



# Visualizing large datasets in the Semantic Web



#VizLinkedData

Feb 2025

 @marienorico

mariano.rico@upm.es

# Attendance check

# Get the material

- Slides and files can be downloaded from

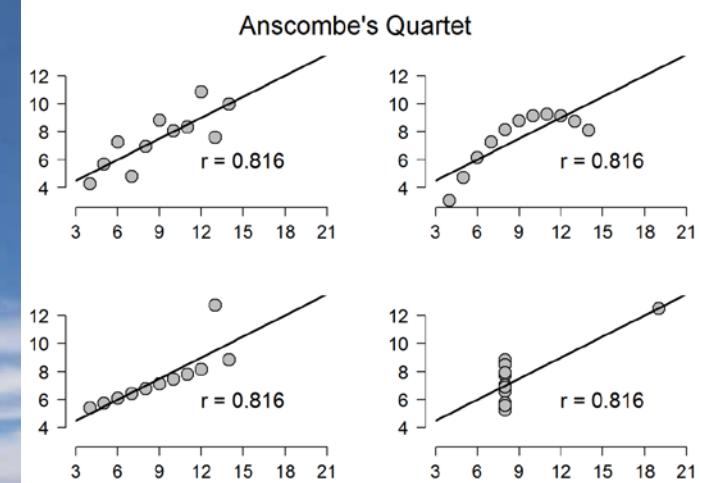
<https://tinyurl.com/bigdatavizMRA>

# Why visualizing?

- Numerical analysis can miss info. A graph can get that info
  - Classical example: [Anscombe's Quartet](#) (1973)

For all four datasets:

Property	Value	Accuracy
Mean of $x$	9	exact
Sample variance of $x : s_x^2$	11	exact
Mean of $y$	7.50	to 2 decimal places
Sample variance of $y : s_y^2$	4.125	$\pm 0.003$
Correlation between $x$ and $y$	0.816	to 3 decimal places
Linear regression line	$y = 3.00 + 0.500x$	to 2 and 3 decimal places, respectively
Coefficient of determination of the linear regression : $R^2$	0.67	to 2 decimal places



# Why visualizing?

- Numerical analysis can miss info. A graph can get that info
  - Modern example: [The Datasaurus Dozen](#) (A. Cairo 2017)

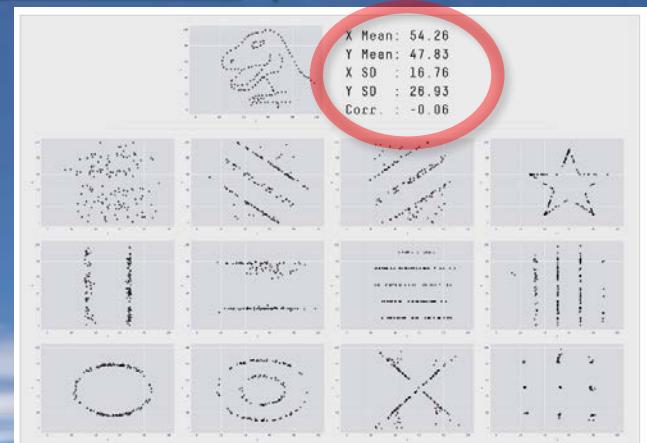
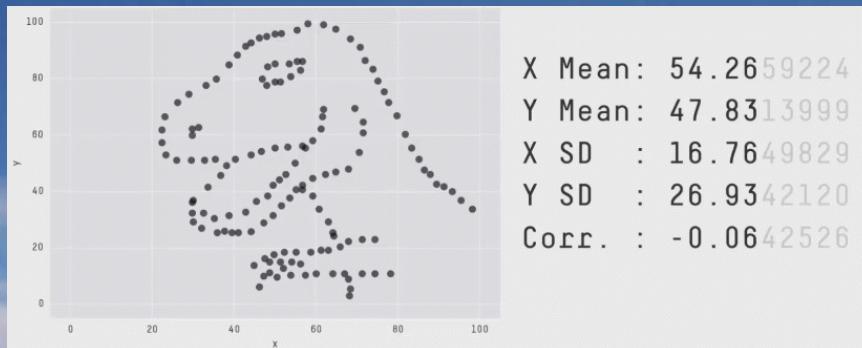
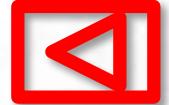


Fig 2. The Datasaurus Dozen. While different in appearance, each dataset has the same summary statistics (mean, standard deviation, and Pearson's correlation) to two decimal places.

