
data_tools Documentation

Release 0.0.2

Nicolas Palacio

May 25, 2018

CONTENTS

1	Dependencies	3
2	Installation	5
3	Reference	7
3.1	data_tools.plots	7
3.2	data_tools.sets	9
3.3	data_tools.strings	11
	Python Module Index	13
	Index	15

Collection of Python functions and classes designed to make a Computational Biologist's life easier.

Copyright (C) 2018 Nicolás Palacio

Contact: nicolaspalacio91@gmail.com

GNU-GLPv3: This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

A full copy of the GNU General Public License can be found on file [LICENSE.md](#). If not, see <http://www.gnu.org/licenses/>.

DEPENDENCIES

In module **data_tools.plots**:

- NumPy
- Matplotlib
- Pandas

INSTALLATION

First download/clone `data_tools` from the [GitHub repository](https://github.com/Nic-Nic/data_tools). From the terminal:

```
git clone https://github.com/Nic-Nic/data_tools.git
cd data_tools
```

Then you can install it by running `setup.py` as follows:

```
python setup.py sdist
```

Or using `pip`:

```
pip install .
```


REFERENCE

3.1 data_tools.plots

Plotting functions module.

`data_tools.plots.volcano(logfc, logpval, thr_pval=0.05, thr_fc=2.0, c=('C0', 'C1'), legend=True, title=None, filename=None, figsize=None)`

Generates a volcano plot from the differential expression data provided.

- **Arguments:**

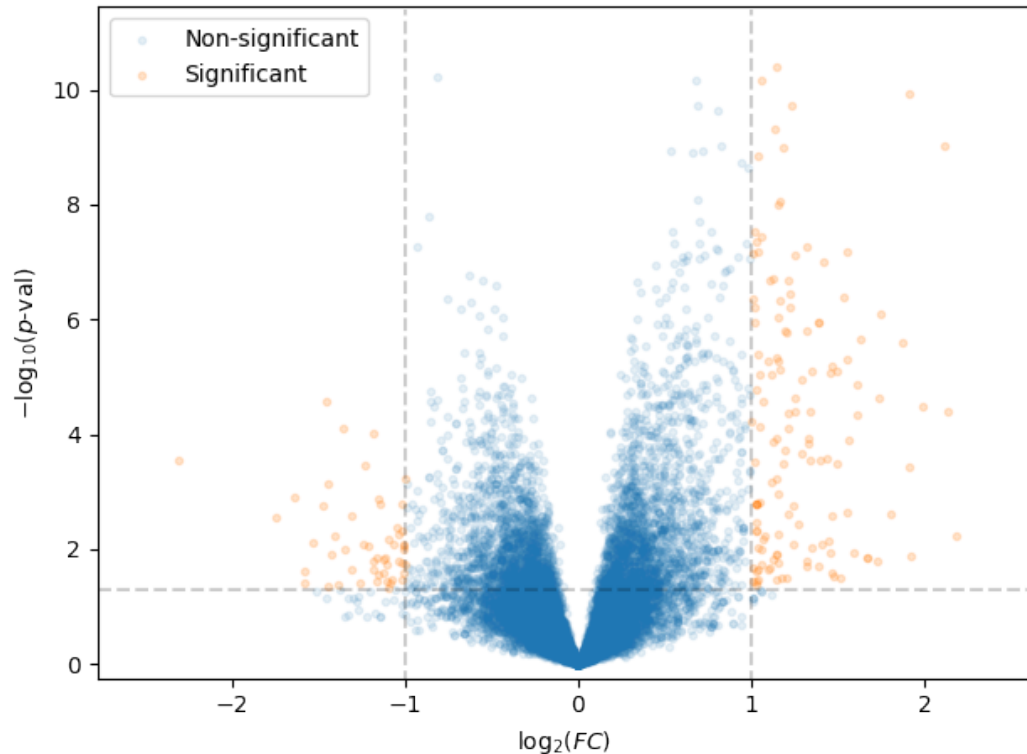
- *logfc* [list]: Or any iterable type. Contains the log (usually base 2) fold-change values. Must have the same length as *logpval*.
- *logpval* [list]: Or any iterable type. Contains the -log p-values (usually base 10). Must have the same length as *logfc*.
- *thr_pval* [float]: Optional, 0.05 by default. Specifies the p-value (non log-transformed) threshold to consider a measurement as significantly differentially expressed.
- *thr_fc* [float]: Optional, 2. by default. Specifies the FC (non log-transformed) threshold to consider a measurement as significantly differentially expressed.
- *c* [tuple]: Optional, ('C0', 'C1') by default (matplotlib default colors). Any iterable containing two color arguments tolerated by matplotlib (e.g.: ['r', 'b'] for red and blue). First one is used for non-significant points, second for the significant ones.
- *legend* [bool]: Optional, True by default. Indicates whether to show the plot legend or not.
- *title* [str]: Optional, None by default. Defines the plot title.
- *filename* [str]: Optional, None by default. If passed, indicates the file name or path where to store the figure. Format must be specified (e.g.: .png, .pdf, etc)
- *figsize* [tuple]: Optional, None by default (default matplotlib size). Any iterable containing two values denoting the figure size (in inches) as [width, height].

