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 Installation and Compilation

Prerequisites

SNAP works under Windows with Visual Studio or Cygwin with GCC, Mac OS X, Linux and other Unix variants with GCC. Make sure that a C++ compiler is installed on the system.

SNAP is largely self-contained and requires external packages only for drawing and visualization. The following packages need to be installed on the system to support drawing and visualization in SNAP:

- Gnuplot for plotting structural properties of networks (e.g., degree distribution);
- Graphviz for drawing and visualizing small graphs.

Set the system PATH variable, so that Gnuplot and Graphviz are available, or put the executables in the working directory.

SNAP Components

The SNAP distribution package contains the following components:

- `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.

More details about the package contents are available **HERE**!

Installation

To install SNAP, download the latest SNAP distribution package **HERE**, and unzip it.

For plotting structural properties of networks (e.g., degree distribution) SNAP expects to find Gnuplot in the system path. Similarly for drawing and visualizing small graphs SNAP utilizes Graphviz. If needed, install those packages and make sure that the system PATH variable points to their executables or put the executables in the working directory.

Compilation of SNAP Programs

SNAP on MAC OS X, Linux and Windows with Cygwin

Make sure that the GCC package with a C++ compiler is installed on your system.

Run "make all" in the main SNAP directory. This compiles the core SNAP library and all the application examples.

For other platforms, modify `Makefile.config` in the main SNAP directory.

SNAP on Windows with Visual Studio

Make sure that Visual Studio is installed on your system. Start Visual Studio and open the solutions file SnapExamples in the `examples` directory.

Configure Visual Studio for use with SNAP, which requires that the character set is set to Multi-byte and that the locations of SNAP include directories are specified.

To set the character set to Multi-byte, right-click on your project, go to *Properties –> Configuration Properties –> General –> Projects Defaults –> Character Set –> Select "Use Multi-Byte Character Set"*.

To specify the location of SNAP include directories, go to *Options –> Preferences –> C++ Directories –> Include Directories* and add the paths on your system to SNAP folders `glib-core`, `snap-core` and `snap-adv`.

Be aware that your version of Visual Studio might have the settings at slightly different locations than used above, so the navigation steps to find them need to be adjusted appropriately.

Once Visual Studio is configured, build your solution, which compiles the core SNAP library all the examples.

Execution of SNAP Programs

After the code is compiled as shown above, it can be executed. The `graphgen` application is used here as an example.

To run the `graphgen` example, open a command line application and execute the following command:

```
cd examples/graphgen
./graphgen -g:w -n:1000 -k:4 -p:0.1 -o:smallworld.txt
```

The command generates a Watts-Strogatz small-world graph where each node is connected to k=4 closest nodes and with the rewiring probability of p=0.1.