# Content

- [Large-scale graph analytics](#)
- [Visualizing big graphs](#)
- [The LOD cloud as a big data use case](#)
- [Big graphs with Gephi. An intro](#)
  - [Gephi Layouts](#)
  - [Gephi + DBpedia](#)
  - [Gephi + Wikidata](#)
  - [Gephi + RDF/OWL](#)
  - [Gephi + R](#)
- [Igraph](#)
- Other tools
  - [Loupe](#). Explore datasets
  - [RelFinder](#). A first graphical approach
  - [Gruff](#). Graph SPARQL query results



Viewing **without** visualizing

# LARGE-SCALE GRAPH ANALYTICS

# Problem

- Huge graphs
  - Facebook: 1 billion users → 1G nodes (aka vertex) graph
  - Google: the indexed web is a ~55 billion pages graph (Common Crawl has 25G as of April 2020)
  - Wayback Machine (Internet Archive) has ~654 billion pages (as of Feb. 2022)
- Complex/heavy algorithms
  - Page rank
  - Shortest path in a graph

# State of the art

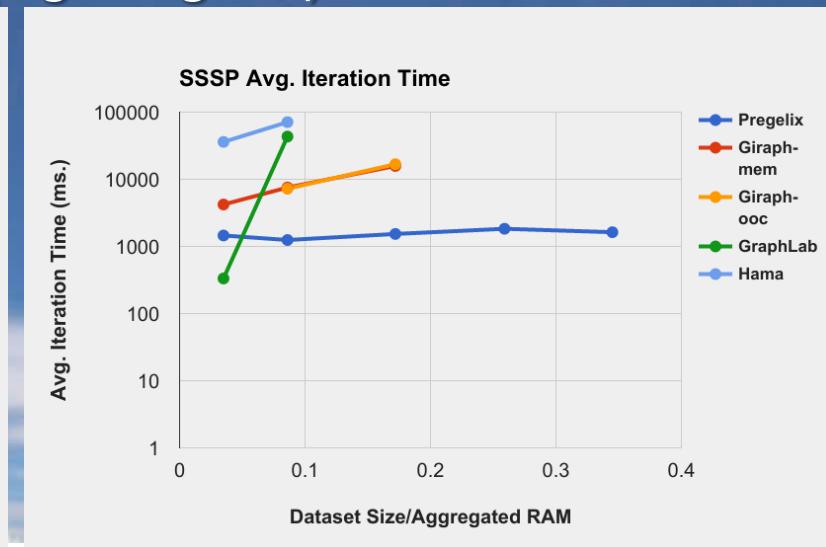
- Pregelix
  - Open source (Java)
  - Distributed graph processing system
    - Designed to handle resources efficiently (memory and disc)

# Pregelix

- Comparison with other Big Graph Analytics platforms (Apache [Giraph](#), Apache [GraphX](#), [GraphLab](#))
  - Specific problem: Single-Source Shortest Paths (SSSP) problem
    - Find the shortest paths from a source vertex to all other vertices in the graph.
      - [Dijkstra's algorithm](#) was the first successful algorithm
  - Configuration: 32 machine cluster

