
pyFlapjack

Release 1.0.0

RudgeLab

Nov 16, 2022

CONTENTS

1	pyFlapjack	1
1.1	Instalation	1
1.2	Tutorials	1
2	API Reference	3
2.1	flapjack	3
3	pyFlapjack	9
3.1	Instalation	9
3.2	Tutorials	9
	Python Module Index	11
	Index	13

PYFLAPJACK

pyFlapjack is our Python package that allows you to interface the [Flapjack](#) with Pandas and Numpy stack, so that you can easily implement it in your projects.

1.1 Instalation

Installing pyFlapjack is quite simple. Please refer to our Wiki for installation instructions, available here: [Installation](#).

1.2 Tutorials

Now that you have pyFlapjack installed you can familiarize yourself with the tool using the [Jupyter notebook tutorials](#) designed for this purpose.

1.2.1 Related repositories

The repository for Flapjack API is available here: [Flapjack API](#); and the one for the Flapjack Frontend is available here: [Flapjack Frontend](#).

1.2.2 Reference this paper

Please reference our Flapjack's paper—available [here](#), using the following reference: > Guillermo Yáñez Feliú, Benjamín Earle Gómez, Verner Codoceo Berrocal, Macarena Muñoz Silva, Isaac N. Nuñez, Tamara F. Matute, Anibal Arce Medina, Gonzalo Vidal, Carlos Vidal Céspedes, Jonathan Dahlin, Fernán Federici, and Timothy J. Rudge *ACS Synthetic Biology* **2021 10** (1), 183-191 DOI: 10.1021/acssynbio.0c00554

API REFERENCE

This page contains auto-generated API reference documentation¹.

2.1 flapjack

2.1.1 Submodules

`flapjack.flapjack`

Module Contents

Classes

Flapjack

Attributes

index_params

plot_option_keys

replace_columns_with_ids

```
class Flapjack(url_base='localhost:8000')
    models = ['study', 'assay', 'sample', 'strain', 'media', 'vector', 'dna', 'signal',
              'chemical', ...
    __del__(self)
    async _analysis(self, **kwargs)
    async _measurements(self, **kwargs)
    async _plot(self, **kwargs)
```

¹ Created with `sphinx-autoapi`

```
async _upload_measurements(self, df, **kwargs)
analysis(self, **kwargs)
create(self, model, confirm=True, overwrite=False, **kwargs)
delete(self, model, id, confirm=True)
get(self, model, **kwargs)
handle_response(self, s)
log_in(self, username, password)
log_out(self)
measurements(self, **kwargs)
parse_params(self, **kwargs)
patch(self, model, id, **kwargs)
plot(self, **kwargs)
refresh(self)
upload_measurements(self, df, **kwargs)
index_params = ['biomass_signal', 'ref_signal', 'analyte', 'analyte1', 'analyte2']
plot_option_keys = ['normalize', 'subplots', 'markers', 'plot']
replace_columns_with_ids = []
```

flapjack.simulator

Module Contents

Classes

Simulator

Attributes

colors

```
class Simulator(study_name="", assay_name="", study_description="", assay_description="", dna_name="",
                init_proteins=[0], concentrations=[0], n_signals=1, fluo_noise=0.01, od_noise=0.01)
    create_data(self, fj, step, n_samples, nt, dt, sim_steps)
    create_meta_objects(self, fj)
colors = ['red', 'green', 'blue']
```


flapjack.util**Module Contents****Functions**

exponential_growth(t, y0, k)

exponential_growth_rate(t, y0, k)

fit_curve(func, data, x, y, **kwargs)

gompertz(t, y0, ymax, um, l)

gompertz_growth_rate(t, y0, ymax, um, l)

hill(x, a, b, k, n)

layout_print(fig, width=3.3, height=1.5, Layout figure optimized for print at 300dpi
font_size=6)

exponential_growth(t, y0, k)**exponential_growth_rate**(t, y0, k)**fit_curve**(func, data, x, y, **kwargs)**gompertz**(t, y0, ymax, um, l)**gompertz_growth_rate**(t, y0, ymax, um, l)**hill**(x, a, b, k, n)**layout_print**(fig, width=3.3, height=1.5, font_size=6)

Layout figure optimized for print at 300dpi

fig = figure to layout width,height = size in inches font_size = font size in pts

Returns: fig = figure with correct layout

2.1.2 Package Contents**Classes**

Flapjack

Simulator

Functions

exponential_growth(t, y0, k)

exponential_growth_rate(t, y0, k)

fit_curve(func, data, x, y, **kwargs)

gompertz(t, y0, ymax, um, l)

gompertz_growth_rate(t, y0, ymax, um, l)

hill(x, a, b, k, n)

layout_print(fig, width=3.3, height=1.5, Layout figure optimized for print at 300dpi
font_size=6)

