with-data [OPTIONS]
```

Options

-l, --log

Activate debug mode

-n, --name <name>

Name for output file

-o, --output <output>

Output directory

--simple

Count only simple paths, i.e. paths without cycles.

-l, --lmax <lmax>

Required Maximum length of paths

-t, --target <target>

Name of the drug to be used as target

-t, --target <target>

Name of the drug to be used as source

-f, --fmt <fmt>

Graph fmt

Options

csv | tsv | graphml | bel | json | pickle

-g, --graph <graph>

Required Path to the network

ALGORITHM

Graph traversal methods.

```
drug2ways.graph_traversal.compute_all_paths_multitarget(graph: DiGraph, source: Iterable[Any],
                                                    targets, lmax: int, previous_history: Dict)
                                                    → Tuple[int, List[List[Union[float, int]]]]
```

Compute all paths to all the targets, separately.

Parameters

- **graph** – graph
- **source** – source nodes
- **targets** – target nodes
- **lmax** – lmax
- **previous_history** – previous history of visited nodes

Returns

```
drug2ways.graph_traversal.compute_all_paths_multitarget_dict(graph: DiGraph, source, targets,
                                                            lmax: int, previous_history: Dict,
                                                            cycle_history: Dict, simple_paths:
                                                            bool = False) → Tuple[int,
                                                            List[List[Union[float, int]]]]
```

Compute all paths to all the targets, separately. Uses dict to store target path count instead of array.

Parameters

- **graph** – graph
- **source** – source node
- **targets** – target nodes
- **lmax** – lmax
- **previous_history** – history of visited nodes
- **cycle_history** – history of cycles
- **simple_paths** – simple paths mode

Alternative methods for path calculations in graphs (depricated and not used in the package).

```
drug2ways.alternative_graph_traversal.enumerate_paths(graph, source, targets, lmax,
                                                    cycle_free=False)
```

Enumerate paths.

Parameters

- **graph** – graph
- **source** – source node
- **targets** – target nodes
- **lmax** – lmax
- **cycle_free** –

Returns

CONSTANTS

Constants of drug2ways.

`drug2ways.constants.DEFAULT_DRUG2WAYS_DIR = PosixPath('/home/docs/.data/drug2ways')`

Default drug2ways directory using PyStow

`drug2ways.constants.ensure_genesets()`

Download gene sets.

`drug2ways.constants.ensure_output_dirs()`

Ensure that the output directories exists.

`drug2ways.constants.download_pathway(url: str, export_path: str) → None`

Make a function that downloads the data for you, or uses a cached version at the given path.

Parameters

- **url** – The URL of some data
- **export_path** – folder where decompressed file will be exported

`drug2ways.constants.CSV = 'csv'`

csv

`drug2ways.constants.TSV = 'tsv'`

tsv

`drug2ways.constants.GRAPHML = 'graphml'`

graphML

`drug2ways.constants.BEL = 'bel'`

bel

`drug2ways.constants.NODE_LINK_JSON = 'json'`

node link json

`drug2ways.constants.BEL_PICKLE = 'pickle'`

pickle

`drug2ways.constants.GML = 'gml'`

gml

`drug2ways.constants.EDGE_LIST = '.lst'`

edge list

`drug2ways.constants.FORMATS = ['csv', 'tsv', 'graphml', 'bel', 'json', 'pickle']`

drug2ways available network formats

```
drug2ways.constants.FORMAT_SEPARATOR_MAPPING = {'csv': ',', 'tsv': '\t'}
```

Acceptable column names for the graph