- **Returns:**

- [matplotlib.figure.Figure]: Figure object containing the volcano plot.

- **Examples:**

```
>>> volcano(my_log_fc, my_log_pval)
```



`data_tools.plots.piano_consensus` (*df*, *nchar*=40, *boxes*=True, *title*=None, *filename*=None, *figsize*=None)

Generates a GSEA consensus score plot like R package `piano`'s `consensusScores` function, but prettier. The main input is assumed to be a `pandas.DataFrame` whose data is the same as the `rankMat` from the result of `consensusScores`.

- **Arguments:**

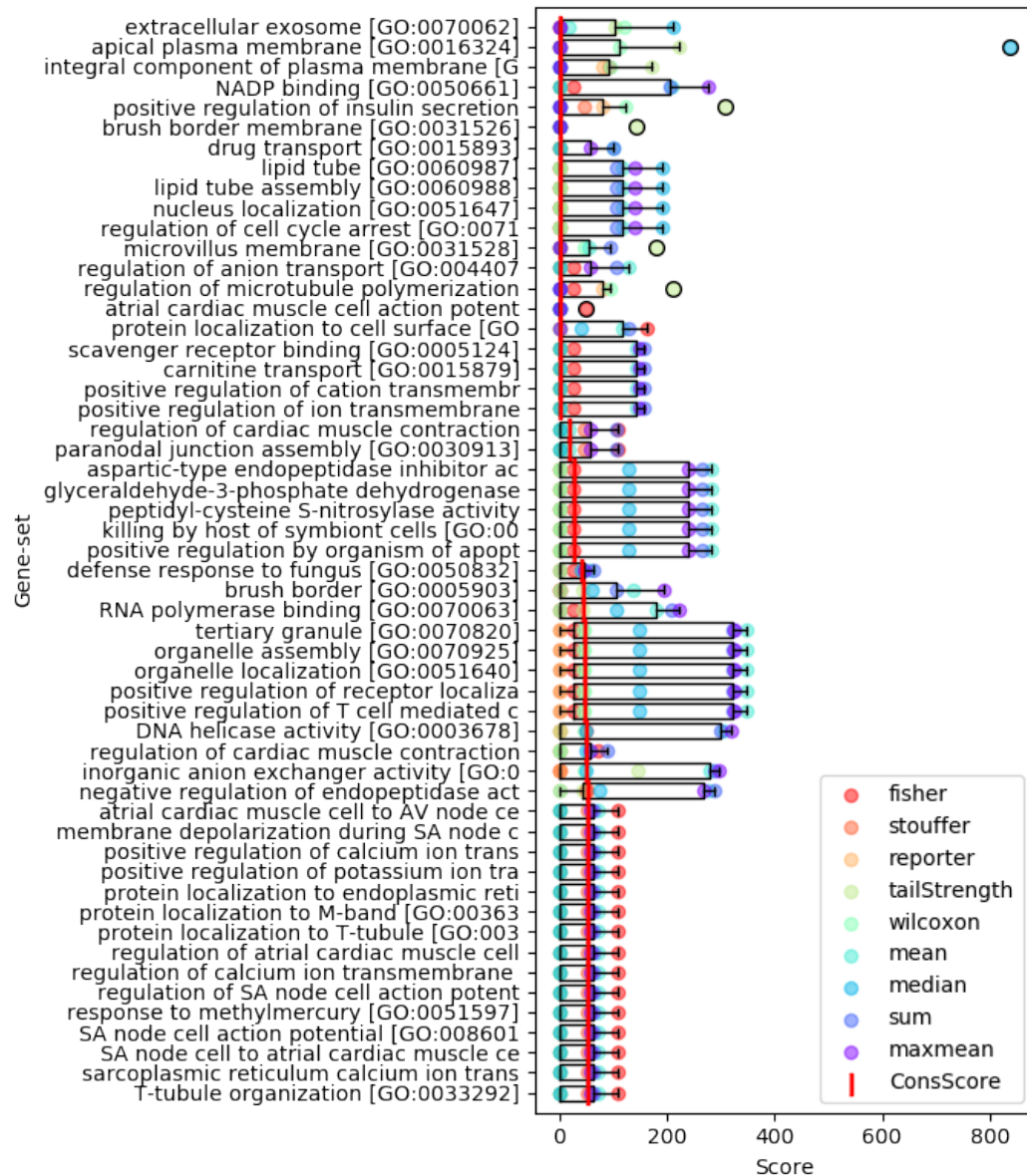
- *df* [`pandas.DataFrame`]: Values contained correspond to the scores of the gene-sets (consensus and each individual methods). Index must contain the gene-set labels. Columns are assumed to be `ConsRank` (ignored), `ConsScore` followed by the individual methods (e.g.: `mean`, `median`, `sum`, etc).
- *nchar* [`int`]: Optional, 40 by default. Number of string characters of the gene-set labels of the plot.
- *boxes* [`bool`]: Optional, `True` by default. Determines whether to show the boxplots of the gene-sets or not.
- *title* [`str`]: Optional, `None` by default. Defines the plot title.
- *filename* [`str`]: Optional, `None` by default. If passed, indicates the file name or path where to store the figure. Format must be specified (e.g.: `.png`, `.pdf`, etc)
- *figsize* [`tuple`]: Optional, `None` by default (default `matplotlib` size). Any iterable containing two values denoting the figure size (in inches) as [`width`, `height`].

