

```
# -*- coding:utf-8 -*-
#
# written by Shotaro Fujimoto
# 2016-08-15
```

Modules

[numpy](#)

[matplotlib.pyplot](#)

Classes

[growing\\_string.Main\(base.Main\)](#)

[Count\\_in\\_r](#)

class **Count\_in\_r**([growing\\_string.Main](#))

Method resolution order:

[Count\\_in\\_r](#)

[growing\\_string.Main](#)

[base.Main](#)

Methods defined here:

**`__init__`**(self)

Methods inherited from [growing\\_string.Main](#):

**`calc_weight`**(self, s, i, r\_i=None, r\_rev=None)  
ベクトルの内積を元に、Boltzmann分布に従って成長点選択の重みを決定

**`cleanup_bonding_pairs`**(self, key, index\_start, index\_stop)

**`dot`**(self, v, w)  
0〜5で表された6つのベクトルの内積を計算する。  
  
v, w (int): ベクトル(0〜5の整数で表す)

**`get_bonding_pairs`**(self, s, index\_start, index\_stop)

**`get_neighbor_xy`**(self, key)  
Stringクラスのインスタンスsの隣接する非占有格子点の座標を取得する  
  
s (String): 対象とするStringクラスのインスタンス

**`plot_all`**(self)  
軸の設定、三角格子の描画、線分描画要素の用意などを行う  
  
ここからFuncAnimationを使ってアニメーション表示を行うようにする

**`plot_string`**(self)  
self.strings内に格納されているStringを参照し、グラフ上に図示する

**`start_animation`**(self, filename='')

**`update`**(self, num=0)  
FuncAnimationから各フレームごとに呼び出される関数  
  
1時間ステップの間に行う計算はすべてここに含まれる。

**`update_each_string`**(self, key)

Methods inherited from [base.Main](#):

**`create_random_strings`**(self, N=3, size=[10, 5, 3])  
Create N strings of each size is specified with 'size'.  
  
This process is equivalent to self-avoiding walk on triangular lattice.

**`get_next_xy`**(self, x, y, \*vec)

Functions

```
count_point_in_r(self, s, N_r, r=None)
main()
```