Software Tools

- Networkit:
  - Python interface, C++ backend
  - Dynamic algorithms?
  - Supports an approximate algorithm for calculating betweenness centrality (basically a random sampling method, "dynamic approximate betweenness centrality")
  - For networks with 10^7, 10^8 nodes, BC's runtime is only a few minutes --> 10 to 100 times faster than traditional algorithms (ApproxBetweenness and exact betweenness methods)
  - The accuracy might be an issue during implementation
- Raphtory
  - Raphtory's code is more intuitive to write.
  - Smaller and faster? Suitable for demos but not as flexible as graph-tool and networkit.
  - Possibly supports Pytorch projects and GPU.

Software Tools

- XGI
  - Features (provides a standard protocol to unify graph object formats for cross-tool usage):
    - Standardization: Provides a unified data format, enabling compatibility across different tools and platforms.
    - Flexibility: Supports various graph structures, including simple graphs, multigraphs, weighted graphs, etc.
    - Extensibility: Users can extend the XGI format as needed to fit specific application requirements.
  - Higher-order networks
    - Commonly used to study group behaviors and multi-node interactions in complex systems.
    - In biological networks, higher-order networks are used to model multiple interactions between proteins.
    - In social networks, they are used to analyze complex interactions within groups.
  - Mainly used for the analysis and interaction of higher-order networks.

Software Tools

- Rustworkx
  - A NetworkX library implemented in Rust, but also supports Python (Python frontend, Rust backend).
  - Supports both Python and Rust.
  - Consumes the least time when calculating the shortest paths: rustworkx < graph-tool < igraph < NetworkX.
- Structify_net
  - Structify_net is a Python library for generating random networks with predefined structures. It allows users to create networks with specific numbers of nodes and edges and to control the randomness and structure of the networks. The main goal of this library is to provide a general framework to cover community/block structures, spatial structures, and other types of structures.
  - Main features:
    - Structure definition: Users can define the number of nodes and expected edges in the network. Then, a ranking function assigns connection probabilities to node pairs.
    - Multiple models: Structify_net offers various built-in structural models, including block structures, spatial structures, fractal structures, etc. These models can be found in the "Structure Zoo."
    - Visualization and scoring: The library includes visualization tools and scoring functions to help users analyze and present the generated networks.

Tutorial on Cross-Package Network Analysis

- NetworkX --> nx-cugraph
  - Pip install nx-cugraph-cu12 --extra-index-ul=https://pypi.nvidia.com (nx-cugraph)
  - A new method of saving networks --> converted into csv.gz files, not saving node information separately, but saving nodes, edges, and edge weights simultaneously as node pairs + weights.
  - Nx-parallel, a parallel computing method built into NetworkX.
  - Speed estimates for different libraries:
    - NetworkX: 1000 seconds (baseline)
    - nx-parallel: 100 - 500 seconds
    - graph-tool: 10 - 100 seconds
    - cuGraph: 10 - 50 seconds

Tutorial on Cross-Package Network Analysis

- Graph-tool
  - Good visualization effects.
  - Graph_draw(………inline=False) --> operate images in a pop-up window (prerequisite, run in Jupyter).
  - Native save and load functions
    - g.save("filename.gt.gz"),
    - g2 = load_graph("filename.gt.gz").
- Igraph
  - Dual use of Python and R (the functionality in Python seems not as comprehensive as in R).
  - Recommended to use this library when working with R language.