

NODEXL for Beginners

Nasri Messarra, 2013-2014

<http://nasri.messarra.com>

Why do we study social networks?

Definition from: http://en.wikipedia.org/wiki/Social_network: A social network is a social structure made up of a set of social actors (such as individuals or organizations) and a set of the dyadic ties between these actors. Social networks and the analysis of them is an inherently interdisciplinary academic field which emerged from social psychology, sociology, statistics, and graph theory.

From <http://en.wikipedia.org/wiki/Sociometry>: "Sociometric explorations reveal the hidden structures that give a group its form: the alliances, the subgroups, the hidden beliefs, the forbidden agendas, the ideological agreements, the 'stars' of the show".

In social networks (like Facebook and Twitter), sociometry can help us understand the diffusion of information and how word-of-mouth works (virality).

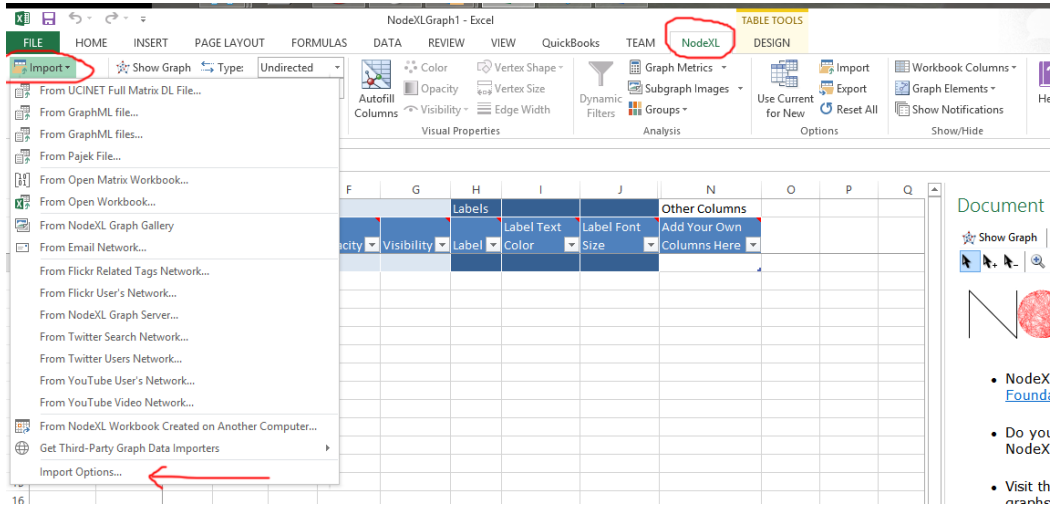
Installing NODEXL (Microsoft Excel required)

NodeXL Template 2014

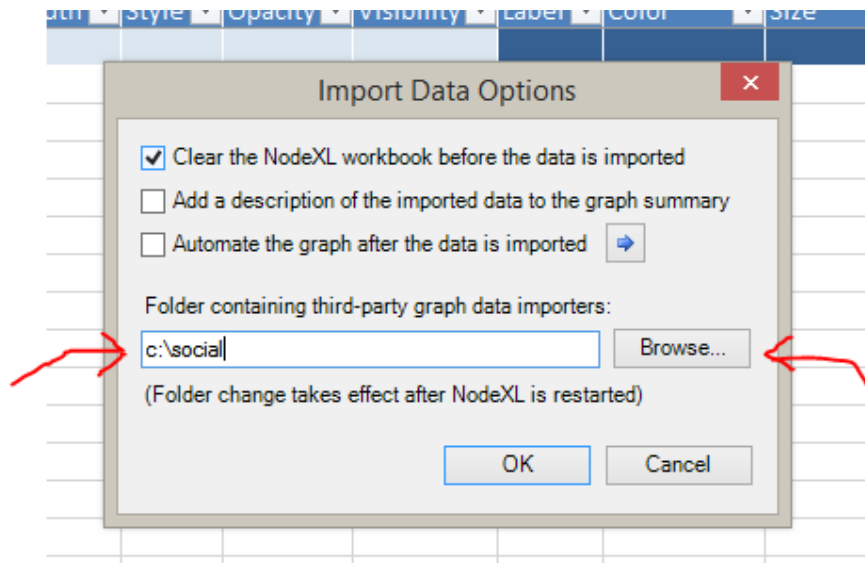
- Visit <http://nodexl.codeplex.com>
- Download the latest version of NodeXL
- Double-click, follow the instructions

The SocialNetImporter extends the capabilities of NodeXL mainly with extracting data from the Facebook network. To install:

- Download the latest version of the social importer plugins from <http://socialnetimporter.codeplex.com>
- Open the Zip file and save the files into a directory you choose, e.g. c:\social
- Open the NodeXL template (you can click on the Windows Start button and type its name to search for it)
- Open the NodeXL tab, Import, Import Options (see screenshot below)



- In the import dialog, type or browse for the directory where you saved your social importer files (screenshot below):



- Close and open NodeXL again

For older Versions:

- Visit <http://nodexl.codeplex.com>
- Download the latest version of NodeXL
- Unzip the files to a temporary folder
- Close Excel if it's open
- Run setup.exe
- Visit <http://socialnetimporter.codeplex.com>
- Download the latest version of the socialnetimporter plug in

- Extract the files and copy them to the NodeXL plugin direction. Defaults to `C:\Program Files\Social Media Research Foundation\NodeXL Excel Template\Plugins`

First steps with NodeXL

The following table is a matrix showing trust within a group of 6 people. An "X" in a cell means that the person who's name is in the cell's row trusts the person who's name is in the cell's column

	Ann	Bob	Claire	Don	Edna	Fred
Ann		X			X	
Bob				X	X	
Claire				X	X	X
Don		X				
Edna	X	X		X		
Fred	X	X			X	

Because trust is not automatically reciprocated (Ann trusting Bob does not necessarily mean that Bob trusts Ann), the graph that we will build will be **directed**.

