Network**L**

a Python package for the **L**ongitudinal analysis of
Large-scale time-varying graphs

Moreno Bonaventura

@morenobonav

m.bonaventura@qmul.ac.uk

13th PyData Meetup, 7th July 2015, London

NetworkL

a Python package for the Longitudinal analysis of
Large-scale time-varying graphs

Startup-Network.org

Queen Mary
University of London

Network**L**

a Python package for the **L**ongitudinal analysis of
Large-scale time-varying graphs

Startup-Network.org

Age	3 months
Lines of code	341
Developers	1+1/2

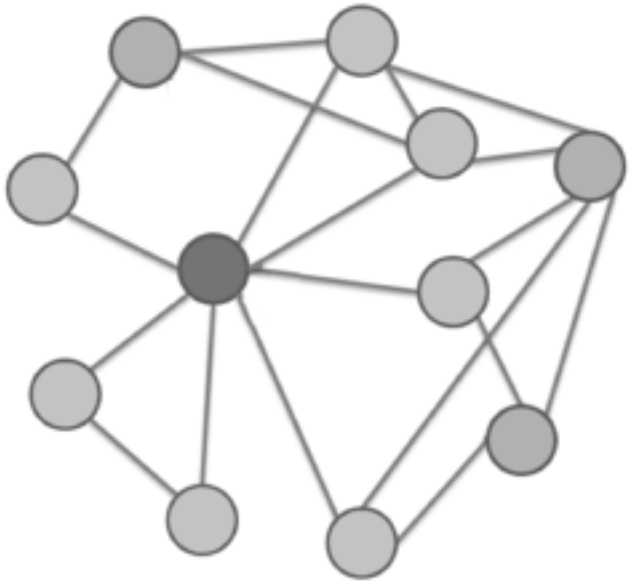
Queen Mary
University of London

Performances:
-saving up to 50% memory
-computation in centiseconds

STARTUP NETWORK

Queen Mary
University of London

NetworkL



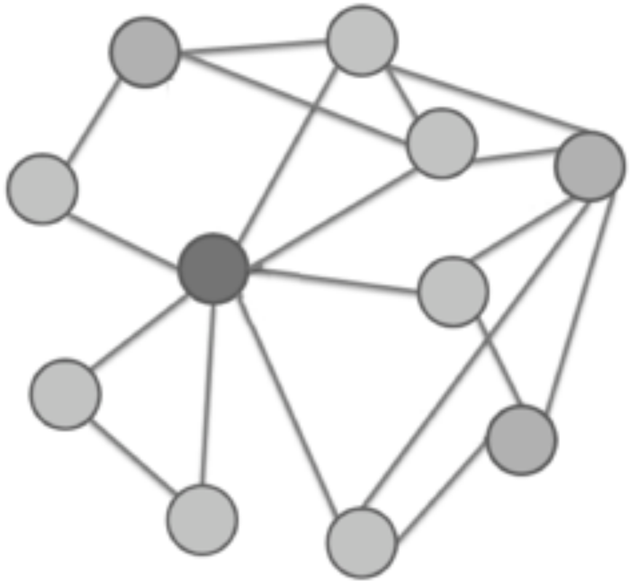
1°
2°
3°
...

The most famous rank?

