
PyBEL-Jupyter Documentation

Release 0.3.0

Charles Tapley Hoyt

Apr 16, 2020

CONTENTS

1	Installation	3
2	Getting Started	5
3	Indices and tables	7
	Python Module Index	9
	Index	11

A PyBEL extension for Jupyter notebooks.

INSTALLATION

`pybel_jupyter` can be installed easily from [PyPI](#) with the following code in your favorite terminal:

```
$ pip install pybel-jupyter
```

or from the latest code on [GitHub](#) with:

```
$ pip install git+https://github.com/pybel/pybel-jupyter.git
```


GETTING STARTED

Inside a Jupyter notebook, run the following code at the end of the cell to get an interactive visualization:

```
>>> from pybel.examples import sialic_acid_graph
>>> from pybel_jupyter import to_jupyter
>>> to_jupyter(sialic_acid_graph)
```

`pybel_jupyter.to_html` (*graph*, *color_map=None*)

Create an HTML visualization for the given JSON representation of a BEL graph.

Parameters

- **graph** (BELGraph) – A BEL graph
- **color_map** (Optional[Mapping[str, str]]) – A dictionary from PyBEL internal node functions to CSS color strings like #FFEE00. Defaults to `default_color_map`

Return type `str`

Returns HTML string representing the graph

`pybel_jupyter.to_html_file` (*graph*, *file*, *color_map=None*)

Write the HTML visualization to a file or file-like.

Parameters

- **graph** (BELGraph) – A BEL graph
- **color_map** (Optional[Mapping[str, str]]) – A dictionary from PyBEL internal node functions to CSS color strings like #FFEE00. Defaults to `default_color_map`
- **file** (*file*) – A writable file or file-like or file path

Return type `None`

`pybel_jupyter.to_jupyter` (*graph*, *width=1000*, *height=650*, *color_map=None*)

Display a BEL graph inline in a Jupyter notebook.

To use successfully, make run as the last statement in a cell inside a Jupyter notebook.

Parameters

- **graph** (BELGraph) – A BEL graph
- **width** (*int*) – The width of the visualization window to render
- **height** (*int*) – The height of the visualization window to render
- **color_map** (Optional[Mapping[str, str]]) – A dictionary from PyBEL internal node functions to CSS color strings like #FFEE00. Defaults to `default_color_map`

Returns An IPython notebook Javascript object

Return type `IPython.display.Javascript`

`pybel_jupyter.to_jupyter_str(graph, width=1000, height=650, color_map=None)`

Return the string to be javascript-ified by the Jupyter notebook function `IPython.display.Javascript`.

Parameters

- **graph** (`BELGraph`) – A BEL graph
- **width** (`int`) – The width of the visualization window to render
- **height** (`int`) – The height of the visualization window to render
- **color_map** (`Optional[Mapping[str, str]]`) – A dictionary from PyBEL internal node functions to CSS color strings like `#FFEE00`. Defaults to `default_color_map`

Return type `str`

Returns The javascript string to turn into magic

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

p

pybel_jupyter, ??

INDEX

M

module
 pybel_jupyter, 1

P

pybel_jupyter
 module, 1

T

to_html() *(in module pybel_jupyter)*, 5
to_html_file() *(in module pybel_jupyter)*, 5
to_jupyter() *(in module pybel_jupyter)*, 5
to_jupyter_str() *(in module pybel_jupyter)*, 6