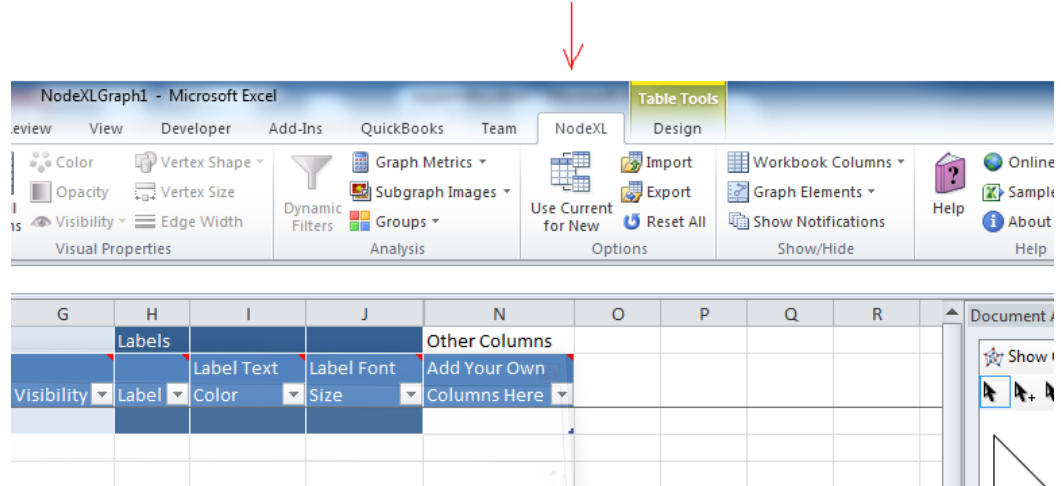
A directed graph implies that the **edges** (links) between two **vertices** (in our example, people) have a direction: A *trusts* B. An example of a directional graph is Twitter. In Twitter, you can follow a person that does not follow you and vice-versa.

In an undirected graph, the edge is reciprocated. This means that, if you have a connection to a person, this person has the same connection towards you. An example of an undirected graph is Facebook: if A is a friend of B then, automatically, B is a friend of A.

Running NodeXL

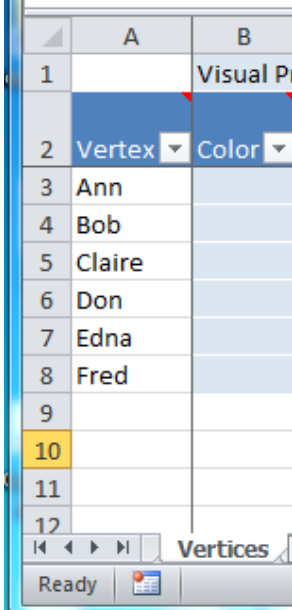
- In the Windows Start menu, click "All Programs", then "NodeXL," then "Excel Template." (in Windows 8, open the tile menu and type "nodexl" to search and find the program)

- Notice the new “NodeXL” tab in the Ribbon:



Drawing your first network graph

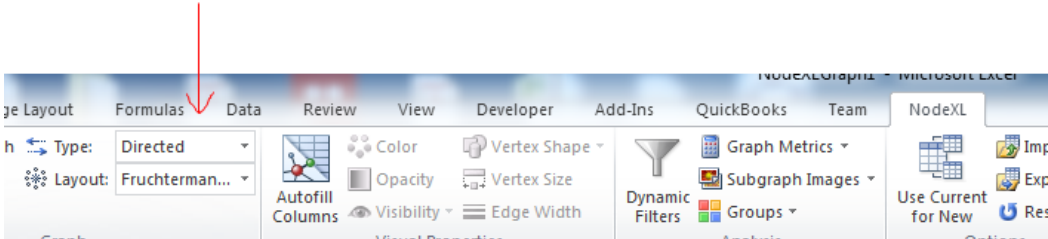
Open the vertices sheet and enter the name of the persons from the social matrix provided (above):



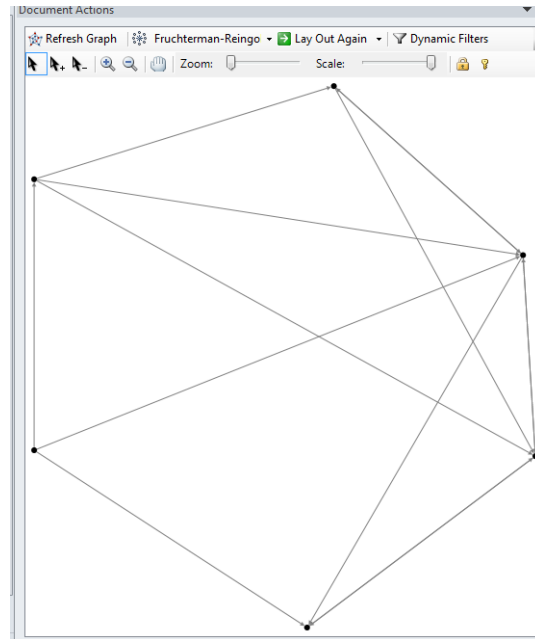
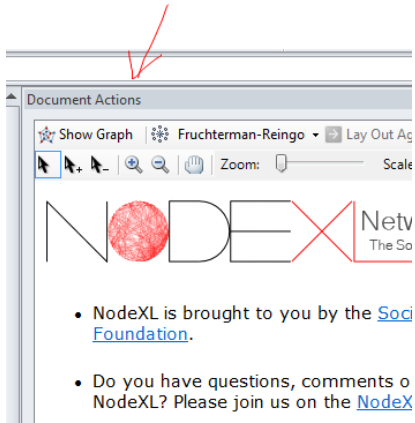
Go to the edges sheet and enter the name of the trusting person in the Vertex 1 column and the name of the trusted person in Vertex 2.