```
drug2ways.constants.SOURCE = 'source'
```

Column name for source node

```
drug2ways.constants.TARGET = 'target'
```

Column name for target node

```
drug2ways.constants.RELATION = 'polarity'
```

Column name for relation

```
drug2ways.constants.EMOJI = ''
```

drug2ways emoji

PERFORMANCE

Drug2ways enables for parallelization to improve the performance. The only requirement is to install the MPI package (<https://mpi4py.readthedocs.io/>). Once the package is installed, drug2ways recognizes it automatically to calculate the predictions using parallelization.

DISCLAIMER

Drug2Ways is a scientific software that has been developed in an academic capacity, and thus comes with no warranty or guarantee of maintenance, support, or back-up of data.

PYTHON MODULE INDEX

d

`drug2ways.alternative_graph_traversal`, 15

`drug2ways.constants`, 17

`drug2ways.graph_traversal`, 15

Symbols

- activation-threshold
 - drug2ways-combine command line option, 9
 - drug2ways-optimize command line option, 11
- combination-length
 - drug2ways-combine command line option, 9
- drug-search-bel
 - drug2ways-explore command line option, 10
- fmt
 - drug2ways-combine command line option, 9
 - drug2ways-explore command line option, 10
 - drug2ways-optimize command line option, 11
 - drug2ways-pathway-analysis command line option, 12
 - drug2ways-validate-with-data command line option, 12
- graph
 - drug2ways-combine command line option, 10
 - drug2ways-explore command line option, 10
 - drug2ways-optimize command line option, 11
 - drug2ways-pathway-analysis command line option, 12
 - drug2ways-validate-with-data command line option, 12
- lmax
 - drug2ways-combine command line option, 9
 - drug2ways-explore command line option, 10
 - drug2ways-optimize command line option, 11
 - drug2ways-pathway-analysis command line option, 12
 - drug2ways-validate-with-data command line option, 12
- log
 - drug2ways-combine command line option, 9
 - drug2ways-explore command line option, 10
 - drug2ways-optimize command line option, 11
 - drug2ways-pathway-analysis command line option, 11
 - drug2ways-validate-with-data command line option, 12
- name
 - drug2ways-explore command line option, 10
 - drug2ways-validate-with-data command line option, 12
- output
 - drug2ways-combine command line option, 9
 - drug2ways-explore command line option, 10
 - drug2ways-optimize command line option, 11
 - drug2ways-pathway-analysis command line option, 11
 - drug2ways-validate-with-data command line option, 12
- simple
 - drug2ways-combine command line option, 9
 - drug2ways-explore command line option, 10
 - drug2ways-optimize command line option, 11
 - drug2ways-pathway-analysis command line option, 11
 - drug2ways-validate-with-data command line option, 12
- sources
 - drug2ways-combine command line option, 9
 - drug2ways-explore command line option, 10
 - drug2ways-optimize command line option, 11
 - drug2ways-pathway-analysis command line option, 12
- target
 - drug2ways-validate-with-data command line option, 12
- targets
 - drug2ways-combine command line option, 9
 - drug2ways-explore command line option, 10
 - drug2ways-optimize command line option, 11
 - drug2ways-pathway-analysis command line option, 12

--time
 drug2ways-explore command line option, 10

-a
 drug2ways-combine command line option, 9
 drug2ways-optimize command line option, 11

-c
 drug2ways-combine command line option, 9

-d
 drug2ways-explore command line option, 10

-f
 drug2ways-combine command line option, 9
 drug2ways-explore command line option, 10
 drug2ways-optimize command line option, 11
 drug2ways-pathway-analysis command line option, 12
 drug2ways-validate-with-data command line option, 12

-g
 drug2ways-combine command line option, 10