STARTUP NETWORK

Queen Mary
University of London

NetworkL



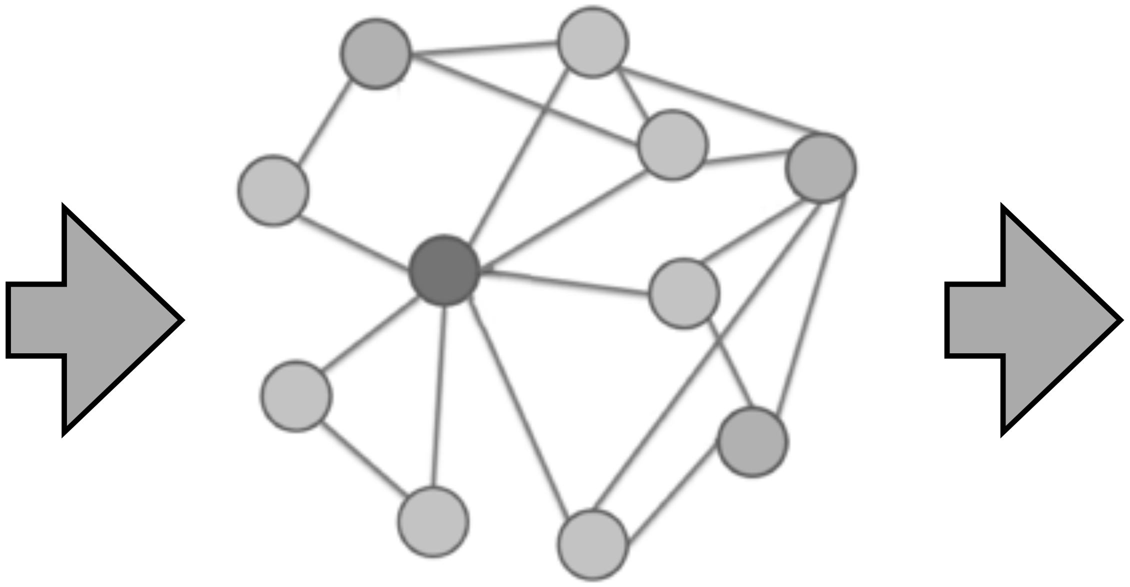
1°
2°
3°
…

Google PageRank!

STARTUP NETWORK

Queen Mary
University of London

NetworkL



1°
2°
3°
…

Google
PageRank

Python Libraries:
SNAP
NetworkX
Graph-Tool
iGraph

Network**Longitudinal**

12am

1°
2°
...

13am

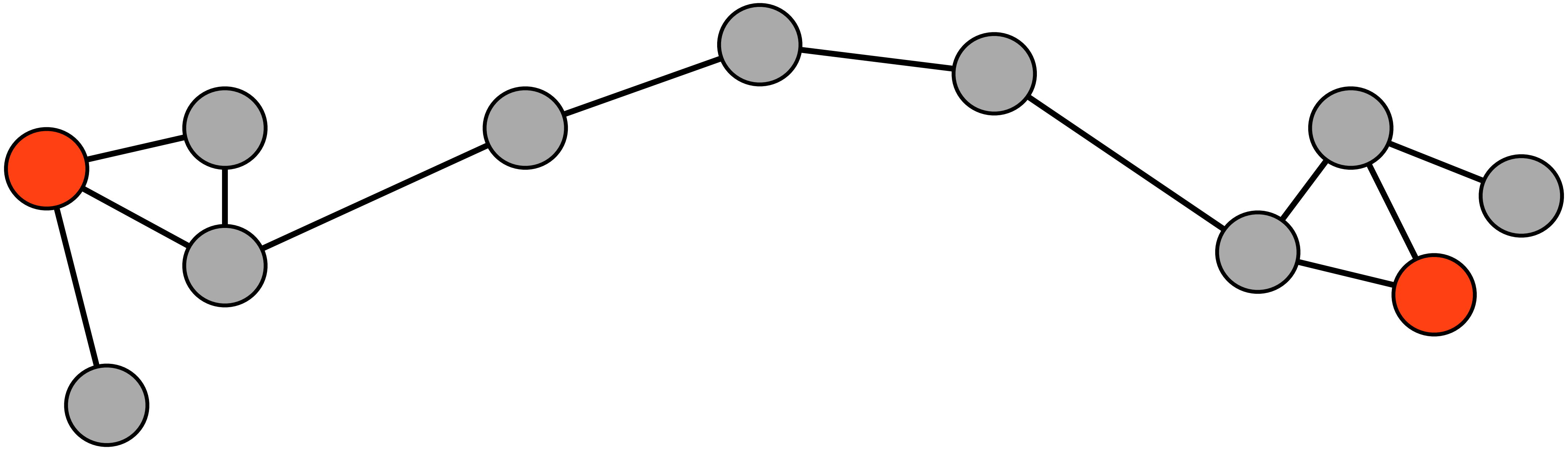
1°
2°
...

