NetworkKit

An Interactive Tool Suite for High-Performance Network Analysis

Christian L. Staudt, Aleksejs Sazonovs, Henning Meyerhenke

Parallel Computing Group - Institute of Theoretical Informatics - Karlsruhe Institute of Technology (KIT)

NetworKit

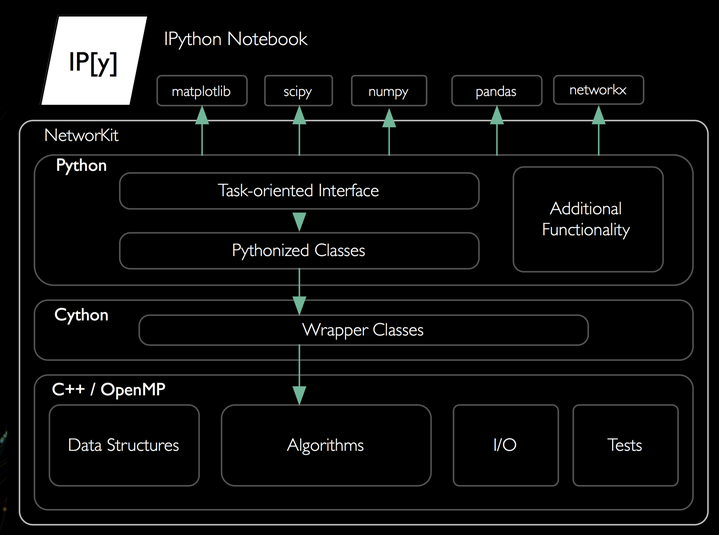
- is an open-source software package for high-performance analysis of large complex networks
- uses shared-memory parallelism and scales from notebooks to compute servers
- combines kernels written in C++ with a convenient interactive interface written in Python.

Design Goals

- performance
- interface
- integration

- scalable algorithms, employing
  parallelism
  approximation
- performance-aware implementation
- modular design
- interactive usage via Python
- Python ecosystem for scientific computing & data analysis
- additional network analysis software (e.g. Gephi, NetworkX)

www.network-analysis.info



NetworKit architecture

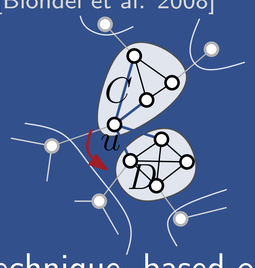
Analytics

Community Detection

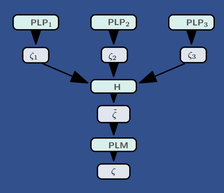
Parallel Community Detection Heuristics

[Staudt, Meyerhenke, ICPP 2013]

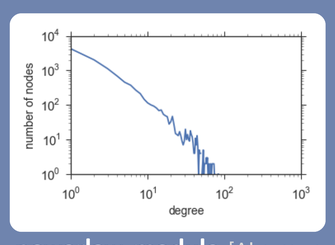
- **PLM**: modularity-driven multi-level technique, based on sequential **Louvain method** [Blondel et al. 2008]
  - high modularity
  - fast, scales to billions of edges



- **PLP**: parallel label-propagation technique, based on [Raghavan et al. 2007]
  - fastest community detection heuristic
  - scales well with the number of processors
- **EPP**: ensemble technique, combining several weak classifiers into a strong one



Degree Distribution



- **powerlaw module** [Alstott et al. 2014] tests statistically for powerlaw distribution

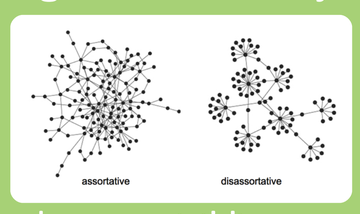
Clustering Coefficients

- local and global
- exact parallel computation in $O(nd_{\max}^2)$ time
- very fast approximation with error guarantee
  [Schank, Wagner 2005]

Additional Algorithms

- **breadth-first** and **depth-first search**
- **Dijkstra's algorithm**
- approximate maximum weight **matching**
- **algebraic distance**
- **maximum flows**
- ...

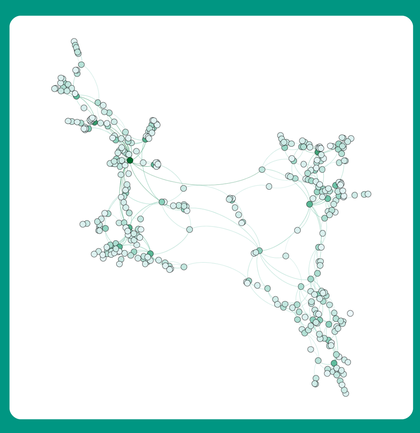
Degree Assortativity



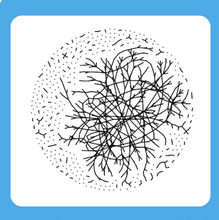
- **degree assortativity coefficient**: correlation of node degrees among neighbors
- $O(m)$ time algorithm

Centrality

- PageRank
- eigenvector centrality
- betweenness centrality [Brandes 2001]
- betweenness approximation
  - fast heuristic [Geisberger, Sanders, Schultes 2008]
  - approximation with maximum error guarantee
    [Riondato, Kornaropoulos 2014]