 drug2ways-explore command line option, 10
 drug2ways-optimize command line option, 11
 drug2ways-pathway-analysis command line option, 12
 drug2ways-validate-with-data command line option, 12

-l
 drug2ways-combine command line option, 9
 drug2ways-explore command line option, 10
 drug2ways-optimize command line option, 11
 drug2ways-pathway-analysis command line option, 11, 12
 drug2ways-validate-with-data command line option, 12

-n
 drug2ways-explore command line option, 10
 drug2ways-validate-with-data command line option, 12

-o
 drug2ways-combine command line option, 9
 drug2ways-explore command line option, 10
 drug2ways-optimize command line option, 11
 drug2ways-pathway-analysis command line option, 11
 drug2ways-validate-with-data command line option, 12

-s
 drug2ways-combine command line option, 9
 drug2ways-explore command line option, 10
 drug2ways-optimize command line option, 11
 drug2ways-pathway-analysis command line option, 12
 drug2ways-validate-with-data command line option, 12

-t
 drug2ways-combine command line option, 9
 drug2ways-explore command line option, 10
 drug2ways-optimize command line option, 11
 drug2ways-pathway-analysis command line option, 12
 drug2ways-validate-with-data command line option, 12

B

BEL (*in module drug2ways.constants*), 17
 BEL_PICKLE (*in module drug2ways.constants*), 17

C

compute_all_paths_multitarget() (*in module drug2ways.graph_traversal*), 15
 compute_all_paths_multitarget_dict() (*in module drug2ways.graph_traversal*), 15
 CSV (*in module drug2ways.constants*), 17

D

DEFAULT_DRUG2WAYS_DIR (*in module drug2ways.constants*), 17
 download_pathway() (*in module drug2ways.constants*), 17
 drug2ways.alternative_graph_traversal module, 15
 drug2ways.constants module, 17
 drug2ways.graph_traversal module, 15
 drug2ways-combine command line option
 --activation-threshold, 9
 --combination-length, 9
 --fmt, 9
 --graph, 10
 --lmax, 9
 --log, 9
 --output, 9
 --simple, 9
 --sources, 9
 --targets, 9
 -a, 9
 -c, 9
 -f, 9
 -g, 10
 -l, 9
 -o, 9
 -s, 9

-t, 9

drug2ways-explore command line option

- drug-search-bel, 10
- fmt, 10
- graph, 10
- lmax, 10
- log, 10
- name, 10
- output, 10
- simple, 10
- sources, 10
- targets, 10
- time, 10
- d, 10
- f, 10
- g, 10
- l, 10
- n, 10
- o, 10
- s, 10
- t, 10

drug2ways-optimize command line option

- activation-threshold, 11
- fmt, 11
- graph, 11
- lmax, 11
- log, 11
- output, 11
- simple, 11
- sources, 11
- targets, 11
- a, 11
- f, 11
- g, 11
- l, 11
- o, 11
- s, 11
- t, 11

drug2ways-pathway-analysis command line option

- fmt, 12
- graph, 12
- lmax, 12
- log, 11
- output, 11
- simple, 11
- sources, 12
- targets, 12
- f, 12
- g, 12
- l, 11, 12
- o, 11
- s, 12
- t, 12

drug2ways-validate-with-data command line option

- fmt, 12
- graph, 12
- lmax, 12
- log, 12
- name, 12
- output, 12
- simple, 12
- target, 12
- f, 12
- g, 12
- l, 12
- n, 12
- o, 12
- t, 12

E

EDGE_LIST (in module *drug2ways.constants*), 17

EMOJI (in module *drug2ways.constants*), 18

ensure_genesets() (in module *drug2ways.constants*), 17

ensure_output_dirs() (in module *drug2ways.constants*), 17

enumerate_paths() (in module *drug2ways.alternative_graph_traversal*), 15

F

FORMAT_SEPARATOR_MAPPING (in module *drug2ways.constants*), 17

FORMATS (in module *drug2ways.constants*), 17

G

GML (in module *drug2ways.constants*), 17

GRAPHML (in module *drug2ways.constants*), 17

M

module

drug2ways.alternative_graph_traversal, 15

drug2ways.constants, 17

drug2ways.graph_traversal, 15

N

NODE_LINK_JSON (in module *drug2ways.constants*), 17

R

RELATION (in module *drug2ways.constants*), 18

S

SOURCE (in module *drug2ways.constants*), 18

T

TARGET (*in module drug2ways.constants*), 18

TSV (*in module drug2ways.constants*), 17