14am

1°
2°
...

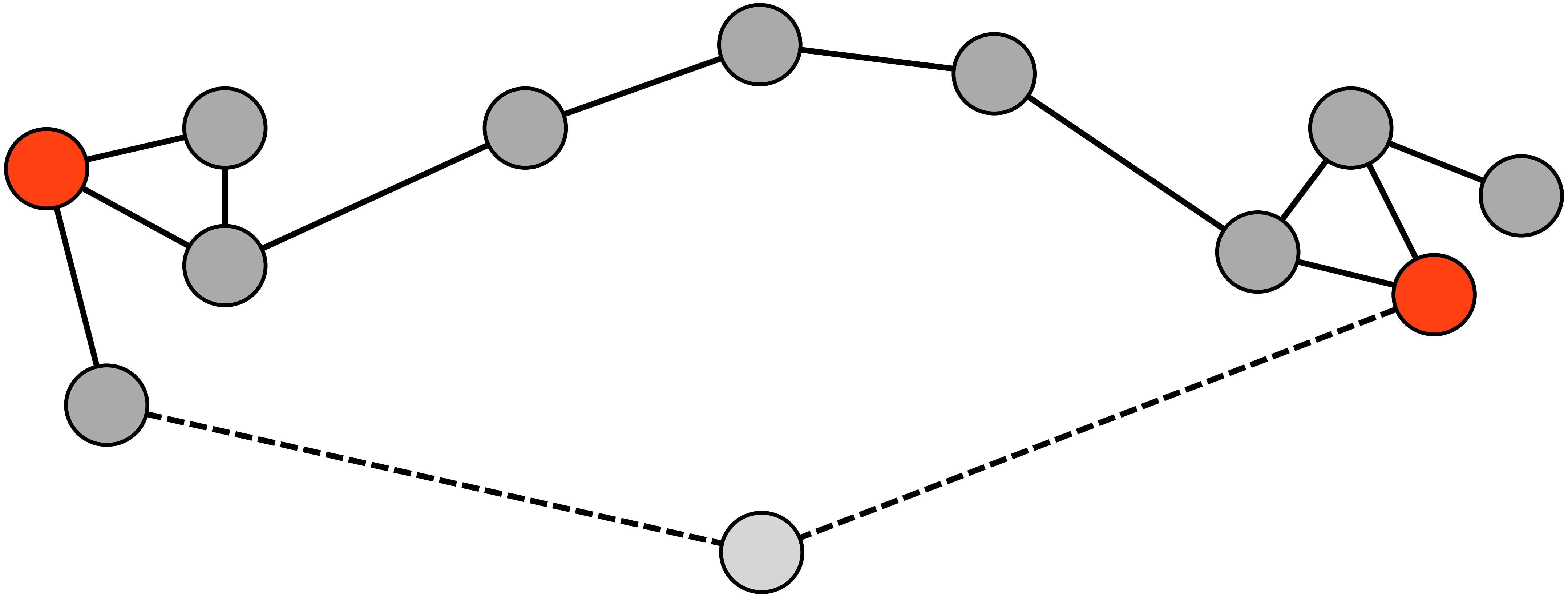
NetworkLongitudinal

Distance: 6 hops



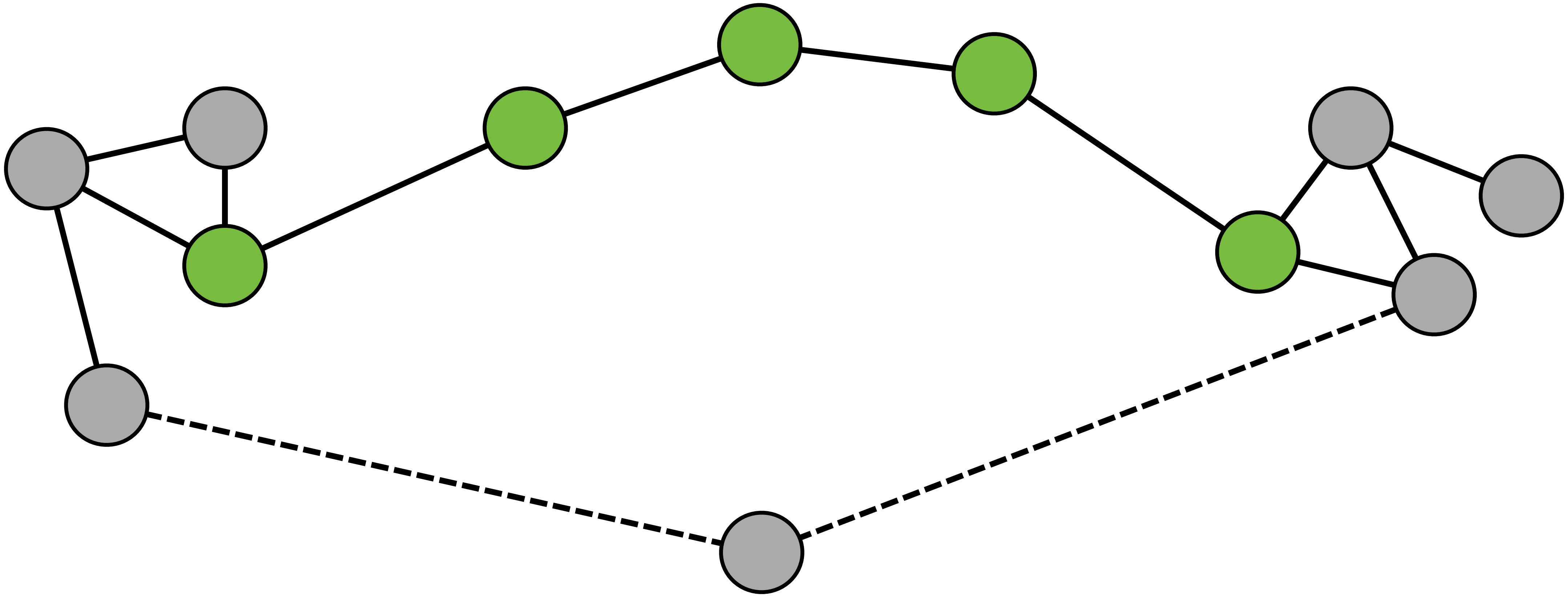
Network**Longitudinal**

Distance: 3 hops



NetworkLongitudinal

Unaffected distances



Ramalingam, G., & Reps, T. (1996). On the computational complexity of dynamic graph problems. Theoretical Computer Science, 158(1), 233-277.

Network**Large graph**

many many distances....

