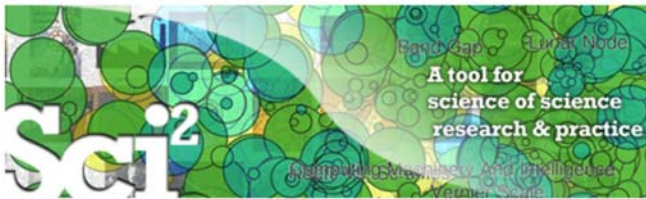


# Hierarchical Networks using Sci2 and OSLOM

**Ted Polley**

Research & Editorial Assistant  
Cyberinfrastructure for Network Science Center  
School of Library and Information Science  
[dapolley@indiana.edu](mailto:dapolley@indiana.edu)



Please download Sci2 at <http://sci2.cns.iu.edu>  
See documentation at <http://wiki.cns.iu.edu/display/SCI2TUTORIAL>

1

In this hands-on session we will introduce the Blondel community detection algorithm and the circular hierarchy network visualization. We will also look at the multifunctional algorithm package OSLOM ([www.oslom.org](http://www.oslom.org)) that handles edge directions, edge weights, overlapping communities, and hierarchies.

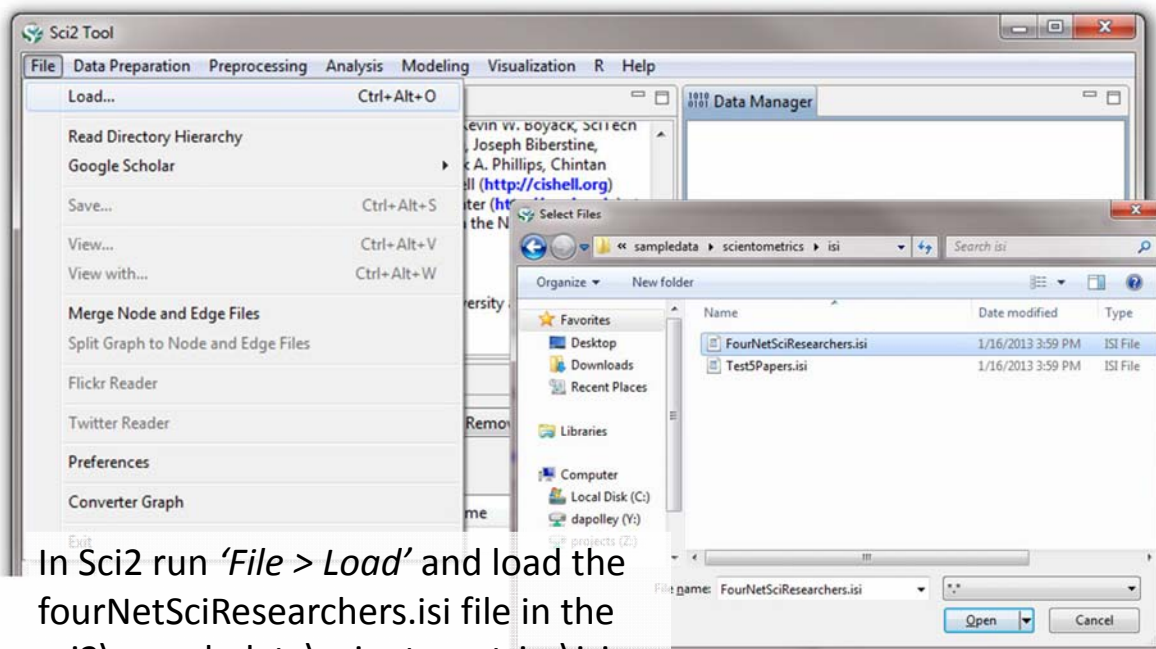
2

- Works based on modularity optimization
- Modularity is measured by looking at the density of edges inside communities versus the density of edges between communities
- Highly scalable – detecting communities in a 118 million node network took 152 minutes<sup>1</sup>

<sup>1</sup> Blondel, V.D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. Doi: [10.1088/1742-5468/2008/10/P10008](https://doi.org/10.1088/1742-5468/2008/10/P10008)

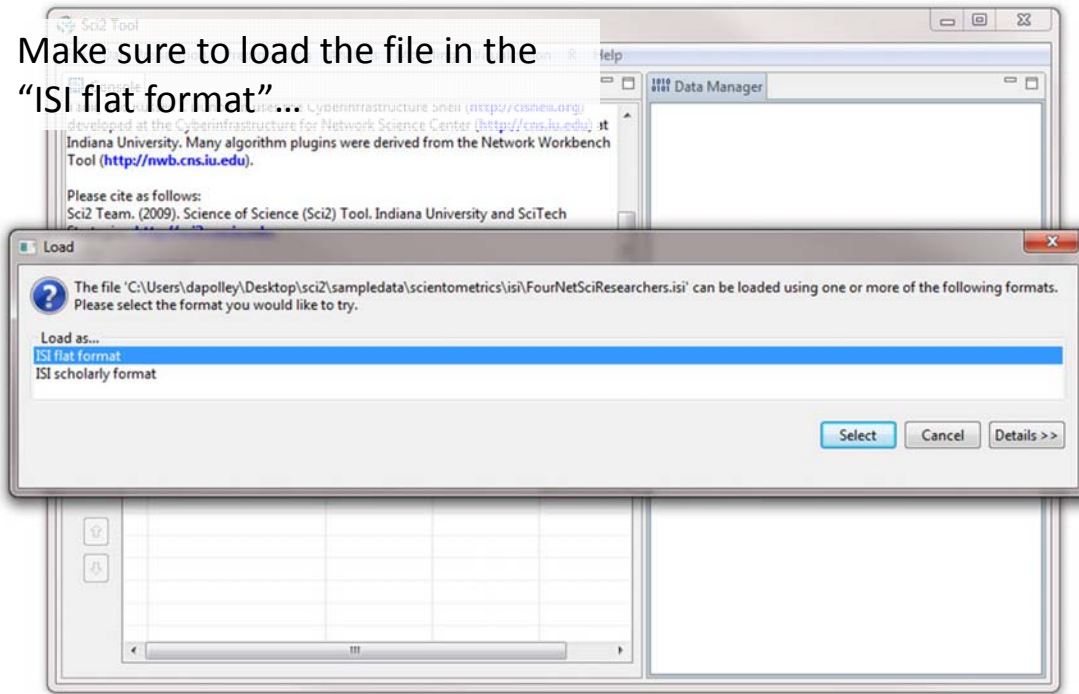
- Algorithm works in two phases
- First phase: each node is assigned to a community, modularity is calculated by examining each node and its neighbors, and this process repeats until all nodes are assigned to communities with the highest modularity possible
- Second phase: each of the communities identified in the first phase are treated as nodes in network
- This process iterates and the number of communities decrease with each level