1		
2	Vertex 1 ▾	Vertex 2 ▾
3	Ann	Bob
4	Ann	Edna
5	Bob	Don
6	Bob	Edna
7	Claire	Don
8	Claire	Edna
9	Claire	Fred
10	Don	Bob
11	Edna	Ann
12	Edna	Bob
13	Edna	Don
14	Fred	Ann
15	Fred	Bob
16	Fred	Edna
17		
18		

On the new NodeXL tab, define the graph as “directed”.



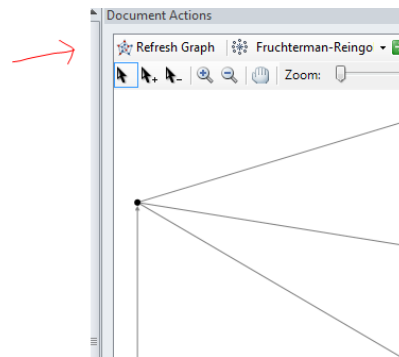
Click the “show graph” button for a visual representation of your graph:

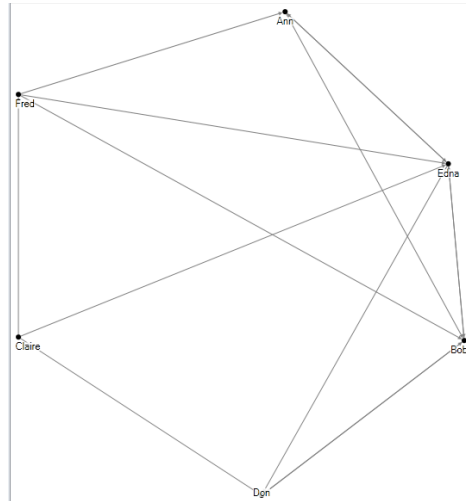


Copy the names in the Label column and refresh your graph to have the names displayed next to the vertices:

	A	B	C	D	E	F	G	H	I	
1		Visual Properties							Labels	Label F
2	Vertex	Color	Shape	Size	Opacity	Image	Visibility	Label	Color	
3	Ann					http://static4.wikia.r	Ann			
4	Bob					http://burnaway.org	Bob			
5	Claire					http://2.bp.blogspot	Claire			
6	Don					http://24.media.tum	Don			
7	Edna					http://fc09.deviantaj	Edna			
8	Fred					https://encrypted-tb	Fred			
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										

Vertices Groups Group Vertices Overall Metrics

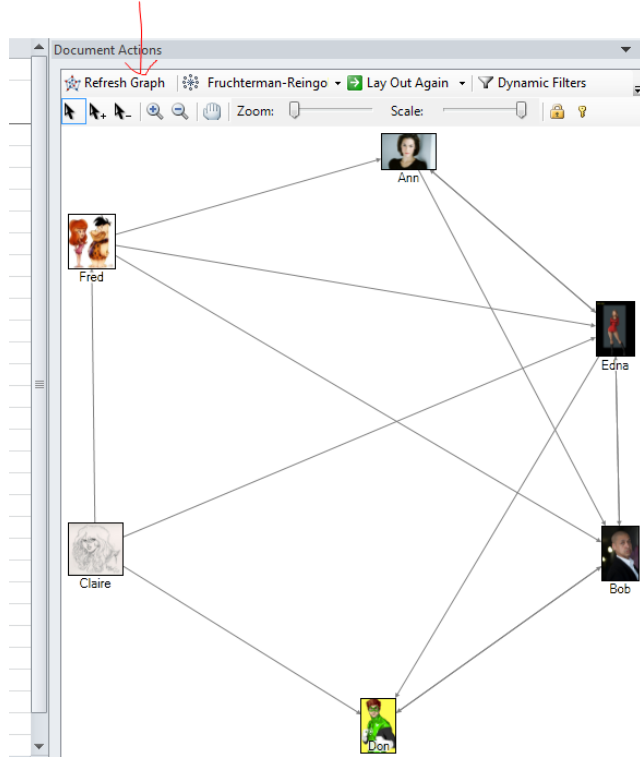




To replace vertices with images, copy and paste photo links (from Facebook profile photos or elsewhere) into the image column and define the shape (in the shape column) as “Image”:

