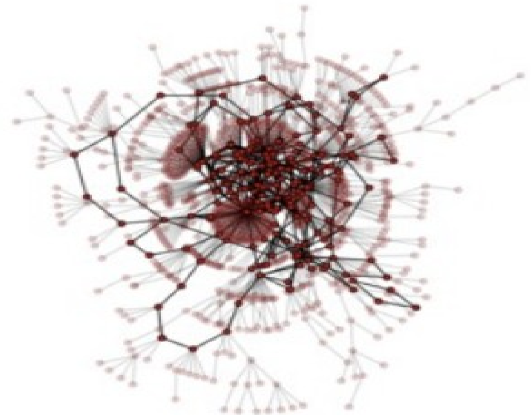
NetworkX: a Python module (and related modules)

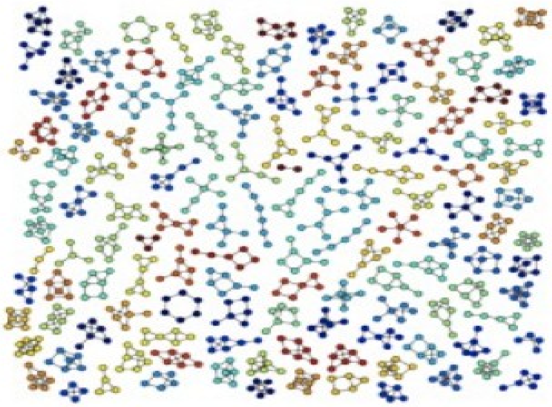
Inspired by the tutorial of Salvatore Scellato for the course “Social and Technological Network Analysis”, University of Cambridge (2011)