Let's examine communities of authors in the fourNetSciResearchers.isi file. This dataset contains 361 publications spanning 52 years. It includes publications from four network scientists: Albert-László Barabási, Eugene Garfield, Alessandro Vespignani, & Stanley Wasserman. This data was collected in 2007 from Web of Science. In Sci2...

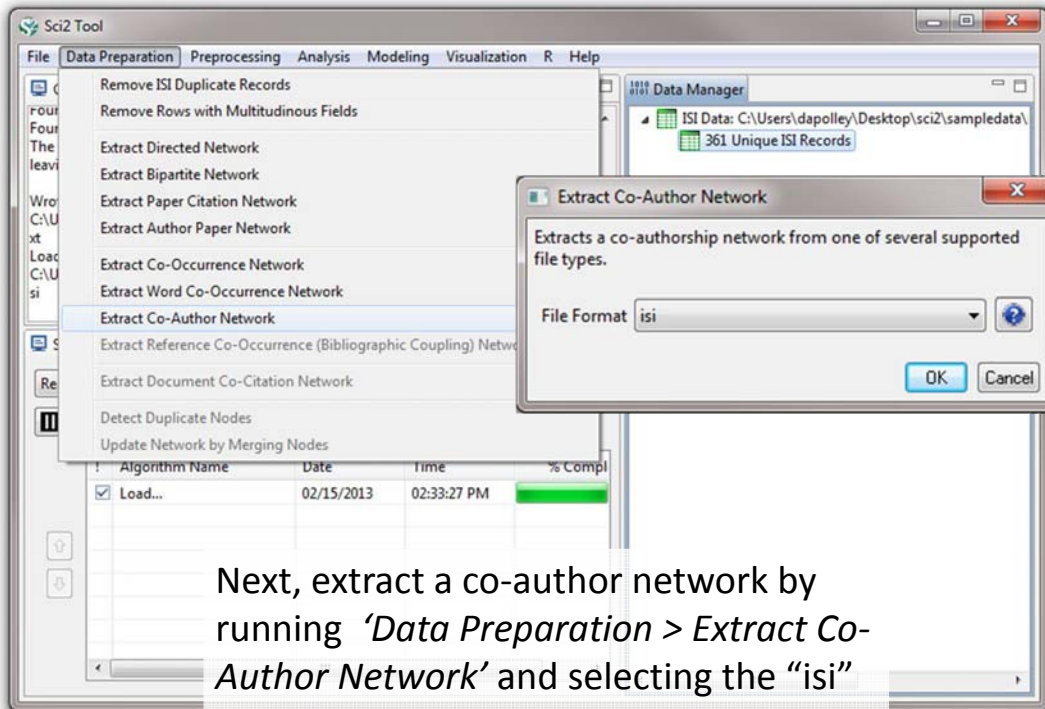


In Sci2 run 'File > Load' and load the fourNetSciResearchers.isi file in the sci2\sampladata\scientometrics\isi directory...

Make sure to load the file in the “ISI flat format”...