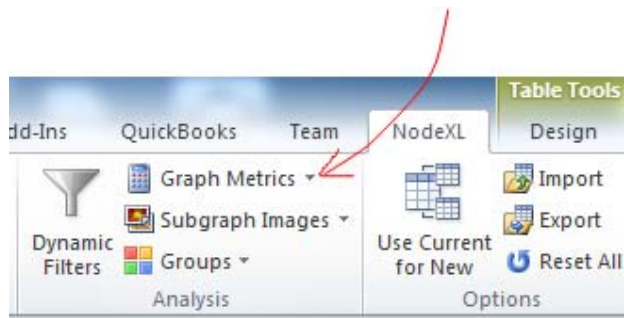
	A	B	C	D	E	F	G	H	I	
1		Visual Properties						Labels		
2	Vertex	Color	Shape	Size	Opacity	Image	Visibility	Label	Color	
3	Ann		Image			http://static4.wikia.r		Ann		
4	Bob		Image			http://burnaway.org		Bob		
5	Claire		Image			http://2.bp.blogspot		Claire		
6	Don		Image			http://24.media.tum		Don		
7	Edna		Image			http://fc09.devianta		Edna		
8	Fred		Image			https://encrypted-tb		Fred		
9										
10										
11										

Refresh the graph:

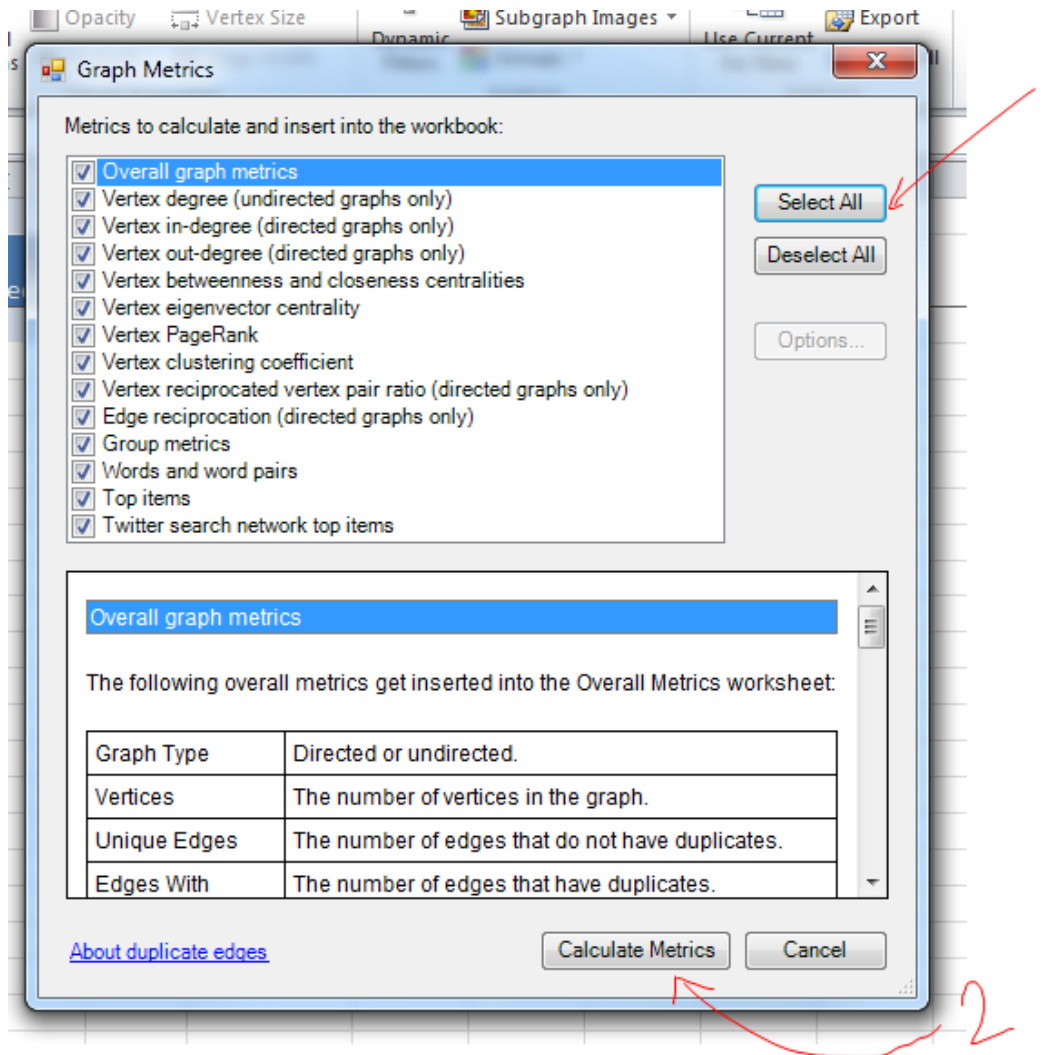


Analyzing Data

To calculate the metrics from the graph, go to the NodeXL tab and click on “Graph Metrics”:



Click on the “Select All” button and “calculate metrics”:



Understanding Metrics

In the Vertices sheet, scroll the right to display the metrics columns:

	A	R	S	T	U	V	W	X	Y	Z
1		Graph Metrics								
2	Vertex	Degree	In-Degree	Out-Degree	Betweenness Centrality	Closeness Centrality	Eigenvector Centrality	PageRank	Clustering Coefficient	Reciprocated Vertex Pair Ratio
3	Ann	3			0.000	0.143	0.151	0.833	1.000	
4	Bob	4			0.833	0.167	0.180	1.079	0.667	
5	Claire	3			0.333	0.143	0.140	0.843	0.667	
5	Don	3			0.333	0.143	0.140	0.843	0.667	
7	Edna	5			1.667	0.200	0.209	1.322	0.600	
8	Fred	4			0.833	0.167	0.180	1.079	0.667	

- In degree: most trusted person (Edna, Bob). It is a count for the number of “arrows” that link to the person (people who trust the vertex).
- Out degree: most trustful (Edna, Fred, Claire). It is a count for the number of “arrows” that go out from this person (how many people are trusted by the vertex).

