PhD Diary Week Beginning 19th November

Nathan Hughes

November 23, 2018

Contents

1 Lab work

- Spent 2 days in lab with Jeroen
  - Harvested leaf tissue
  - Did PCR
  - Ran gels on DNA to check for homozygous

2 Ideas

2.1 TODO Jeroen suggested using 2D diffusion model to work out potential diffusion of chitin

- This would allow us to know where to make our "no-mans-zone" for sampling before RNA-seq
- Possible reference here (Chalykh et al., 2014) for the diffusion constants of chitin

2.1.1 TODO Find out diffusion constant of chitin

3 General Maths/Programming

This week I have been working on some more maths in relation to programming. In particular better utilising the numpy library to support faster operations. An online resource: "From Python to Numpy" (Rougier, 2016) has been most enlightening.

3.1 DONE Generators

https://www.geeksforgeeks.org/use-yield-keyword-instead-return-keyword-python/

3.2 DONE Python Decorators

3.3 Timing/Profiling functions

```
def timeit(method):
    import time
    def timed(*args, **kw):
        ts = time.time()
        result = method(*args, **kw)
        te = time.time()
        return (result, (te-ts))

    return timed
```

3.4 Vectorising with Numpy

3.4.1 OOP Approach

```
import matplotlib.pyplot as plt
from random import randint
import seaborn as sns
sns.set()
%matplotlib inline


class RandomWalker:
    def __init__(self):
        self.position = 0

    def walk(self, n):
        self.position = 0
        for i in range(n):
            yield self.position
            self.position += 2*randint(0, 1)-1


@timeit
def make_walkers(x,y):
    return [[p for p in RandomWalker().walk(y)] for _ in range(x) ]

N = 10000
res, time = make_walkers(N,N)
for walker in res:
    plt.plot(walker)
plt.suptitle('Time taken: {0:.2f}'.format(time))
```

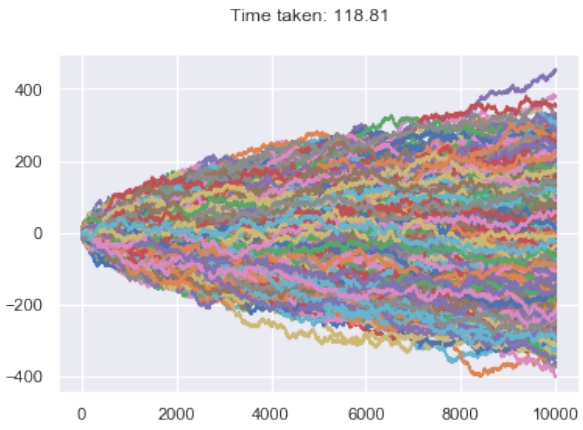

Figure 1: Random Walkers

3.4.2 Vectorised

```
from itertools import accumulate
from random import choices
import matplotlib.pyplot as plt
%matplotlib inline



def random_walk_faster(n=1000):
    steps = choices([-1, +1], k=n)
    return [0]+list(accumulate(steps))

@timeit
def make_walkers(x, y):
    return [[p for p in random_walk_faster(x)] for _ in range(y)]


res, time = make_walkers(N,N)
for walker in res:
    plt.plot(walker)
plt.suptitle('Time taken: {0:.2f}'.format(time))
```

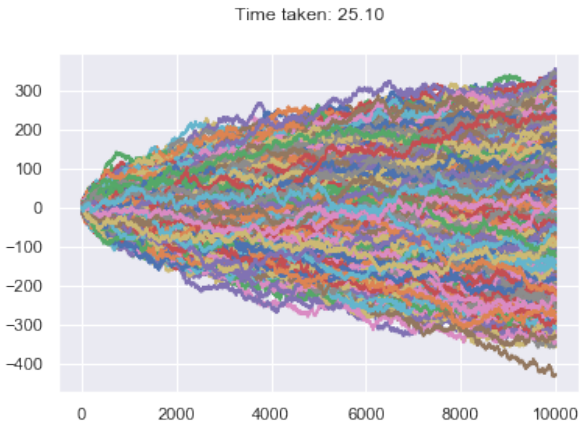

Figure 2: Random Walkers (vectorised)

3.4.3 Vectorised with numpy

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline


def random_walk_fastest(n=1000):
    steps = np.random.choice([-1, 1], n)
    return np.cumsum(steps)

@timeit
def make_walkers(x, y):
    return [[p for p in random_walk_fastest(x)] for _ in range(y)]

res, time = make_walkers(N,N)

for walker in res:
      plt.plot(walker)
plt.suptitle('Time taken: {0:.2f}'.format(time))
```

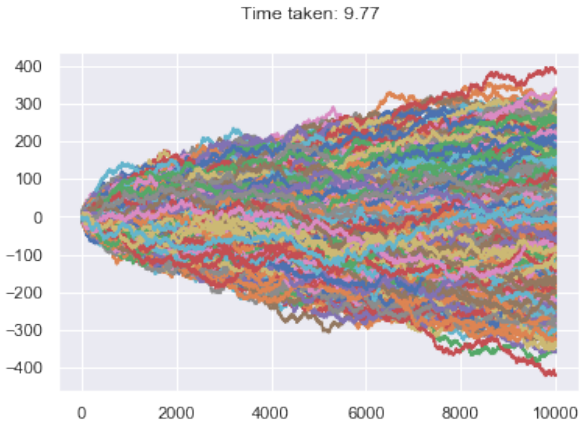

Figure 3: Random Walkers (numpy vectorised)

References

A. E. Chalykh, T. F. Petrova, R. R. Khasbiullin, and A. N. Ozerin. Water sorption on and water diffusion in chitin and chitosan. *Polymer Science Series A*, 56(5):614–622, September 2014. ISSN 1555-6107. doi: 10.1134/S0965545X14050034.

Nicolas P. Rougier. *Rougier/from-Python-to-Numpy: Version 1.1*. Zenodo, December 2016. doi: 10.5281/zenodo.225783.