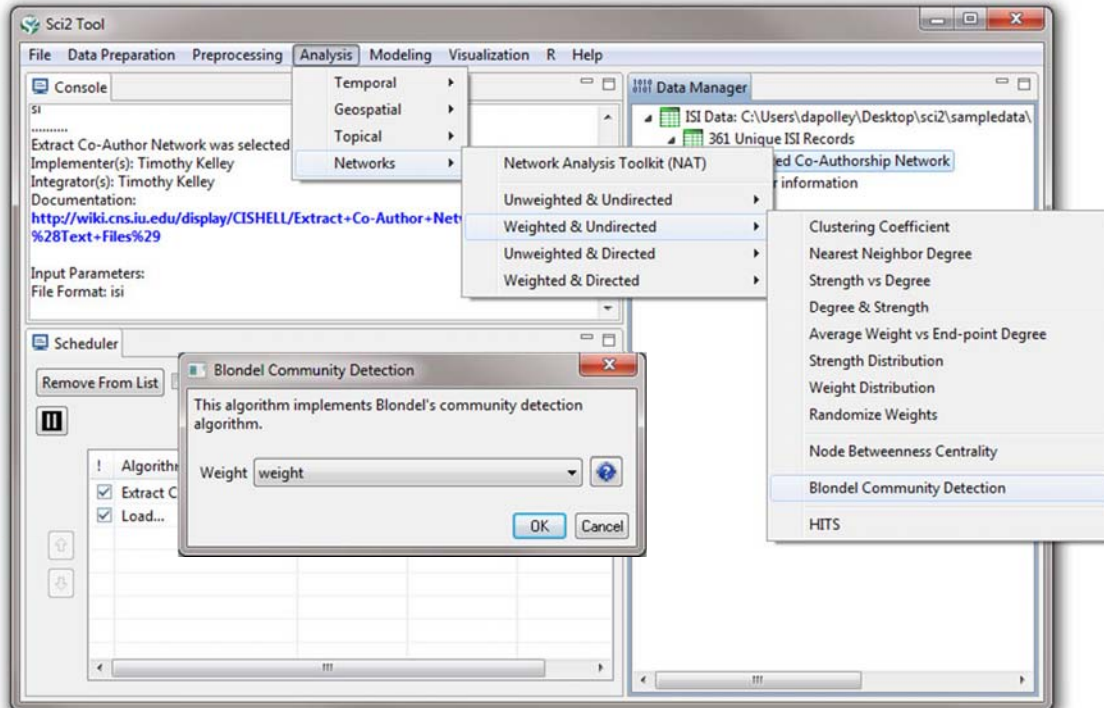


7

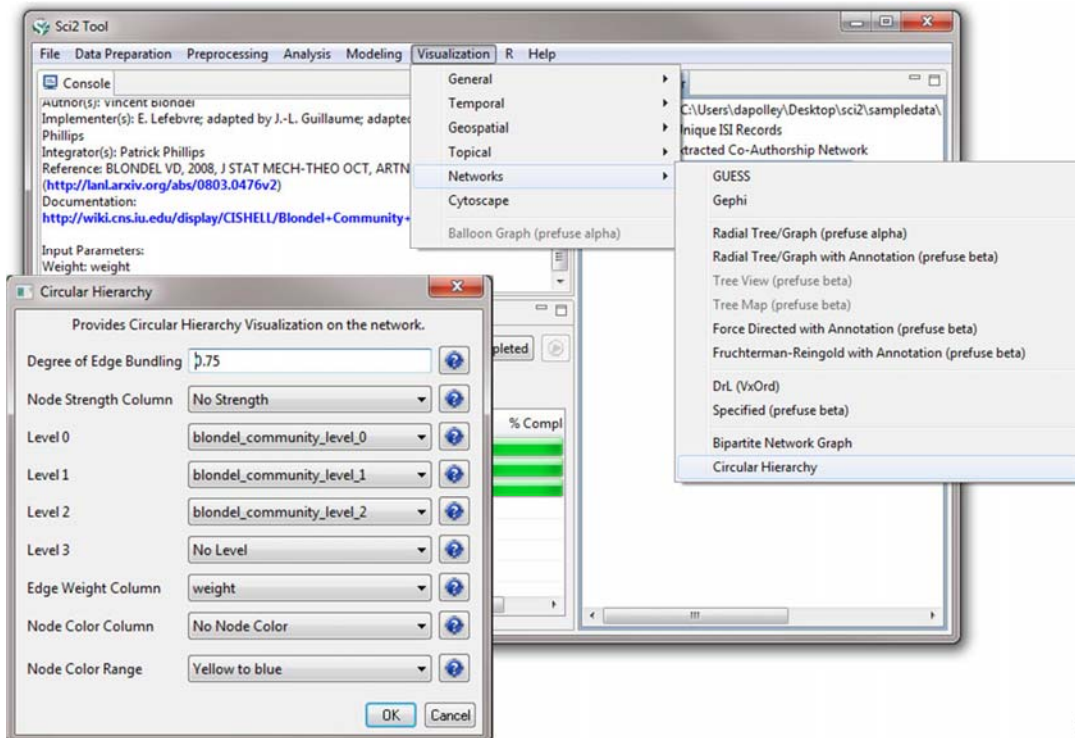


Next, extract a co-author network by running ‘Data Preparation > Extract Co-Author Network’ and selecting the “isi” file format

8



9



10



11

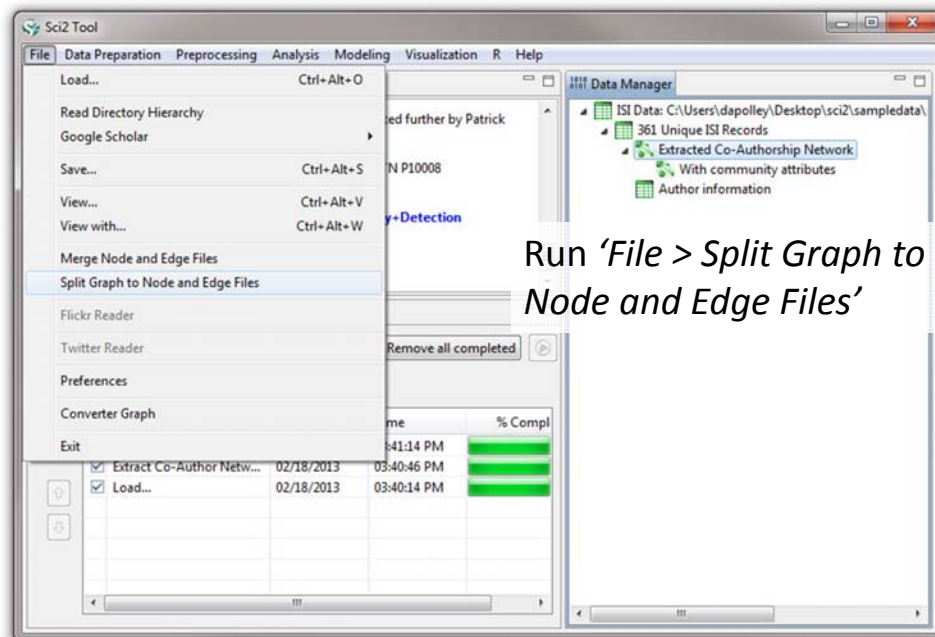
- Each ring around the network indicates a level in the hierarchy. In the pervious visualization there are three levels: community\_level\_0 (inner-most ring), community\_level\_1, and community\_level\_2 (outer-most ring)
- Each mark on the rings indicates a community and corresponds, in this case, to the authors names who belong in that community. You can see at community\_level\_0, there are 21 communities
- This structure allows you to see sub-communities within larger communities and conveys the hierarchical structure of this data