- Prestige is a metric you can choose to calculate yourself, it is the number of links to the person divided by the total number of possible links (in this example, the maximum would be 5 links or 5 people trusting the 6th person). A prestige of 1 means that "everyone trusts the vertex"

$$prestige = \frac{\text{number of people linking to the vertex}}{\text{population} - 1}$$

- Betweenness centrality (Centralité d'intermédiation) : A high number means that the person is in a central position in the graph. This metric is based on the shortest paths between all people in the graph. It measures how important a node is by counting the number of shortest paths that it is a part of.

how many pairs of individuals would have to go through the vertex in order to reach one another in the minimum number of hops?

- Closeness Centrality (Centralité de proximité) : who has faster access to information (in case of information diffusion). This number is an evaluation of the **average distance** to all nodes in the network.

What if it is not so important to have many direct friends? or be "between" others. Yet, one still wants to be in the "middle" of things, not too far from the center.

- Eigenvector Centrality (Centralité vectorielle) : Eigenvector centrality calculates the position of a node to **all other nodes in the network** by giving a weight based on the distance (La centralité vectorielle essaie de mesurer l'importance ou l'influence d'un nœud dans le réseau en donnant un poids relatif à chaque lien du réseau. L'idée étant que la centralité d'un nœud est égale à la centralité des liens auxquels il est rattaché. → on tient compte de l'importance des liens auquel il est attaché). It is different from betweenness centrality in the sense that it considers all paths between nodes, not only the shortest one.

Eigenvector Centrality measures the importance of a node by the importance of its neighbors.

Eigenvector centrality is a measure of the influence of a node in a network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes

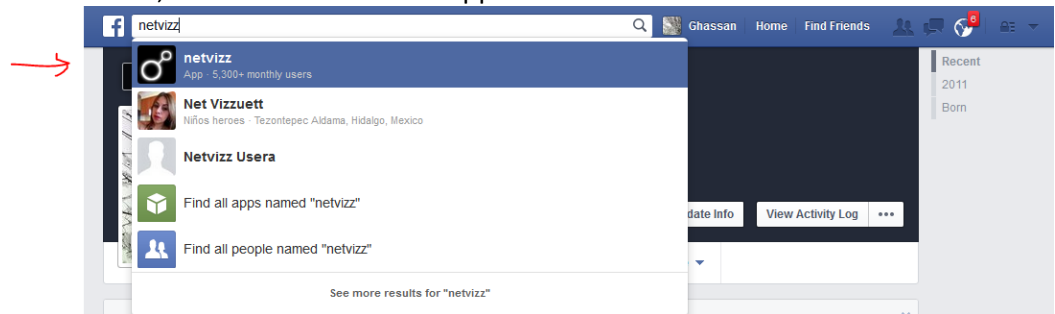
- Centralité de Katz and page rank: Alors que le degré de centralité mesure le nombre de liens adjacents, la centralité de Katz mesure tous les nœuds qui peuvent être connectés à travers un chemin en pénalisant les nœuds distants.

Katz and Page Rank Centrality are a median solution between degree centrality (shortest paths) and eigenvector centrality (all paths) as they measure the number of all nodes that can be connected through a path, while the contributions of distant nodes are penalized.

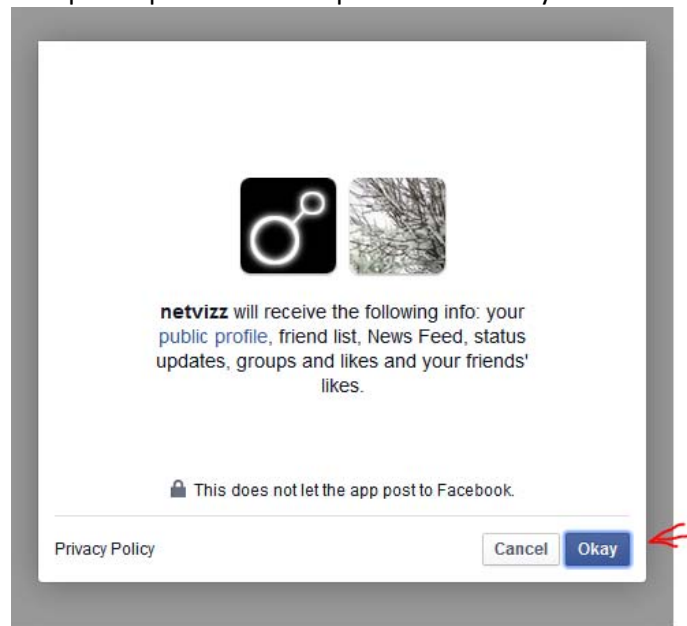
- Clustering coefficient : The clustering coefficient is a measure of an "all-my-friends-know-each-other" property. If the value of the Clustering Coefficient is equal to 1, this means that each of my friends is friend with all the others.
- Reciprocated vertex pair ratio: reciprocated vertex pair ratio: ratio between ingoing and outgoing connections (only valid in directed graphs).

