

Extracted from:

Complex Network Analysis in Python

Recognize → Construct → Visualize → Analyze → Interpret

This PDF file contains pages extracted from *Complex Network Analysis in Python*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2018 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Raleigh, North Carolina

Complex Network Analysis in Python

*Recognize → Construct → Visualize →
Analyze → Interpret*



Dmitry Zinoviev
edited by Adaobi Obi Tulton

Complex Network Analysis in Python

Recognize → Construct → Visualize → Analyze → Interpret

Dmitry Zinoviev

The Pragmatic Bookshelf

Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Publisher: Andy Hunt

VP of Operations: Janet Furlow

Managing Editor: Brian MacDonald

Supervising Editor: Jacquelyn Carter

Development Editor: Adaobi Obi Tulton

Indexing: Potomac Indexing, LLC

Copy Editor: Nicole Abramowitz

Layout: Gilson Graphics

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2018 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-13: 978-1-68050-269-5

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—January 2018

*To my beautiful and most intelligent wife,
Anna, and to our children: graceful ballerina,
Eugenia, and romantic gamer, Roman.*

Explore the Network

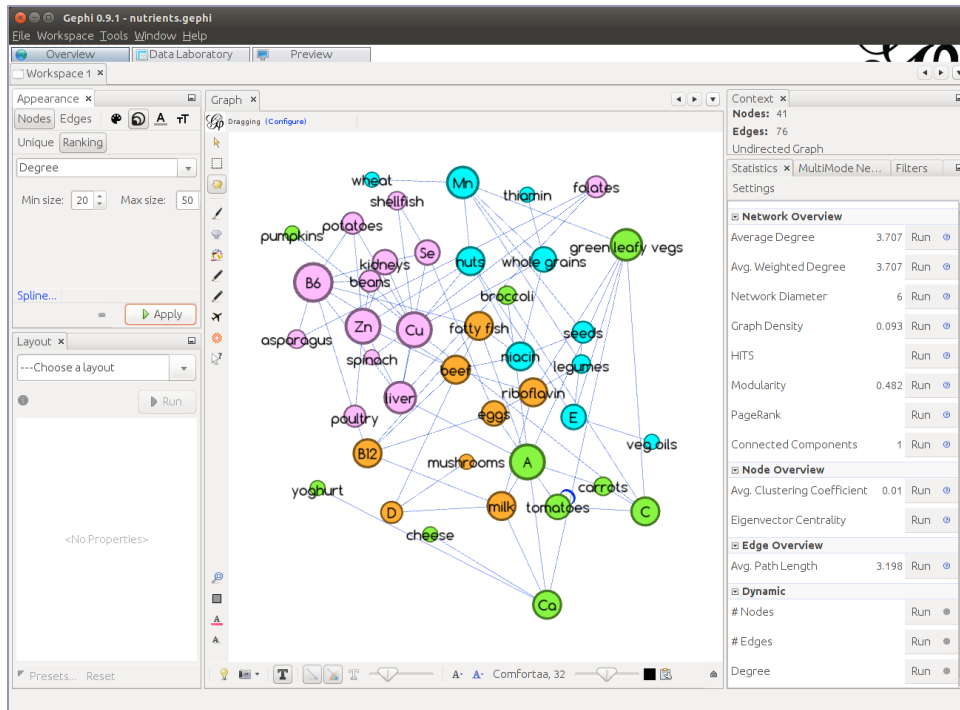
Network exploration in Gephi goes hand in hand with selecting visual properties. Let's paint and resize the graph nodes based on some of their measures.

You will learn about network measures in [Chapter 8, *Measuring Networks*, on page ?](#) and [Chapter 11, *Unearthing the Network Structure*, on page ?](#). For now, it suffices to know several basic facts about two of them, as detailed in the following table:

Measure	Meaning
Degree	The number of immediate neighbors—adjacent nodes. The degree is a non-negative integer number. The larger the degree of a food item is, the more nutrients it provides. The larger the degree of a nutrient is, the more food items provide it.
Community structure	Nodes form tightly knit groups called communities. All foods and nutrients within a community serve some common purpose. Each community has a unique integer identifier called modularity class.

Node degree is the simplest possible node measure. There is no need to calculate it explicitly. To make node size proportional to the degree, click the icon with concentric circles in the Appearance window, then on the Ranking button. Select Degree from the “—Choose an attribute” pull-down menu. Select node sizes that correspond to the smallest and largest degrees (10 and 40 are good choices). And don't forget to click Apply. Can you see which nodes have the highest degrees?

Playing with size is fun; playing with color is more fun. Let's paint the nodes according to their modularity classes, as I've done in the [figure on page 8](#). To partition a network into communities, click the Run button next to the Modularity command in the Network Overview section of the Statistics window. Proceed by clicking OK and Close in the next two dialogs. In the end, you will see a floating-point number next to Modularity. The number is a measure of the quality of the decomposition from [Outline Modularity-Based Communities, on page ?](#). Return to the artist's palette icon, then click the Partition button. Select Modularity Class from the “—Choose an attribute” pull-down menu. Don't forget to click Apply. If you feel artistic, like me, play with the node colors. Painting the beef group pink and the vegetable group green is a no-brainer, but can you choose a good color for the vitamins?