12

- OSLOM (Order Statistics Local Optimization Method)
- First method capable of handling directed networks, edge weights, overlapping communities, hierarchical data structure, and community dynamics
- Get OLSOM here <http://olsom.org>. You will have to extract the directory from the OLSOM2.tar.gz
- You will have to compile the OSLOM code in a Unix (MAC) terminal (if you are using Windows, you can try to use [MinGW](#), Minimalist GNU for Windows)

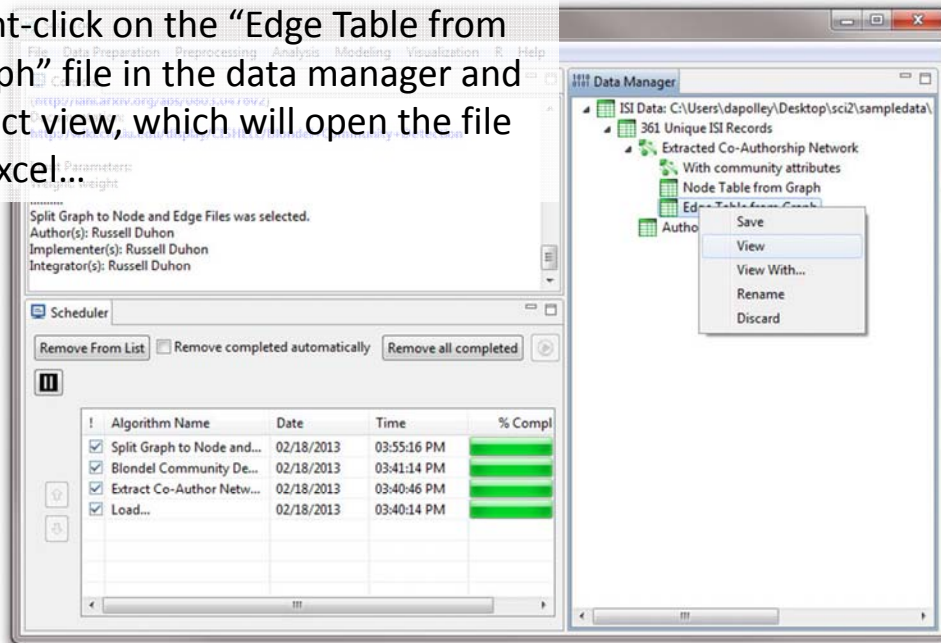
13

In order to get data into a format to be processed by OSLOM, you need to save out the edge list from Sci2



14

Right-click on the “Edge Table from Graph” file in the data manager and select view, which will open the file in Excel...



15

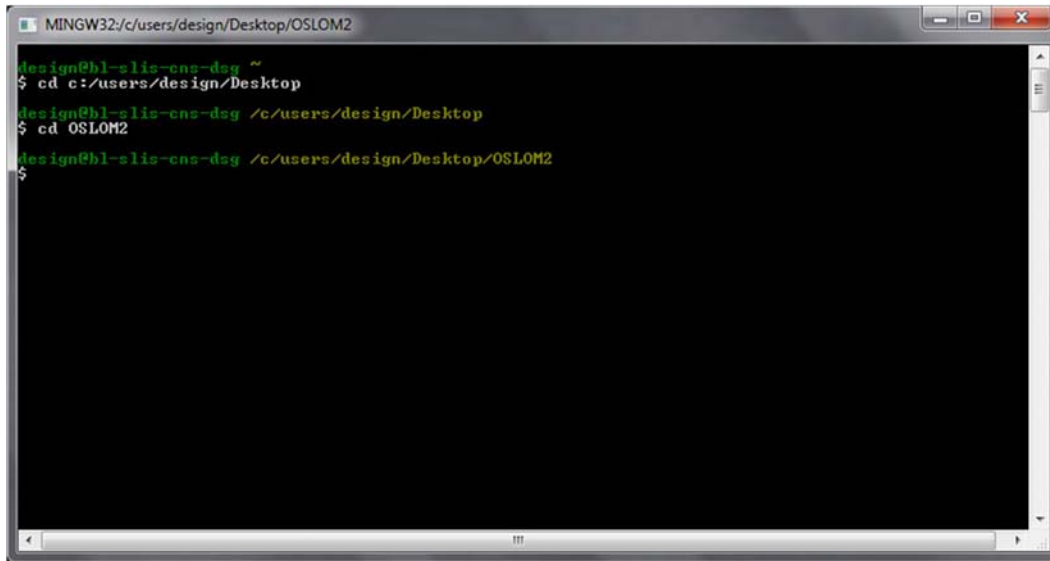
	A	B	C	D	E
1	source	target	number_of_coauthored_works	weight	
2	0	1	1	1	
3	0	2	1	1	
4	1	2	2	2	
5	3	4	4	4	
6	2	3	17	17	
7	2	4	4	4	
8	2	5	1	1	
9	2	6	2	2	
10	5	6	1	1	
11	2	7	2	2	
12	2	8	1	1	
13	7	9	1	1	
14	2	10	6	6	
15	2	11	1	1	
16	9	12	1	1	
17	2	13	1	1	
18	2	14	2	2	
19	2	15	1	1	
20	11	16	1	1	
21	12	17	1	1	
22	11	18	1	1	
23	2	19	2	2	
24	2	20	1	1	
25	2	21	1	1	
26	3	22	1	1	
27	2	23	1	1	

You will need to remove the “number\_of\_coauthored\_works” column since it is the same as the weight column. Then you should remove the top row because OSLOM will not be able to parse these string values. You should save the file as a text (.txt) file in you OLSOM2 directory. Before you can use it with OSLOM you will need to change the .txt extension to .dat

16



First, you will need to move into the OSLOM directory, this can be done by using the change directory command (cd) and entering the path to the OSLOM2 folder. Note, here it has been done in two steps here, but it can be done in one.

