JUNG

Java Universal Network/Graph Framework

Links

JUNG Frequently Asked Questions

General questions

Technical questions

(See also the Developer's FAQ for more detailed information.)

1. # What is JUNG?

   JUNG is a Java-based open-source software library designed to support the modeling, analysis, and visualization of data that can be represented as graphs. Its focus is on mathematical and algorithmic graph applications pertaining to the fields of social network analysis, information visualization, knowledge discovery and data mining. However, it is not specific to these fields and can be used for many other applications pertaining to graphs and networks.

2. # Who created JUNG?

   JUNG was created by three Information and Computer Science PhD students at the University of California, Irvine: Joshua O'Madadhain, Danyel Fisher, and Scott White.

3. # I heard JUNG is open-source. What license does it use?

   JUNG is licensed and made freely available under the Berkeley Software Distribution (BSD) license.

4. # What does JUNG stand for?

   JUNG stands for "**J**ava **U**niversal **N**etwork/**G**raph" (Framework). If you find various references in the code such as ArchetypeGraph you can be sure that we did not forget about our dear friend Carl. :) (JUNG is not, however, associated with any of the various graph theorists named Jung, such as Heinz Jung or Hwan-Ok Jung.)

5. # Why was JUNG written in Java?

   Java is a language that arguably has the most to offer people who want to analyze and visualize complex networks. It provides strong support for object-oriented design, excellent developer tools for writing code, and an extensive API for such tasks as database connectivity, writing GUIs and layout algorithms, and web support. In addition, it is a widely adopted programming language, so there are many third-party packages that can be leveraged for statistical analysis, data structures, visualization, etc.

6. # How do I use JUNG?

   JUNG is a software library, so you write Java programs that call our routines. Programs that use JUNG can be simple (snippets of code to test out algorithms or ideas) or sophisticated (applications with GUIs).

7. # Can I contribute to the Jung Project?

   Absolutely; we welcome contributions! If you have some specific code you'd like to contribute, please send a package containing your code, documentation, and unit tests, to the mailing list jung-support@lists.sourceforge.net. (You can find examples of unit tests in our distribution, in the test package.) Once we feel comfortable that your contributions are consistently of good quality, we may invite you to become an official developer. We also have specific capabilities that we'd like to

add to JUNG, so if you'd like to help us create them, please send a description of your background and the area(s) of the code you'd like to work on to the mailing list. Current areas where we could use help are: social network algorithms, 2D graph drawing algorithms, general graph theory algorithms, and general statistical routines for network analysis.

8. What specific algorithms does JUNG provide?

JUNG provides a number of algorithms. Since new algorithms are constantly being added it's best to look at the Javadoc documentation. General algorithms are listed under the jung.algorithms package.

9. Does JUNG support dynamic graphs?

Yes, JUNG supports dynamic graphs that can be changed both through a system of filters or by explicitly adding and removing nodes. Either way, it's easy to visualize the results, to apply graph algorithms to the results, and to manipulate those results further.

10. Is there a limit to the size of graphs JUNG can handle?

No. The only limit to the size graphs JUNG can support is the hardware you are using to run the Java Virtual Machine. We have on many occasions used JUNG to analyze sparse graphs with more than 150,000 nodes.

11. What kind of data can JUNG attach to graph elements (e.g. labels, weights, ...)?

One of JUNG's key strengths is that it can annotate graphs, vertices (entities), or edges (relations) with any Java data type.

12. Is JUNG specific to Swing? Can I use JUNG with SWT, or other windowing toolkits?

The renderers that are currently available in the JUNG distribution use the Java Swing tools. However, the renderer (the piece of the library that draws things on the screen) is decoupled from the layout algorithm (the piece of the library that determines where things get drawn), so you can use other renderers if you like.
Update: a member of the JUNG community (Lucas Bigeardel) has contributed an alpha version of a SWT port of the JUNG visualization system; you can find it here.

13. What version of Java does JUNG use? What third-party libraries does JUNG use?

See the download page for the most up-to-date answer to this question.

14. Why did you create JUNG? How does it differ from existing tools?

We created JUNG because we perceived a need for a general, flexible, and powerful API for manipulating, analyzing, and visualizing graphs and networks in Java. However, there are other tools available for manipulating networks, which may be more appropriate for you, depending on your specific needs and abilities.

The following is a very brief summary of our understanding of the major distinctions between JUNG and a few of the more popular tools for network analysis and visualization. It is not intended to be a complete characterization of any of these tools; see the resources provided by the tool's developers for details. (If you believe that we have misrepresented the capabilities of one of these tools, please contact us and we will edit this text accordingly.)

15. How is JUNG different from...

1. ... UCINET?

UCINET is a widely-used application among social networks researchers for performing standard social network analysis techniques to graphs.
However, UCINET cannot be embedded into applications: you can't call UCINET in an end-user display.
JUNG provides facilities to dynamically change graphs, to programatically call code, and to output the results as the program continues.

2. ... PAJEK?

PAJEK is a stand-alone tool for visualizing and analyzing networks. JUNG provides many algorithms that PAJEK does not (and, currently vice versa), and--as noted for UCINET--is easily incorporated into network applications.
JUNG is capable of both reading and writing simple PAJEK-format files. (JUNG's PAJEK file reader does not currently support the entire PAJEK file format.)

3. ... R? http://www.r-project.org

R is a specialized programming language geared primarily toward the statistics community, offering a broad set of statistical routines. JUNG is intended for a less-specialized audience, and, as a pure JAVA solution, is embeddable within web browsers and pre-existing applications.

4. ... GFC? http://www.alphaworks.ibm.com/tech/gfc

GFC is a graph drawing-oriented package released by IBM. It is specific to using Java's AWT/Swing, and contains few graph manipulation algorithms.

JUNG is open-source, free, and has a wide variety of algorithms available. Better, it's easily extensible through a widely-documented API: if it's not there yet, you can add it yourself.

16. What types of graphs does JUNG support?

JUNG supports graphs, general k-partite graphs (of which bipartite graphs are a special case), hypergraphs, and has limited support for trees.