Attributes

colors

index_params

plot_option_keys

replace_columns_with_ids

```
class Flapjack(url_base='localhost:8000')
    models = ['study', 'assay', 'sample', 'strain', 'media', 'vector', 'dna', 'signal',
              'chemical', ...
    __del__(self)
    async _analysis(self, **kwargs)
    async _measurements(self, **kwargs)
    async _plot(self, **kwargs)
    async _upload_measurements(self, df, **kwargs)
    analysis(self, **kwargs)
    create(self, model, confirm=True, overwrite=False, **kwargs)
    delete(self, model, id, confirm=True)
    get(self, model, **kwargs)
    handle_response(self, s)
    log_in(self, username, password)
    log_out(self)
    measurements(self, **kwargs)
```

```

    parse_params(self, **kwargs)
    patch(self, model, id, **kwargs)
    plot(self, **kwargs)
    refresh(self)
    upload_measurements(self, df, **kwargs)

class Simulator(study_name="", assay_name="", study_description="", assay_description="", dna_name="",
                init_proteins=[0], concentrations=[0], n_signals=1, fluo_noise=0.01, od_noise=0.01)
    create_data(self, fj, step, n_samples, nt, dt, sim_steps)
    create_meta_objects(self, fj)

colors = ['red', 'green', 'blue']
exponential_growth(t, y0, k)
exponential_growth_rate(t, y0, k)
fit_curve(func, data, x, y, **kwargs)
gompertz(t, y0, ymax, um, l)
gompertz_growth_rate(t, y0, ymax, um, l)
hill(x, a, b, k, n)
index_params = ['biomass_signal', 'ref_signal', 'analyte', 'analyte1', 'analyte2']
layout_print(fig, width=3.3, height=1.5, font_size=6)
    Layout figure optimized for print at 300dpi
    fig = figure to layout width,height = size in inches font_size = font size in pts
    Returns: fig = figure with correct layout
plot_option_keys = ['normalize', 'subplots', 'markers', 'plot']
replace_columns_with_ids = []

```


PYFLAPJACK

pyFlapjack is our Python package that allows you to interface the [Flapjack](#) with Pandas and Numpy stack, so that you can easily implement it in your projects.

3.1 Instalation

Installing pyFlapjack is quite simple. Please refer to our Wiki for installation instructions, available here: [Installation](#).

3.2 Tutorials

Now that you have pyFlapjack installed you can familiarize yourself with the tool using the [Jupyter notebook tutorials](#) designed for this purpose.

3.2.1 Related repositories

The repository for Flapjack API is available here: [Flapjack API](#); and the one for the Flapjack Frontend is available here: [Flapjack Frontend](#).

3.2.2 Reference this paper

Please reference our Flapjack's paper—available [here](#), using the following reference: > Guillermo Yáñez Feliú, Benjamín Earle Gómez, Verner Codoceo Berrocal, Macarena Muñoz Silva, Isaac N. Nuñez, Tamara F. Matute, Anibal Arce Medina, Gonzalo Vidal, Carlos Vidal Céspedes, Jonathan Dahlin, Fernán Federici, and Timothy J. Rudge *ACS Synthetic Biology* **2021** **10** (1), 183-191 DOI: 10.1021/acssynbio.0c00554

PYTHON MODULE INDEX

f

`flapjack`, [3](#)
`flapjack.flapjack`, [3](#)
`flapjack.simulator`, [4](#)
`flapjack.util`, [5](#)

Symbols

`__del__()` (*Flapjack method*), 3, 6
`_analysis()` (*Flapjack method*), 3, 6
`_measurements()` (*Flapjack method*), 3, 6
`_plot()` (*Flapjack method*), 3, 6
`_upload_measurements()` (*Flapjack method*), 3, 6

A

`analysis()` (*Flapjack method*), 4, 6

C

`colors` (*in module flapjack*), 7
`colors` (*in module flapjack.simulator*), 4
`create()` (*Flapjack method*), 4, 6
`create_data()` (*Simulator method*), 4, 7
`create_meta_objects()` (*Simulator method*), 4, 7

D

`delete()` (*Flapjack method*), 4, 6

E

`exponential_growth()` (*in module flapjack*), 7
`exponential_growth()` (*in module flapjack.util*), 5
`exponential_growth_rate()` (*in module flapjack*), 7
`exponential_growth_rate()` (*in module flapjack.util*), 5

F

`fit_curve()` (*in module flapjack*), 7
`fit_curve()` (*in module flapjack.util*), 5
`flapjack`
 module, 3
`Flapjack` (*class in flapjack*), 6
`Flapjack` (*class in flapjack.flapjack*), 3
`flapjack.flapjack`
 module, 3
`flapjack.simulator`
 module, 4
`flapjack.util`
 module, 5

G

`get()` (*Flapjack method*), 4, 6
`gompertz()` (*in module flapjack*), 7
`gompertz()` (*in module flapjack.util*), 5
`gompertz_growth_rate()` (*in module flapjack*), 7
`gompertz_growth_rate()` (*in module flapjack.util*), 5

H

`handle_response()` (*Flapjack method*), 4, 6
`hill()` (*in module flapjack*), 7
`hill()` (*in module flapjack.util*), 5

I

`index_params` (*in module flapjack*), 7
`index_params` (*in module flapjack.flapjack*), 4

L

`layout_print()` (*in module flapjack*), 7
`layout_print()` (*in module flapjack.util*), 5
`log_in()` (*Flapjack method*), 4, 6
`log_out()` (*Flapjack method*), 4, 6

M

`measurements()` (*Flapjack method*), 4, 6
`models` (*Flapjack attribute*), 3, 6
module
 flapjack, 3
 flapjack.flapjack, 3
 flapjack.simulator, 4
 flapjack.util, 5

P

`parse_params()` (*Flapjack method*), 4, 6
`patch()` (*Flapjack method*), 4, 7
`plot()` (*Flapjack method*), 4, 7
`plot_option_keys` (*in module flapjack*), 7
`plot_option_keys` (*in module flapjack.flapjack*), 4

R

`refresh()` (*Flapjack method*), 4, 7
`replace_columns_with_ids` (*in module flapjack*), 7

`replace_columns_with_ids` (*in module `flapjack.flapjack`*), [4](#)

S

`Simulator` (*class in `flapjack`*), [7](#)

`Simulator` (*class in `flapjack.simulator`*), [4](#)

U

`upload_measurements()` (*Flapjack method*), [4](#), [7](#)