SNAP Package Description

SNAP for C++
SNAP for Python
SNAP Datasets
What's new
People
Papers
Links
About
Contact us

Open positions

Our group has an open research position for the **summer quarter**. **More info here**.

SNAP Directory Structure

The SNAP package contains the following directories:

- `snap-core`:the core SNAP graph library;
- `snap-adv`:advanced SNAP components, not in the core, but used by examples;
- `snap-exp`:experimental SNAP components, still in development;
- `examples`:small sample applications that demonstrate SNAP functionality;
- `tutorials`:simple programs, demonstrating use of various classes;
- `glib-core`: STL-like library that implements basic data structures, like vectors (`TVec`), hash-tables (`THash`) and strings (`TStr`), provides serialization and so on;
- `test`:unit tests for various classes;
- `doxygen`:SNAP reference manuals.

Example Applications

- **agmfit:** Detects network communities from a given network by fitting the Affiliation Graph Model (AGM) to the given network by maximum likelihood estimation.
- **agmgen:** Implements the Affiliation Graph Model (AGM). AGM generates a realistic looking graph from the community affiliation of the nodes.
- **bigclam:** Formulates community detection problems into non-negative matrix factorization and discovers community membership factors of nodes.
- **cascades:** Simulates a SI (susceptible-infected) model on a network and computes structural properties of cascades.
- **centrality:** Computes node centrality measures for a graph: closeness, eigen, degree, betweenness, page rank, hubs and authorities.
- **cliques:** Finds overlapping dense groups of nodes in networks, based on the Clique Percolation Method.
- **community:** Implements network community detection algorithms: Girvan-Newman and Clauset-Newman-Moore.
- **concomp:** Computes weakly, strongly and biconnected connected components, articulation points and bridge edges of a graph.
- **forestfire:** Generates graphs using the Forest Fire model.
- **graphgen:** Generates undirected graphs using one of the many SNAP graph generators.
- **graphhash:** Demonstrates the use of `TGHash` graph hash table, useful for counting frequencies of small subgraphs or information cascades.
- **infopath:** Implements stochastic algorithm for dynamic network inference from cascade data, see Structure and Dynamics of Information Pathways in On-Line Media
- **kcores:** Computes the k-core decomposition of the network and plots the number of nodes in a k-core of a graph as a function of k.
- **kronem:** Estimates Kronecker graph parameter matrix using EM algorithm.
- **kronfit:** Estimates Kronecker graph parameter matrix.
- **krongen:** Generates Kronecker graphs.
- **magfit:** Estimates Multiplicative Attribute Graph (MAG) model parameter.
- **maggen:** Generates Multiplicative Attribute Graphs (MAG).
- **mkdatasets:** Demonstrates how to load different kinds of networks in various network formats and how to compute various statistics of the network.
- **motifs:** Counts the number of occurence of every possible subgraph on K nodes in the network.
- **ncpplot:** Plots the Network Community Profile (NCP).
- **netevol:** Computes properties of an evolving network, like evolution of diameter, densification power law, degree distribution, etc.
- **netinf:** Implements netinf algorithm for network inference from cascade data, see Inferring Networks of Diffusion and Influence
- **netstat:** Computes structural properties of a static network, like degree distribution, hop plot, clustering coefficient, distribution of sizes of connected components, spectral properties of graph adjacency matrix, etc.
- **testgraph:** Demonstrates some of the basic SNAP functionality.

Functionality of the Core SNAP Library

Brief description of SNAP functionality implemented in various files in the distribution package:

snap-core	
`alg.h`	Simple algorithms like counting node degrees, simple graph manipulation (adding/deleting self edges, deleting isolated nodes) and testing whether graph is a tree or a star.
`anf.h`	Approximate Neighborhood Function: linear time algorithm to approximately calculate the diameter of massive graphs.
`bfsdfs.h`	Algorithms based on Breath First Search (BFS) and Depth First Search (DFS): shortest paths, spanning trees, graph diameter, and similar.

bignet.h	Memory efficient implementation of a network with data on nodes. Use when working with very large networks.
centr.h	Node centrality measures: closeness, betweenness, PageRank, ...
cmty.h	Algorithms for network community detection: Modularity, Girvan-Newman, Clauset-Newman-Moore.
cncom.h	Connected components: weakly, strongly and biconnected components, articular nodes and bridge edges.
ff.h	Forest Fire model for generating networks that densify and have shrinking diameters.
gbase.h	Defines flags that are used to identify functionality of graphs.
ggen.h	Various graph generators: random graphs, copying model, preferential attachment, RMAT, configuration model, Small world model.
ghash.h	Hash table with directed graphs (`TNGraph`) as keys. Uses efficient adaptive approximate graph isomorphism testing to scale to large graphs. Useful when one wants to count frequencies of various small subgraphs or cascades.
gio.h	Graph input output. Methods for loading and saving various textual and XML based graph formats: Pajek, ORA, DynNet, GraphML (GML), Matlab.
graph.h	Implements graph types `TUNGraph`, `TNGraph` and `TNEGraph`.
gstat.h	Computes many structural properties of static and evolving networks.
gsvd.h	Eigen and singular value decomposition of graph adjacency matrix.
gviz.h	Interface to Graphviz.
kcore.h	K-core decomposition of networks.
network.h	Implements network types `TNodeNet`, `TNodeEDatNet` and `TNodeEdgeNet`.
Snap.h	Main include file of the library.
statplot.h	Plots of various structural network properties: clustering, degrees, diameter, spectrum, connected components.
subgraph.h	Extracting subgraphs and converting between different graph/network types.
timenet.h	Temporally evolving networks.
triad.h	Functions for counting triads (triples of connected nodes in the network) and computing clustering coefficient.
util.h	Utilities to manipulate PDFs, CDFs and CCDFs. Quick and dirty string manipulation, URL and domain manipulation routines.
snap-adv	
agm.h	Implements the Affiliation Graph Model (AGM).
cliques.h	Maximal clique detection and Clique Percolation method.
graphcounter.h	Performs fast graph isomorphism testing to count the frequency of topologically distinct sub-graphs.
kronecker.h	Kronecker Graph generator and KronFit algorithm for estimating parameters of Kronecker graphs.
mag.h	Implements the Multiplicative Attribute Graph (MAG).
ncp.h	Network community profile plot. Implements local spectral graph partitioning method to efficiently find communities in networks.
subgraphenum.h	Sub-graph enumeration and network motif computations.
snap-exp	
arxiv.h	Functions for parsing Arxiv data and standardizing author names.
dblp.h	Parser for XML dump of DBLP data.
imdbnet.h	Actors-to-movies bipartite network of IMDB.
mxdag.h	Finds the maximum directed-acyclic subgraph of a given directed graph.
signnet.h	Networks with signed (+1, -1) edges that can denote trust/distrust between the nodes of the network.
sir.h	SIR epidemic model and SIR parameter estimation.
spinn3r.h	Past parser for loading blog post data from Spinn3r.
trawling.h	Algorithm of extracting bipartite cliques from the network.
wgtnet.h	Weighted networks.
wikinet.h	Networks based on Wikipedia.