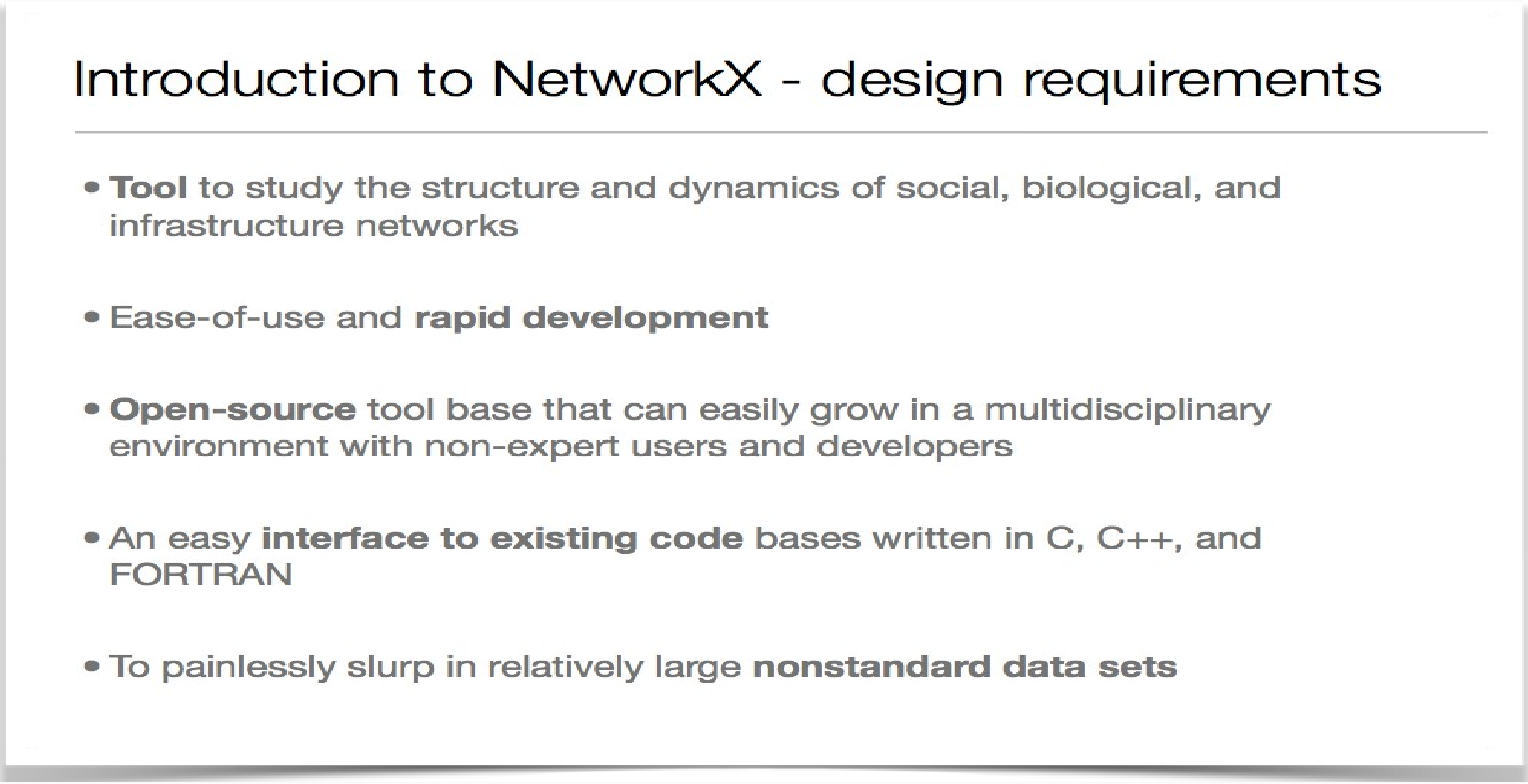
Introduction to NetworkX

> ***"Python package for the creation, manipulation and study of the structure, dynamics and functions of complex networks."***

- Data structures for representing many types of networks, or graphs
- Nodes can be any (hashable) Python object, edges can contain arbitrary data
- Flexibility ideal for representing networks found in many different fields
- Easy to install on multiple platforms
- Online up-to-date documentation
- First public release in April 2005

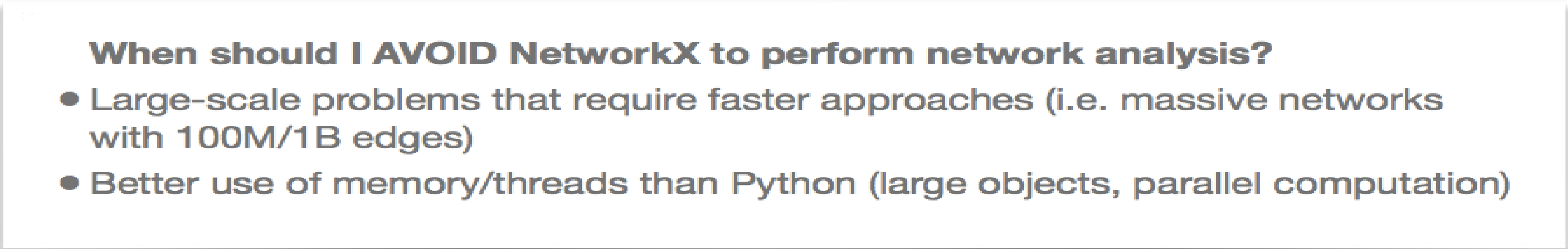






Introduction to NetworkX - design requirements

- **Tool** to study the structure and dynamics of social, biological, and infrastructure networks
- Ease-of-use and **rapid development**
- **Open-source** tool base that can easily grow in a multidisciplinary environment with non-expert users and developers
- An easy **interface to existing code** bases written in C, C++, and FORTRAN
- To painlessly slurp in relatively large **nonstandard data sets**