If you plan to use Gephi for more sophisticated CNA jobs, the [table on page 9](#) will help you find which sections of this book match the items in the Statistics window.

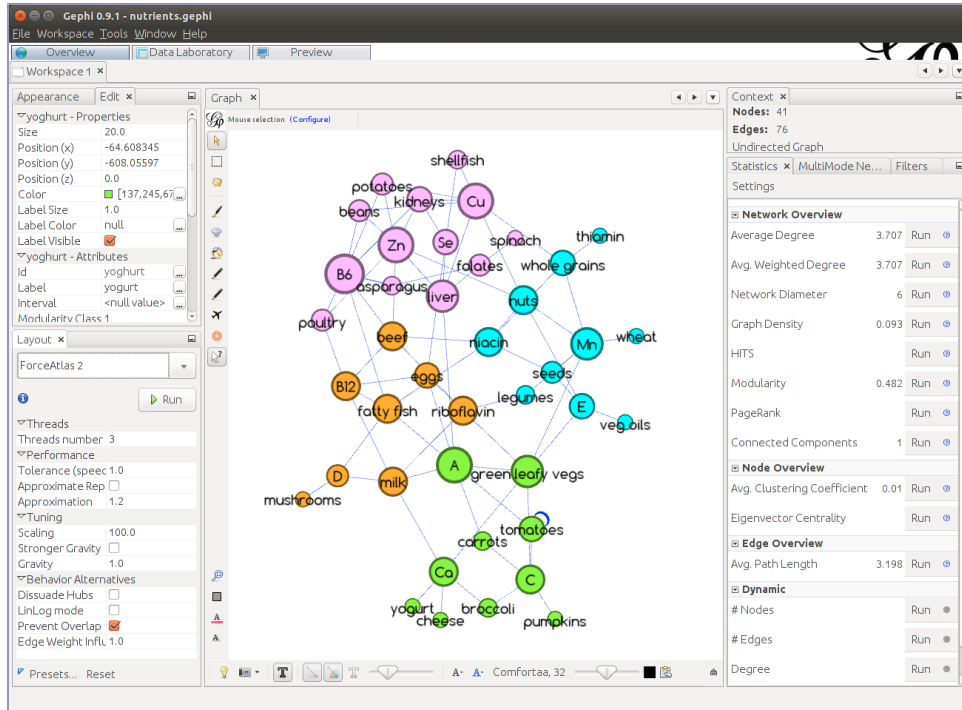
Sketch the Network

You're done with rough rendering, but the layout is still awful. Let's turn our attention to the lower-left corner of Gephi. Select your favorite layout from the "Choose a layout" pull-down menu. When a network is large (500 or more nodes), the Fruchterman-Reingold layout is usually the most efficient. For smaller networks, the ForceAtlas 2 layout, with some tweaking, works marvels. To make things easier, here's a tip: set the scaling to 100.0 (to place nodes reasonably far apart) and check the Prevent Overlap box. Run the tool for a while. You'll notice that after a couple of seconds, the nodes settle at their new positions, but the graph as a whole may continue drifting, rotating, or both.

The last step is to adjust the labels, because surely some of them don't mind their manners and sit on top of each other. Select Label Adjust from the "Choose a layout" pull-down menu and run the tool for a couple of seconds. This layout engine distorts the original Fruchterman-Reingold but makes

Measure	NetworkX reference
Average Degree	<i>Degree Centrality, on page ?</i>
Network Diameter	<i>Think in Terms of Paths, on page ?</i>
↑ Also calculates betweenness centrality	<i>Betweenness Centrality, on page ?</i>
↑ Also calculates closeness centrality	<i>Closeness and Harmonic Closeness Centrality, on page ?</i>
↑ Also calculates eccentricity	<i>Networks as Circles, on page ?</i>
Graph Density	<i>Start with Global Measures, on page ?</i>
HITS (Hubs and Authorities)	<i>HITS Hubs and Authorities, on page ?</i>
Modularity	<i>Outline Modularity-Based Communities, on page ?</i>
PageRank	<i>PageRank, on page ?</i>
Connected Components	<i>Split Networks into Connected Components, on page ?</i>
Avg. Clustering Coefficient	<i>Explore Neighborhoods, on page ?</i>
Eigenvector Centrality	<i>Eigenvector Centrality, on page ?</i>
Avg. Path Length	<i>Think in Terms of Paths, on page ?</i>

sure that neither nodes nor labels overlap. Hopefully, your network layout resembles the following figure.



The Graph window still shows a sketch, but this is a high-quality sketch that nicely displays the structure of the network of foods and nutrients. There are five compact groups in the network that could be tentatively called Veggies, Cereals, Meats, Proteins, and Folate. You can explore each group's internal composition, as well as connections to the other groups. You can even show this sketch to your boss or customer. But it would look much better when rendered at high resolution and converted to a presentation-quality image. (Save the project via File > Save As... into a .gephi file to avoid data loss if Gephi crashes!)