
test_autodoc Documentation

Release 1.0.0

Nicolas Fernandez

February 17, 2017

1	new_module	3
1.1	new_module package	3
2	New Module API	5
3	Local Clustergrammer API	7
4	Indices and tables	9
	Python Module Index	11

Contents:

new_module

new_module package

Module contents

class new_module.**Customer** (*name*, *balance=-1000.0*)

Bases: object

I am modifying the module to update the starting balance

A customer of ABC Bank with a checking account. Customers have the following properties:

Attributes: *name*: A string representing the customer's name. *balance*: A float tracking the current balance of the customer's account.

deposit (*amount*)

Return the balance remaining after depositing *amount* dollars.

withdraw (*amount*)

Return the balance remaining after withdrawing *amount* dollars.

New Module API

I am able to get autodocs to build the module API in a new page too

```
class new_module.Customer (name, balance=-1000.0)
```

Bases: `object`

I am modifying the module to update the starting balance

A customer of ABC Bank with a checking account. Customers have the following properties:

Attributes: `name`: A string representing the customer's name. `balance`: A float tracking the current balance of the customer's account.

deposit (*amount*)

Return the balance remaining after depositing *amount* dollars.

withdraw (*amount*)

Return the balance remaining after withdrawing *amount* dollars.

Local Clustergrammer API

local new module

class clustergrammer_py.**Network**

Bases: object

version 1.2.1 Modifying API

Networks have two states:

- 1.the data state, where they are stored as a matrix and nodes
- 2) the viz state where they are stored as viz.links, viz.row_nodes, and viz.col_nodes.

The goal is to start in a data-state and produce a viz-state of the network that will be used as input to clustergram.js.

Iframe_web_app (*filename=None, width=1000, height=800*)

dat_to_df ()

convert from clusergrammers dat format to pandas dataframe

df_to_dat (*df*)

Convert from pandas dataframe to clustergrammers dat format (will be deprecated)

enrichr (*req_type, gene_list=None, lib=None, list_id=None, max_terms=None*)

under development, get enrichment results from Enrichr and add them to clustergram

export_df ()

export dataframe from network

export_net_json (*net_type='viz', indent='no-indent'*)

export dat or viz json

filter_N_top (*inst_rc, N_top, rank_type='sum'*)

Filter a network's rows or cols based on sum/variance, and only keep the top N

filter_sum (*inst_rc, threshold, take_abs=True*)

Filter a network's rows or columns based on the sum across rows or columns Works on the network object

filter_threshold (*inst_rc, threshold, num_occur=1*)

Filter a network's rows or cols based on num_occur values being above a threshold (in absolute value)

load_data_file_to_net (*filename*)

load my .dat format (saved as json) for a network to a netowrk

load_df (*df*)

Upload pandas datafraeme

load_file (*filename*)
load file to network, currently supporting only tsv

static load_gmt (*filename*)

static load_json_to_dict (*filename*)

load_stdin ()
load stdin tsv formatted string

load_tsv_to_net (*file_buffer*, *filename=None*)
This will load a tsv matrix file buffer, this is exposed so that it will be possible to load data without having to read from a file.

load_vect_post_to_net (*vect_post*)
load vector format to network

make_clust (*dist_type='cosine', run_clustering=True, dendro=True, views=['N_row_sum', 'N_row_var'], linkage_type='average', sim_mat=False, filter_sim=0.1, calc_cat_pval=False, run_enrichr=None*)
The main function run by the user to make their clustergram. views is later referred to as requested_views.

normalize (*df=None, norm_type='zscore', axis='row', keep_orig=False*)
under development, normalize the network rows/cols using zscore

produce_view (*requested_view=None*)
under development, will produce a single view on demand from .dat data

reset ()
function for user to reset network

static save_dict_to_json (*inst_dict, filename, indent='no-indent'*)

swap_nan_for_zero ()

widget ()
export viz json, for use with clustergrammer_widget

write_json_to_file (*net_type, filename, indent='no-indent'*)

write_matrix_to_tsv (*filename=None, df=None*)

Indices and tables

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

c

clustergrammer_py, [7](#)

n

new_module, [5](#)

C

clustergrammer_py (module), 7
Customer (class in new_module), 3, 5

D

dat_to_df() (clustergrammer_py.Network method), 7
deposit() (new_module.Customer method), 3, 5
df_to_dat() (clustergrammer_py.Network method), 7

E

enrichr() (clustergrammer_py.Network method), 7
export_df() (clustergrammer_py.Network method), 7
export_net_json() (clustergrammer_py.Network method), 7

F

filter_N_top() (clustergrammer_py.Network method), 7
filter_sum() (clustergrammer_py.Network method), 7
filter_threshold() (clustergrammer_py.Network method), 7

I

Iframe_web_app() (clustergrammer_py.Network method), 7

L

load_data_file_to_net() (clustergrammer_py.Network method), 7
load_df() (clustergrammer_py.Network method), 7
load_file() (clustergrammer_py.Network method), 7
load_gmt() (clustergrammer_py.Network static method), 8
load_json_to_dict() (clustergrammer_py.Network static method), 8
load_stdin() (clustergrammer_py.Network method), 8
load_tsv_to_net() (clustergrammer_py.Network method), 8
load_vect_post_to_net() (clustergrammer_py.Network method), 8

M

make_clust() (clustergrammer_py.Network method), 8

N

Network (class in clustergrammer_py), 7
new_module (module), 3, 5
normalize() (clustergrammer_py.Network method), 8

P

produce_view() (clustergrammer_py.Network method), 8

R

reset() (clustergrammer_py.Network method), 8

S

save_dict_to_json() (clustergrammer_py.Network static method), 8
swap_nan_for_zero() (clustergrammer_py.Network method), 8

W

widget() (clustergrammer_py.Network method), 8
withdraw() (new_module.Customer method), 3, 5
write_json_to_file() (clustergrammer_py.Network method), 8
write_matrix_to_tsv() (clustergrammer_py.Network method), 8