- **Returns:**

- [`matplotlib.figure.Figure`]: the figure object containing a combination of box and scatter plots of the gene-set scores.

- **Examples:**

```
>>> piano_consensus(df, figsize=[7, 8])
```



3.2 data_tools.sets

Set operations module.

`data_tools.sets.in_all(x, N)`

Checks if a vector x is present in all sets contained in a list N .

- **Arguments:**

- x [tuple]: Or any hashable type as long as is the same contained in the sets of N .
- N [list]: Or any iterable type containing [set] objects.

- **Returns:**
 - [bool]: True if x is found in all sets of N , False otherwise.
- **Examples:**

```
>>> N = [{(0, 0), (0, 1)}, # <- set A
...      {(0, 0), (1, 1), (1, 0)}] # <- set B
>>> x = (0, 0)
>>> in_all(x, N)
True
>>> y = (0, 1)
>>> in_all(y, N)
False
```

`data_tools.sets.bit_or(a, b)`

Returns the bit operation OR between two bit-strings a and b . NOTE: a and b must have the same size.

- **Arguments:**
 - a [tuple]: Or any iterable type.
 - b [tuple]: Or any iterable type.
- **Returns:**
 - [tuple]: OR operation between a and b element-wise.
- **Examples:**

```
>>> a, b = (0, 0, 1), (1, 0, 1)
>>> bit_or(a, b)
(1, 0, 1)
```

`data_tools.sets.multi_union(N)`

Returns the union set of all sets contained in a list N .

- **Arguments:**
 - N [list]: Or any iterable type containing [set] objects.
- **Returns:**
 - [set]: The union of all sets contained in N .
- **Examples:**

```
>>> A = {1, 3, 5}
>>> B = {0, 1, 2}
>>> C = {0, 2, 5}
>>> multi_union([A, B, C])
set([0, 1, 2, 3, 5])
```

`data_tools.sets.find_min(A)`

Finds and returns the subset of vectors whose sum is minimum from a given set A .

- **Arguments:**
 - A [set]: Set of vectors ([tuple] or any iterable).
- **Returns:**
 - [set]: Subset of vectors in A whose sum is minimum.

- **Examples:**

```
>>> A = {(0, 1, 1), (0, 1, 0), (1, 0, 0), (1, 1, 1)}
>>> find_min(A)
set([(0, 1, 0), (1, 0, 0)])
```

3.3 data_tools.strings

String operations module.

`data_tools.strings.is_numeric(s)`

Determines if a string can be considered a numeric value. NaN is also considered, since it is float type.

- **Arguments:**

- `s [str]`: String to be evaluated.

- **Returns:**

- `[bool]`: True/False depending if the condition is satisfied.

- **Examples:**

```
>>> is_numeric('4')
True
>>> is_numeric('-3.2')
True
>>> is_numeric('number')
False
>>> is_numeric('NaN')
True
```

`data_tools.strings.join_str_lists(a, b, sep="")`

Joins element-wise two lists (or any 1D iterable) of strings with a given separator (if provided). Length of the input lists must be equal.

- **Arguments:**

- `a [list]`: Contains the first elements [str] of the joint strings.

- `b [list]`: Contains the second elements [str] of the joint strings.

- `sep [str]`: Optional ' ' (non separated) by default. Determines the separator between the joint strings.

- **Returns:**

- `[list]`: List of the joint strings.

- **Example:**

```
>>> a = ['a', 'b']
>>> b = ['1', '2']
>>> join_str_lists(a, b, sep='_')
['a_1', 'b_2']
```


PYTHON MODULE INDEX

d

`data_tools.plots`, [7](#)
`data_tools.sets`, [9](#)
`data_tools.strings`, [11](#)

INDEX

B

`bit_or()` (in module `data_tools.sets`), 10

D

`data_tools.plots` (module), 7

`data_tools.sets` (module), 9

`data_tools.strings` (module), 11

F

`find_min()` (in module `data_tools.sets`), 10

I

`in_all()` (in module `data_tools.sets`), 9

`is_numeric()` (in module `data_tools.strings`), 11

J

`join_str_lists()` (in module `data_tools.strings`), 11

M

`multi_union()` (in module `data_tools.sets`), 10

P

`piano_consensus()` (in module `data_tools.plots`), 8

V

`volcano()` (in module `data_tools.plots`), 7