box_counting

ボックスカウンティング法によってクラスターのフラクタル次元の時間変化を求める。
最終的にはフラクタル次元の時間変化がtanh(t)に従うことを示したい。
~~シミュレーション時に用意する格子サイズは，2の階乗にし，分割が行い易くなるようにする。~~
格子サイズは，フィッティング時のサンプル数を確保するために，上のようなことを考える必要はない

Modules

argparse numpy matplotlib.pyplot save_data

Classes

__builtin__.object

BoxCounting

class **BoxCounting**(__builtin__.object)

Methods defined here:

__init__(self, frames, beta, frames_list, N_L, save_fitting=False, save_fitting_dir='')

calc_fractal_dim(self, main, i, s)

diecutting_one_cluster(self, width, height, x0, y0)

get_cutting_sizes(self, N_L)

```
    Create the cutting size list for simulation

    self.X0: x coordinates of the left buttom corner
    self.Y0: y coordinates of the left buttom corner
    self.cutting_size_max_width: max width of the cutting size
    self.cutting_size_max_height: max height of the cutting size
    self.cutting_size_xs: cutting size list
    self.cutting_size_ys: cutting size list
    self.cutting_sizes: ndarray [[cutting_size_xs[0], cutting_size_ys[0]],
                                 [cutting_size_xs[1], cutting_size_ys[1]],
                                     ...
                                ]

    In this funciton, cutting size is determined by which the whole region
    is in the cluster.
```

get_results_each_subclusters(self)

start(self)

Data descriptors defined here:

__dict__

```
    dictionary for instance variables (if defined)
```

__weakref__

```
    list of weak references to the object (if defined)
```

Functions

box_count(beta, frames_list, N_L=20, num_of_strings=100)

Data

__warningregistry__ = {("Not importing directory './diecutting': missing __init__.py", <type 'exceptions.ImportWarning'>, 12): True}