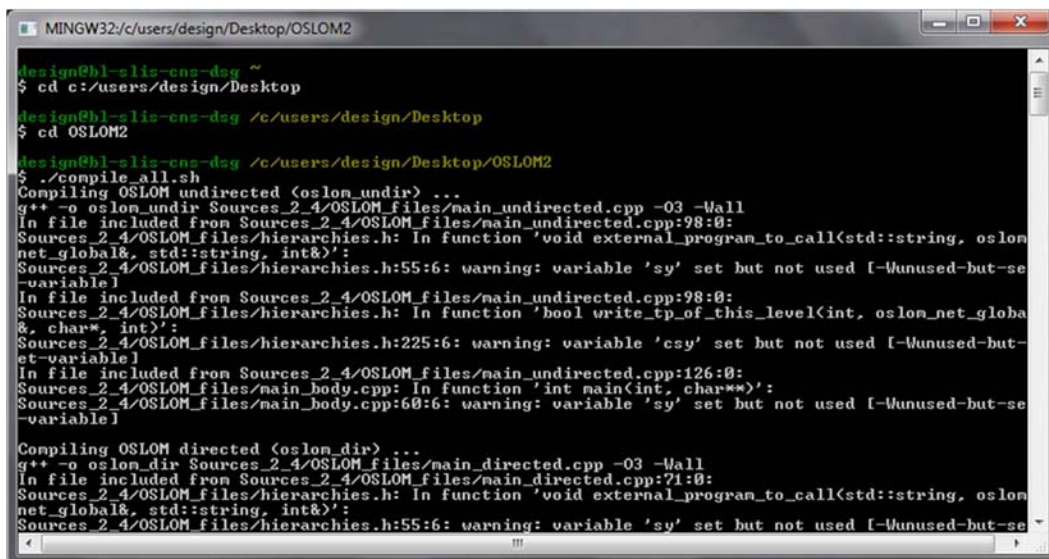


```

MINGW32:/c/users/design/Desktop/OSLOM2
design@phl-slis-cns-dog ~
$ cd c:/users/design/Desktop
design@phl-slis-cns-dog /c/users/design/Desktop
$ cd OSLOM2
design@phl-slis-cns-dog /c/users/design/Desktop/OSLOM2
$
    
```

17

Next, you need to compile the OSLOM code. Type in:  
./compile\_all.sh



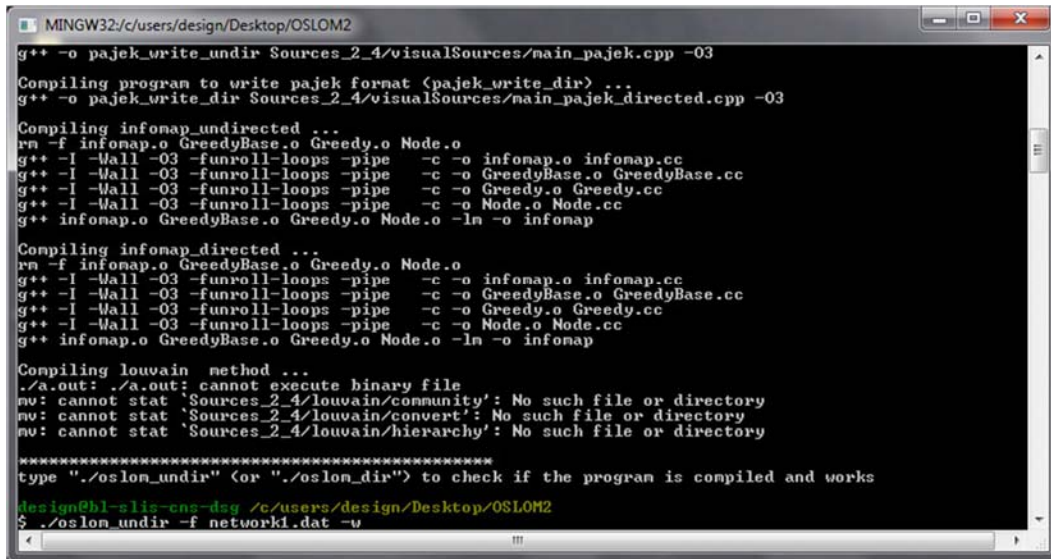
```

MINGW32:/c/users/design/Desktop/OSLOM2
design@phl-slis-cns-dog ~
$ cd c:/users/design/Desktop
design@phl-slis-cns-dog /c/users/design/Desktop
$ cd OSLOM2
design@phl-slis-cns-dog /c/users/design/Desktop/OSLOM2
$ ./compile_all.sh
Compiling OSLOM undirected <oslom_undir> ...
g++ -o oslom_undir Sources_2_4/OSLOM_files/main_undirected.cpp -O3 -Wall
In file included from Sources_2_4/OSLOM_files/main_undirected.cpp:98:0:
Sources_2_4/OSLOM_files/hierarchies.h: In function 'void external_program_to_call(std::string, oslom_net_global&, std::string, int&)':
Sources_2_4/OSLOM_files/hierarchies.h:55:6: warning: variable 'sy' set but not used [-Wunused-but-set-variable]
In file included from Sources_2_4/OSLOM_files/main_undirected.cpp:98:0:
Sources_2_4/OSLOM_files/hierarchies.h: In function 'bool write_tp_of_this_level(int, oslom_net_global&, char**, int)':
Sources_2_4/OSLOM_files/hierarchies.h:225:6: warning: variable 'csy' set but not used [-Wunused-but-set-variable]
In file included from Sources_2_4/OSLOM_files/main_undirected.cpp:126:0:
Sources_2_4/OSLOM_files/main_body.cpp: In function 'int main(int, char**)':
Sources_2_4/OSLOM_files/main_body.cpp:60:6: warning: variable 'sy' set but not used [-Wunused-but-set-variable]
Compiling OSLOM directed <oslom_dir> ...
g++ -o oslom_dir Sources_2_4/OSLOM_files/main_directed.cpp -O3 -Wall
In file included from Sources_2_4/OSLOM_files/main_directed.cpp:71:0:
Sources_2_4/OSLOM_files/hierarchies.h: In function 'void external_program_to_call(std::string, oslom_net_global&, std::string, int&)':
Sources_2_4/OSLOM_files/hierarchies.h:55:6: warning: variable 'sy' set but not used [-Wunused-but-set-variable]
    
```