Drawing your Facebook Social graph using NetVizz

1- In Facebook, search for the NetVizz application:



2- Accept the permissions required to extract your friends list and likes



3- Select "personal network":

netvizz v1.01

Netvizz is a tool that extracts data from different sections of the Facebook platform (personal profile, groups, pages) for research purposes. Fil


For questions, please consult the [FAQ](#) and [privacy](#) sections. Non-commercial use only.

New: there is now an [overview video](#) that introduces the different modules and other things to consider.

Big networks may take some time to process. **Be patient and try not to reload!**

Developing and hosting netvizz costs time and money. If the tool is useful for you, please consider to [Donate](#)

The following modules are currently available:

[personal network](#) - extracts your friends and the friendship connections between them 
[personal like network](#) - creates a network that combines your friends and the objects they liked in a bipartite graph
[group data](#) - creates networks and tabular files for both friendships and interactions in groups
[page like network](#) - creates a network of pages connected through the likes between them
[page data](#) - creates networks and tabular files for user activity around posts on pages

1.01 - 03.07.2014 - added date range selector to page module
1.0 - 24.01.2014 - big refactoring, move to module structure, bugfixes and optimizations, file field descriptions
0.93 - 20.10.2013 - Like network changes: fixing call limit, removing location, adding liked object category and full likecount
0.92 - 03.10.2013 - Various changes to lower load on the API
0.91 - 03.10.2013 - Bugfixes for pages
0.9 - 07.07.2013 - Added page like network feature ([blog post](#)), new page engine (with threaded comments), maintainance
0.84 - 18.05.2013 - Added "link" field to page output, moved to new server
0.83 - 04.05.2013 - Retrieved comments for pages limited to 800 per post because of API changes
0.82 - 12.04.2013 - Optimizations, smaller bugfixes
0.81 - 23.03.2013 - All group data anonymized, pages local bug fixed
0.8 - 15.03.2013 - Comment extraction for pages, minor bugfixes
0.73 - 01.03.2013 - Bug fixes for ego network counts, dynamic request frame sizing for handling very large pages, donate button
0.72 - 15.01.2013 - Page feature now gets both page and user posts
0.71 - 30.12.2012 - More maintainance
0.7 - 14.12.2012 - Maintainance, housekeeping, new measures for personal profiles, users are anonymized for page data
0.61 - 20.11.2012 - Added group interactions feature
0.6 - 23.10.2012 - Added page data feature ([blog post](#))
0.51 - 09.09.2012 - Cosmetic changes
0.5 - 08.09.2012 - Added like network feature ([blog post](#))
0.44 - 25.05.2012 - Major bug fixed, huge and groups should be much faster now
0.43 - 23.05.2012 - Finally moved to new server, https works now
0.42 - 30.01.2012 - Group methods updated
0.41 - 11.10.2011 - Fixed group permissions
0.4 - 29.9.2011 - Moved most things to the graph API

IP: 82.94.190.199 / UID: 100002203230540

4- Click on "start":

netvizz v1.01

your personal friend network:

Creates a network file (gdf format) with all the friendship connections in your personal network, as well as a s

Select user data to include in the file (sex, interface language, and account age ranking are standard):

friends' like and post count (public and visible to logged user, first 1000 only), includes counts for receive

start



file fields (network file - gdf format - nodes are users):

sex: user specified sex

locale: user selected interface language

agerank: accounts ranked by creation date where 1 is youngest

like_count: number of user likes

post_count: number of user posts

post_like_count: number of likes on user's posts

post_comment_count: number of comments on user's posts

post_engagement_count: $post_comment_count + post_like_count$

file fields (stat file - tsv format - rows are users):

sex: user specified sex

locale: user selected interface language

agerank: accounts ranked by creation date where 1 is youngest

like_count: number of user likes

post_count: number of user posts

post_like_count: number of likes on the user's posts

post_comment_count: number of comments on the user's posts

post_engagement_count: $post_comment_count$ and $post_like_count$ summed

Attention: data depends on your friends' privacy settings and the filtering choices you made for your newsf

5- Right click on "gdf file" and save it

netvizz v1.01

getting connections (42):

0

download

42 nodes, 2 edges

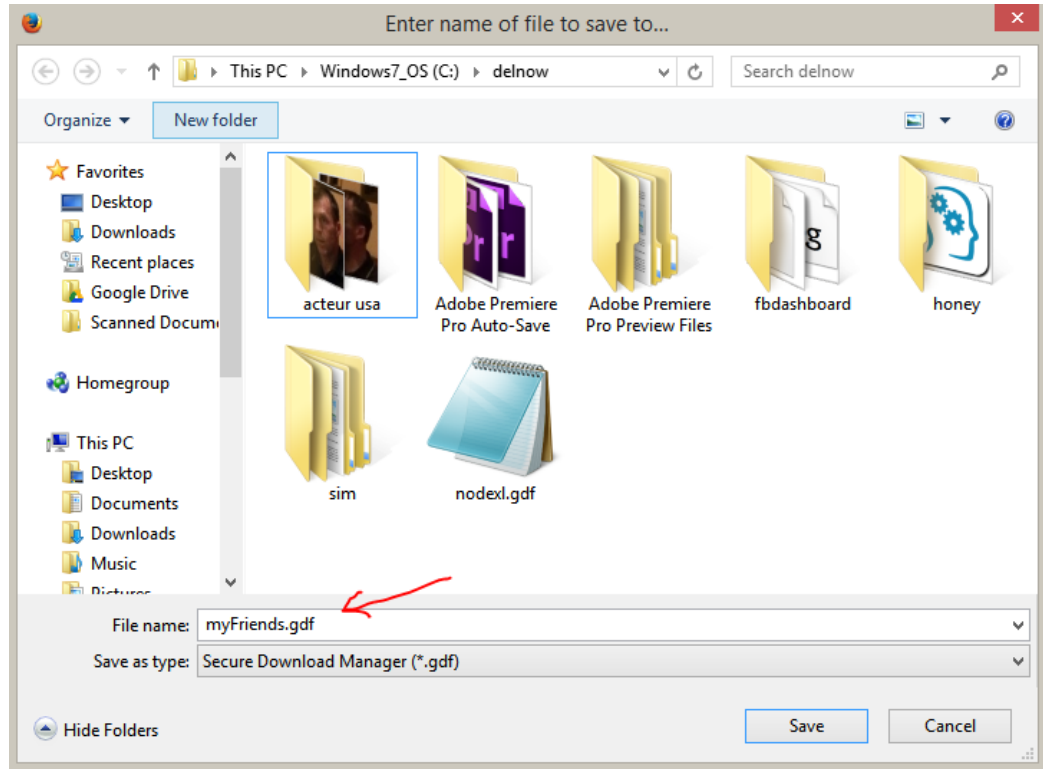
Your **gdf file** (right click, save as...).