Sparse Geodesic Matrix...............up to 50% memory saving

Sparse Biconnected Geodesic Matrix.............up to 75%

NetworkLet’s try

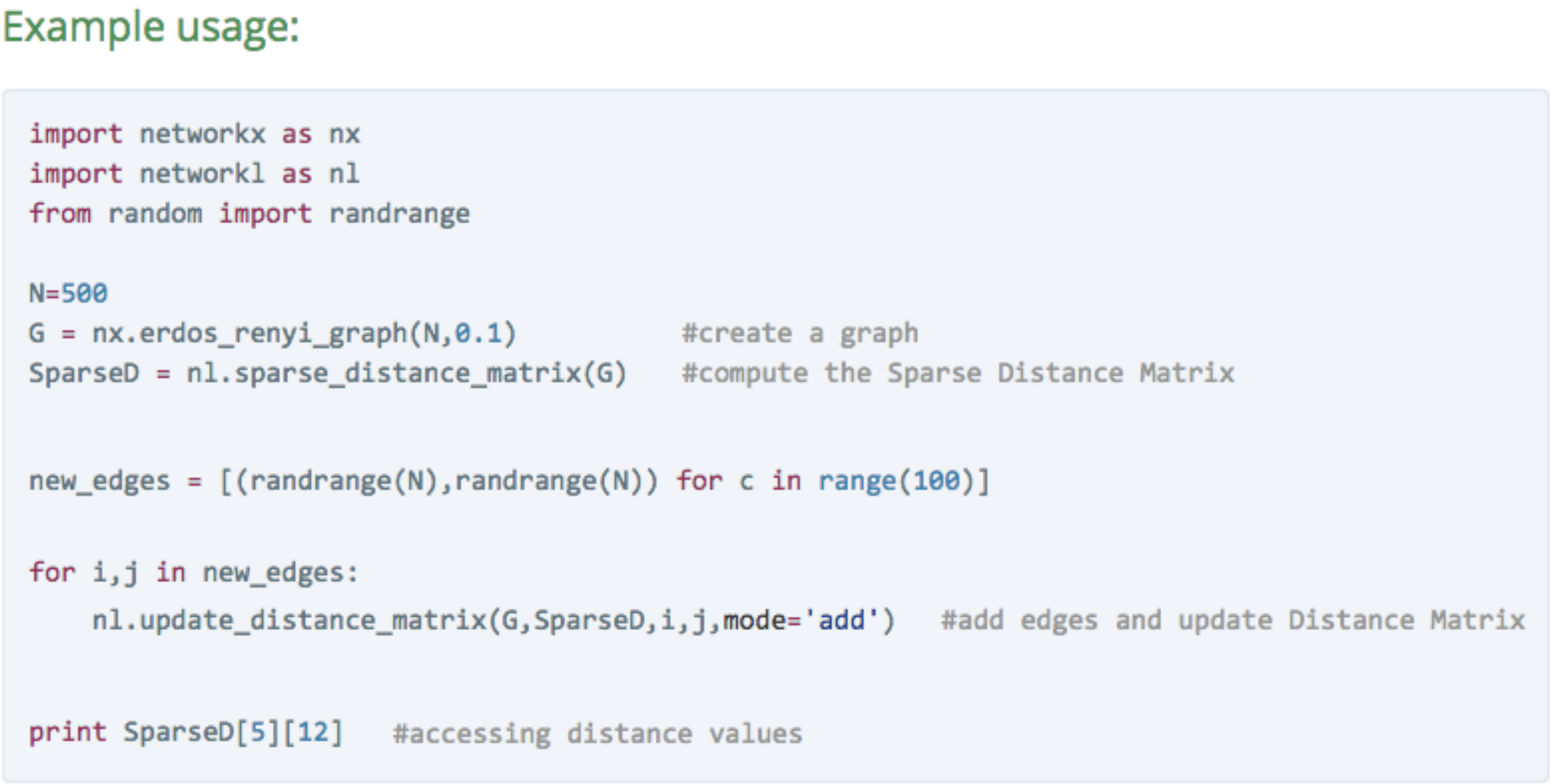
```
import networkl as nl

SparseD = nl.sparse_distance_matrix(G)

i = 5
j = 7

nl.update_distance_matrix(G, SparseD, i, j, mode=‘add’)
```

NetworkLet’s try



Example usage:

```
import networkx as nx
import networkl as nl
from random import randrange

N=500
G = nx.erdos_renyi_graph(N,0.1)           #create a graph
SparseD = nl.sparse_distance_matrix(G)    #compute the Sparse Distance Matrix


new_edges = [(randrange(N),randrange(N)) for c in range(100)]


for i,j in new_edges:
    nl.update_distance_matrix(G,SparseD,i,j,mode='add')   #add edges and update Distance Matrix


print SparseD[5][12]   #accessing distance values
```

GitHub

http://networkl.github.io

Startup Network

LAB.startup-network.org

Queen Mary
University of London

Moreno Bonaventura

@morenobonav

m.bonaventura@qmul.ac.uk

www.maths.qmul.ac.uk/~mbonaventura/