18

Now you need to run OSLOM on the network you saved in the OSLOM2 folder from Sci2. Type the following command:

`./oslom_undir -f <whatever you named the file.dat> -w`



```

MINGW32:/c/users/design/Desktop/OSLOM2
g++ -o pajek_write_undir Sources_2_4/visualSources/main_pajek.cpp -O3
Compiling program to write pajek format (pajek_write_dir) ...
g++ -o pajek_write_dir Sources_2_4/visualSources/main_pajek_directed.cpp -O3

Compiling infomap_undirected ...
rm -f infomap.o GreedyBase.o Greedy.o Node.o
g++ -I -Wall -O3 -funroll-loops -pipe -c -o infomap.o infomap.cc
g++ -I -Wall -O3 -funroll-loops -pipe -c -o GreedyBase.o GreedyBase.cc
g++ -I -Wall -O3 -funroll-loops -pipe -c -o Greedy.o Greedy.cc
g++ -I -Wall -O3 -funroll-loops -pipe -c -o Node.o Node.cc
g++ infomap.o GreedyBase.o Greedy.o Node.o -ln -o infomap

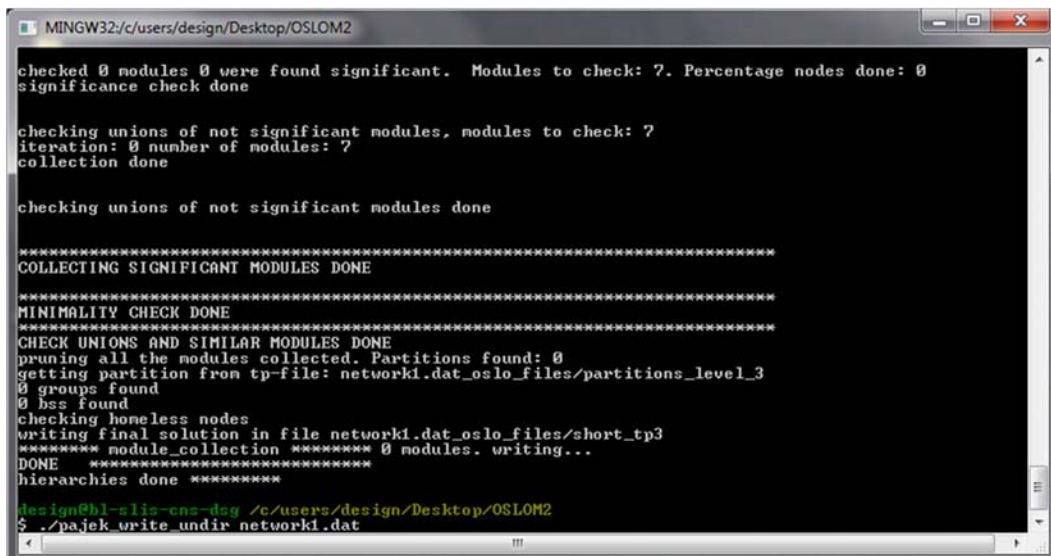
Compiling infomap_directed ...
rm -f infomap.o GreedyBase.o Greedy.o Node.o
g++ -I -Wall -O3 -funroll-loops -pipe -c -o infomap.o infomap.cc
g++ -I -Wall -O3 -funroll-loops -pipe -c -o GreedyBase.o GreedyBase.cc
g++ -I -Wall -O3 -funroll-loops -pipe -c -o Greedy.o Greedy.cc
g++ -I -Wall -O3 -funroll-loops -pipe -c -o Node.o Node.cc
g++ infomap.o GreedyBase.o Greedy.o Node.o -ln -o infomap

Compiling louvain_method ...
./a.out: ./a.out: cannot execute binary file
mv: cannot stat 'Sources_2_4/louvain/community': No such file or directory
mv: cannot stat 'Sources_2_4/louvain/convert': No such file or directory
mv: cannot stat 'Sources_2_4/louvain/hierarchy': No such file or directory
*****
type "./oslom_undir" (or "./oslom_dir") to check if the program is compiled and works
design@hl-slis-cns-dsg /c/users/design/Desktop/OSLOM2
$ ./oslom_undir -f network1.dat -w
  
```

19

Next, you will need to write a Pajek file for network after the communities have been identified. Type in the following command:

`./pajek_write_undir <whatever you named the file.dat>`



```

MINGW32:/c/users/design/Desktop/OSLOM2
checked 0 modules 0 were found significant. Modules to check: 7. Percentage nodes done: 0
significance check done

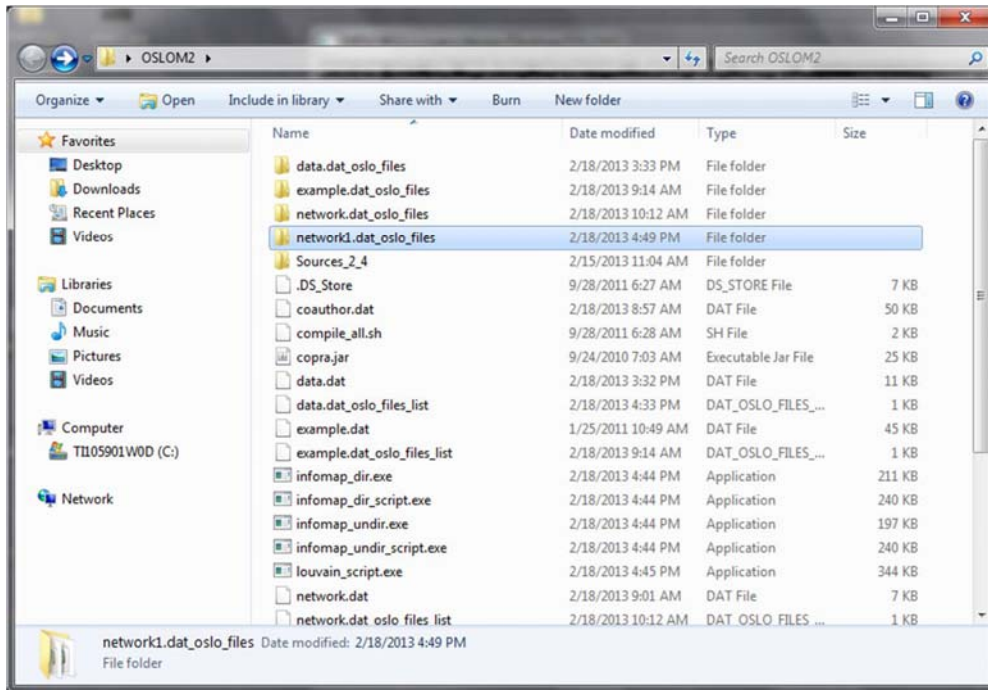
checking unions of not significant modules, modules to check: 7
iteration: 0 number of modules: 7
collection done

checking unions of not significant modules done

*****
COLLECTING SIGNIFICANT MODULES DONE
*****
MINIMALITY CHECK DONE
*****
CHECK UNIONS AND SIMILAR MODULES DONE
pruning all the modules collected. Partitions found: 0
getting partition from tp-file: network1.dat_oslo_files/partitions_level_3
0 groups found
0 hss found
checking homeless nodes
writing final solution in file network1.dat_oslo_files/short_tp3
***** module_collection ***** 0 modules. writing...
DONE *****
hierarchies done *****
design@hl-slis-cns-dsg /c/users/design/Desktop/OSLOM2
$ ./pajek_write_undir network1.dat
  
```

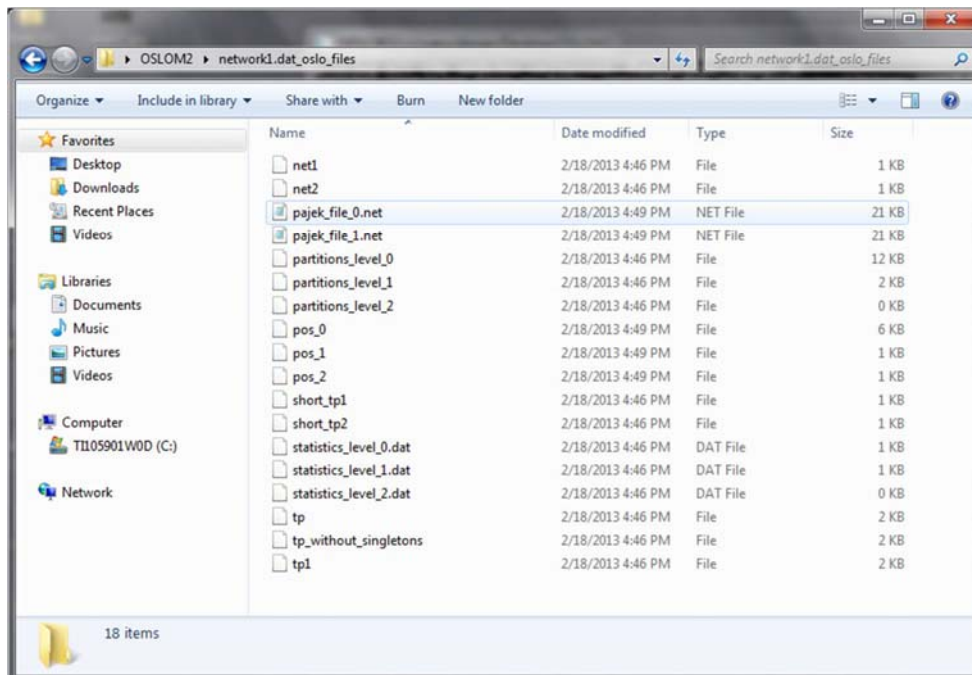
20

A folder will be created in the OSLOM2 folder:



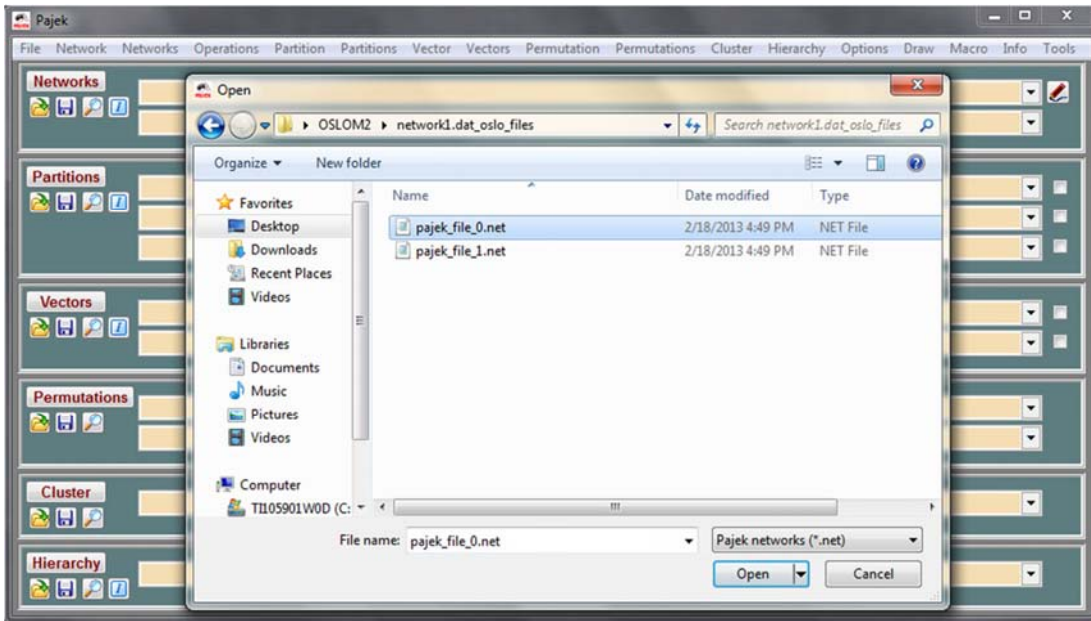
21

In this folder you will find the Pajek files, one for each level of the hierarchy in the network:



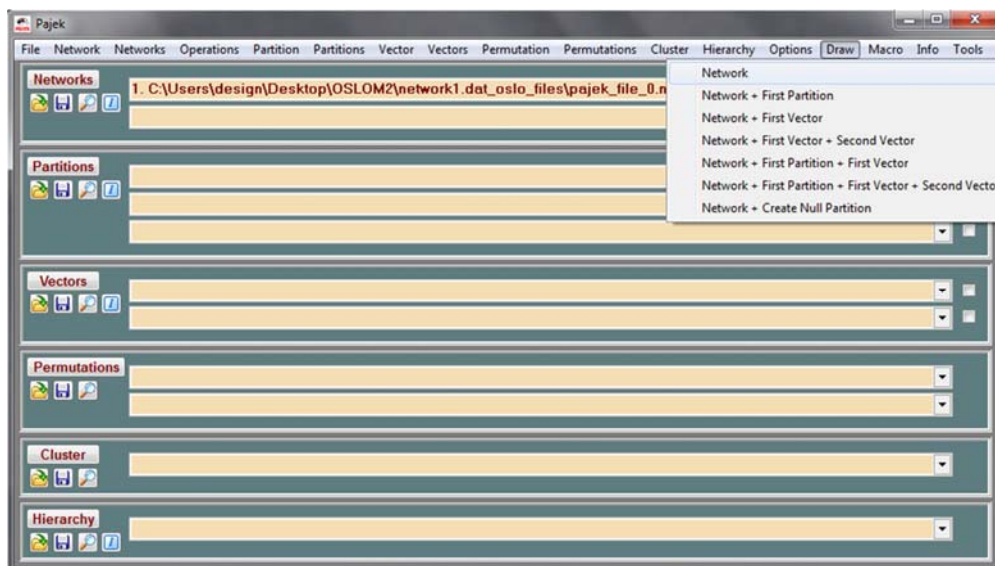
22

Load this file in Pajek (you can also load it in Gephi):



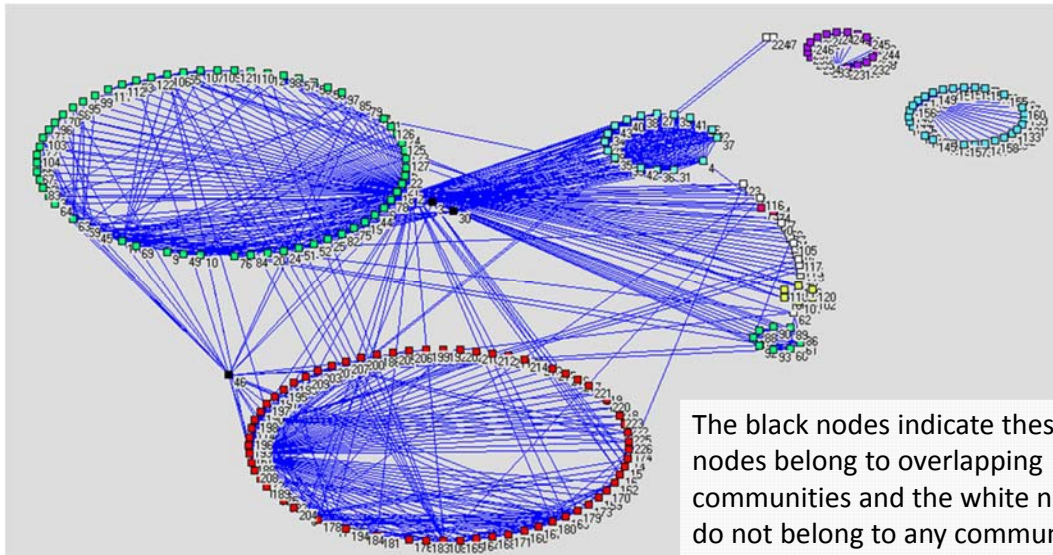
23

Then to draw the network, run *'Draw > Network'*



24

The network should automatically layout like this. To see the color go to 'Options > Colors > Vertices > As Defined on Input File'



The black nodes indicate these nodes belong to overlapping communities and the white nodes do not belong to any community

### OSLOM Paper:

Lancichinetti, A., Radicchi, F., Ramasco, J.J., & Fortunato, S. (2010). Finding statistically significant communities in networks. *PLoS One*, 6(4), e18961. Doi: [10.1371/journal.pone.0018961](https://doi.org/10.1371/journal.pone.0018961)

### Blondel Paper:

Blondel, V.D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. Doi: [10.1088/1742-5468/2008/10/P10008](https://doi.org/10.1088/1742-5468/2008/10/P10008)



**CNS**

Cyberinfrastructure for  
Network Science Center

Questions?