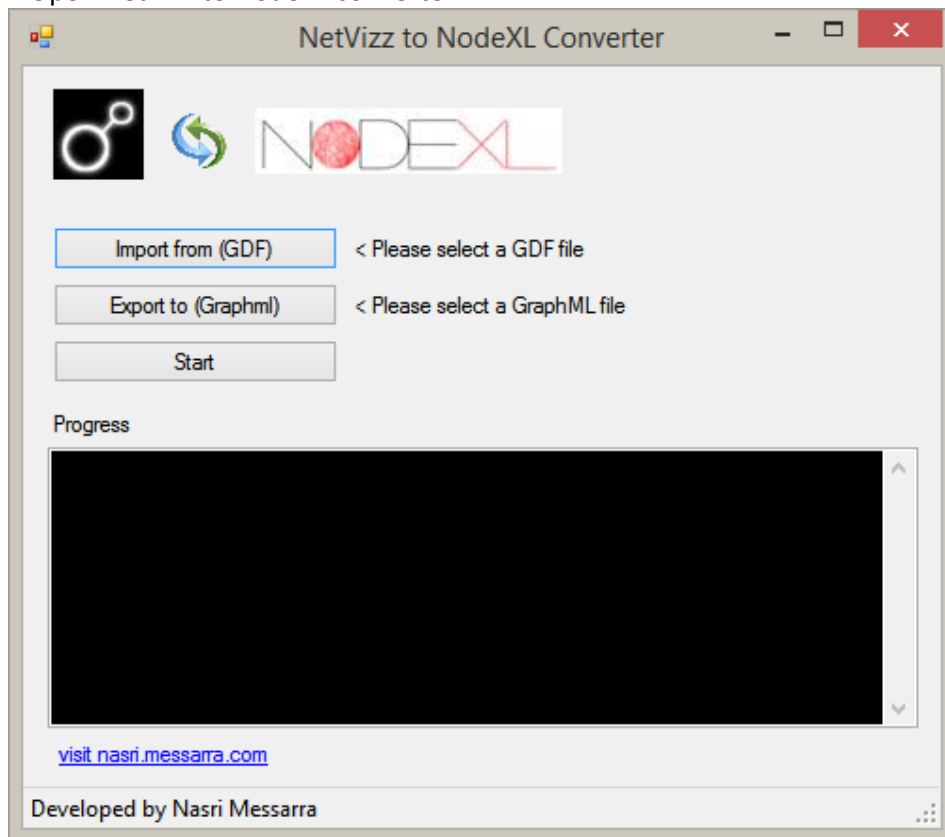
Your **tab file** (right click, save as...).

Attention: some browsers add a .txt extension !

6- Give a name to the file:

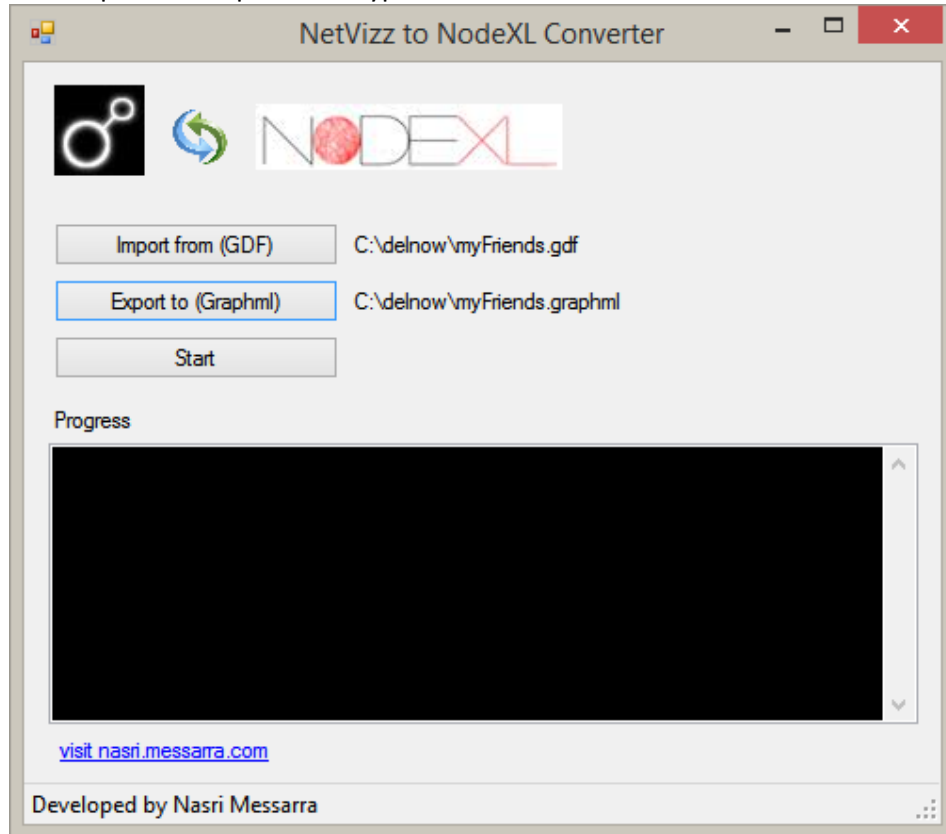


7- Open NetVizz to NodeXL converter

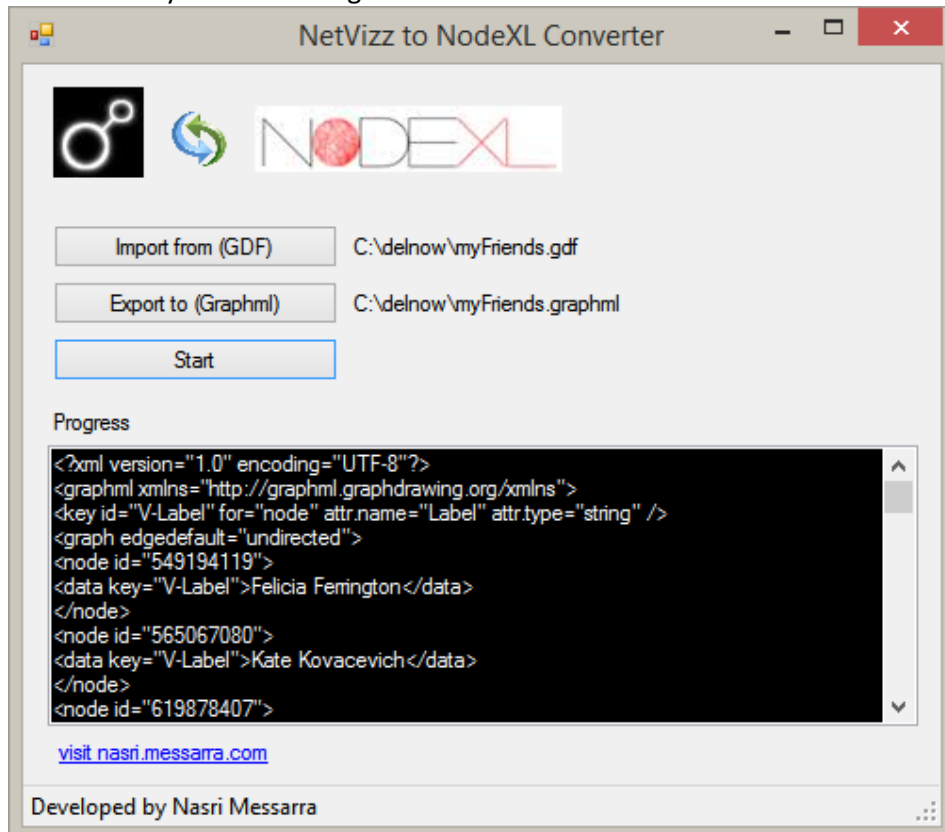


8- Click on import from GDF and select the file you've just saved

9- Click export to Graphml and type a file name

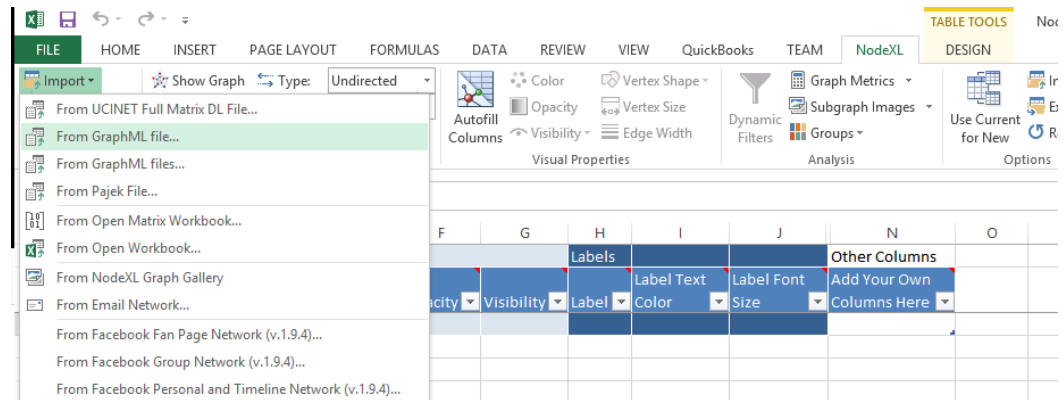


10- Make sure you have the right file names and click start



11- Open a new NodeXL file

12- From the NodeXL tab, select Import, select "from GraphML file" and select the file generated by NetVizz to NodeXL converter



13- Generate the Nodexl graph

You're done!