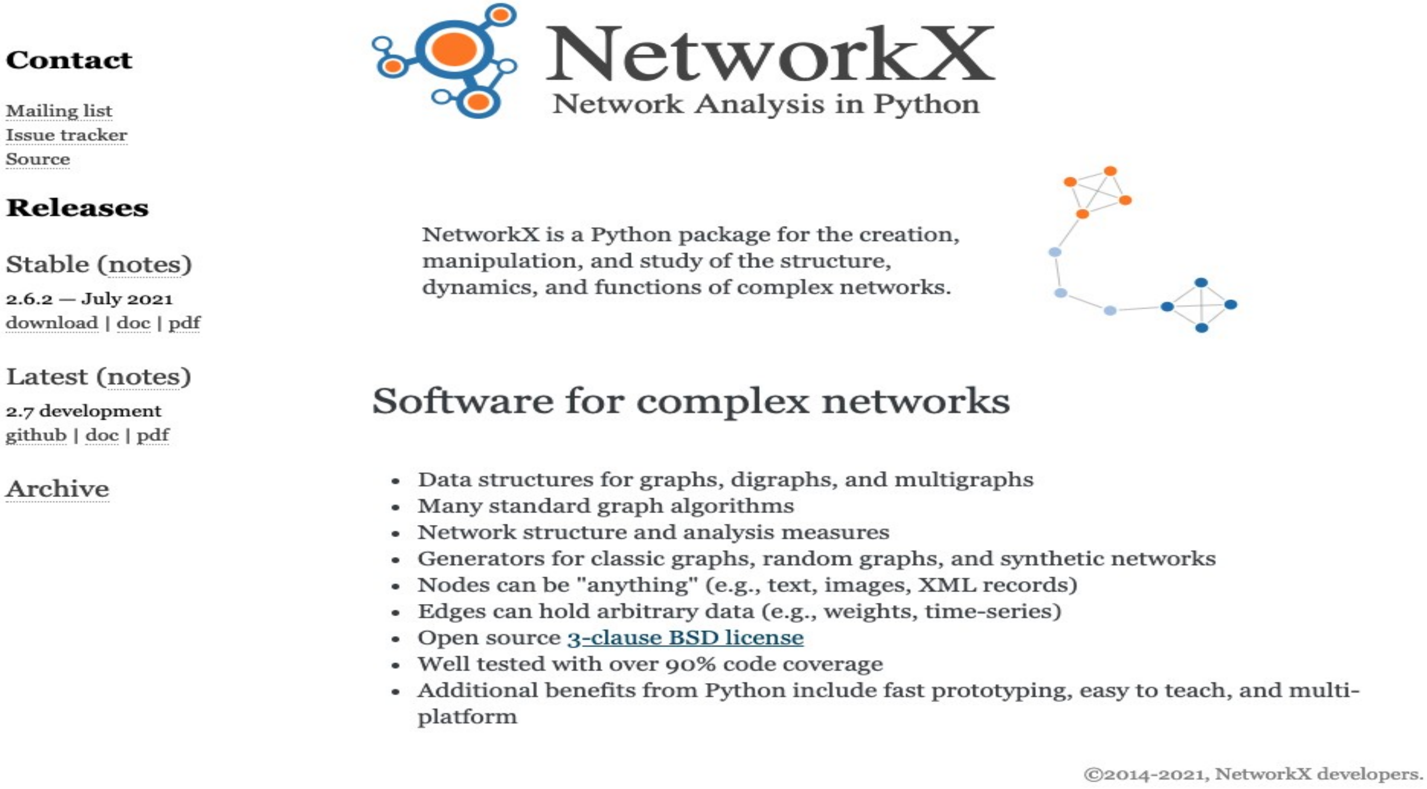


When should I AVOID NetworkX to perform network analysis?

- Large-scale problems that require faster approaches (i.e. massive networks with 100M/1B edges)
- Better use of memory/threads than Python (large objects, parallel computation)

NetworkX: online resources

https://networkx.org/documentation/stable/tutorial.html



Contact

Mailing list
Issue tracker
Source

Releases

Stable (notes)

2.6.2 — July 2021
download | doc | pdf

Latest (notes)

2.7 development
github | doc | pdf

Archive

NetworkX
Network Analysis in Python

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

Software for complex networks

- Data structures for graphs, digraphs, and multigraphs
- Many standard graph algorithms
- Network structure and analysis measures
- Generators for classic graphs, random graphs, and synthetic networks
- Nodes can be "anything" (e.g., text, images, XML records)
- Edges can hold arbitrary data (e.g., weights, time-series)
- Open source 3-clause BSD license
- Well tested with over 90% code coverage
- Additional benefits from Python include fast prototyping, easy to teach, and multi-platform