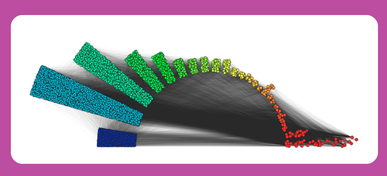


Connected Components



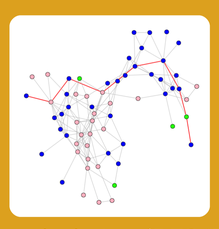
- computed using parallel label propagation scheme

Core Decomposition



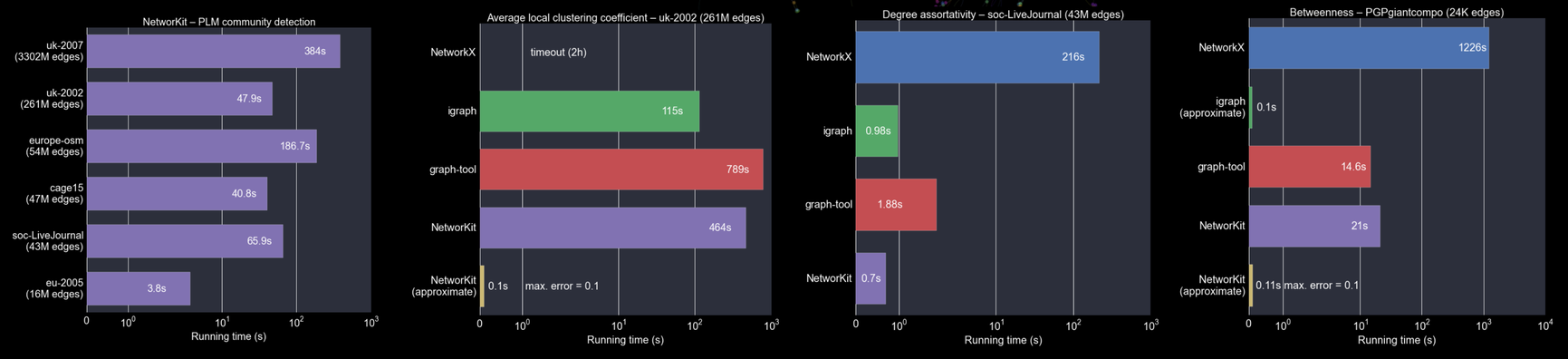
- k-cores result from iteratively peeling away nodes of degree k
- $O(m)$ algorithm
  [Batagelj, Zaversnik 2003]

Diameter



- exact calculation (BFS/Dijkstra)
- fast **approximation** with bounded error [Magien et al. 2009]

Performance



Performance measurements on a shared-memory server with 256 GB RAM and 2x8 Intel(R) Xeon(R) E5-2680 cores (max. 32 threads)

Graph Generators

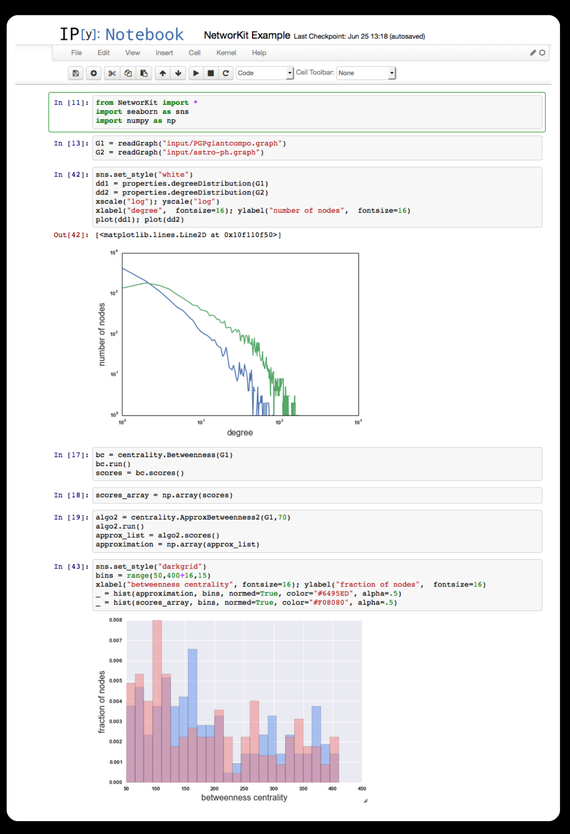
Erdős-Renyi model
- classic random graph model
- fast generator

Barabasi-Albert model
- produces networks with powerlaw degree distribution
- static and dynamic generator

Chung-Lu model
- replicates any given degree distribution

R-MAT generator
- popular high-performance graph generator
- power-law degree distribution, small-world property and self-similarity

PubWeb generator
- geometric disk graph model
- simulates P2P network
- static and dynamic generator



Interactive network analysis using IPython Notebook

Open Source

By publishing NetworKit under the permissive open-source MIT license, we encourage usage and contributions by a community of algorithm engineers and data analysts. We thank all previous contributors.

Get NetworKit: http://www.network-analysis.info

Future
- improved support for dynamic and attributed graphs
- dynamic network analysis algorithms
- sparsification, filtering and compression
- new generative models

KIT – University of the State of Baden- Württemberg and
National Research Center of the Helmholtz Association