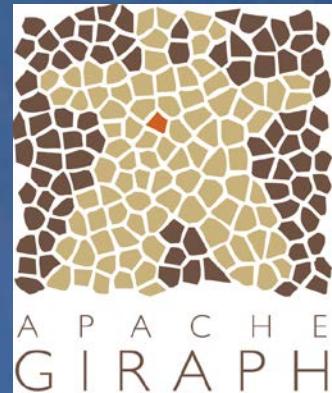
# Pregelix

- Performance results (over a graph with 0.7B nodes and 6B edges)
  - end-to-end execution time (left figure)
  - average iteration time (right figure)



# Apache Giraph

- Apache project (open source)
  - Java
- Exploits Apache  **hadoop**
- Based on Google's Pregel
  - Pregel is C++
- Efficient. Used by Facebook. See [here](#)
- Based on the **BSP method**
  - BSP = Bulk Synchronous Parallel



# BSP method

- Created in 80's by Leslie Valiant (Stanford U.)
  - See Leslie G. Valiant, "A bridging model for parallel computation", Communications of the ACM, Volume 33 Issue 8, Aug. 1990.
- Used by Pregel (Google) and Giraph (Apache)

# BSP method

- The method
  - Algorithms as series of iterations (aka “supersteps”)
  - Vertex oriented. Each vertex in the graph:
    - Calls a method (in parallel)
    - Sends messages to the first-neighbors vertex
      - Will be read in the next superstep
    - Can read messages sent in the previous superstep
    - Can modify the state (create/delete) of the vertex edges

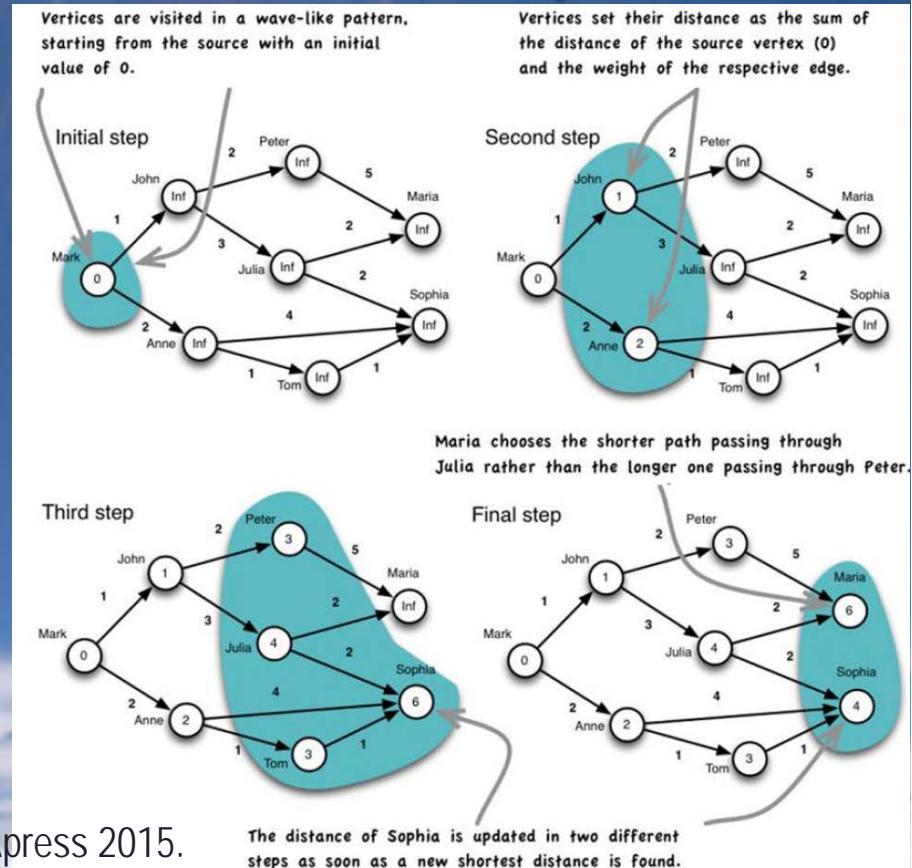
# BSP method. An example

- Maximum age in a social network
  - Algorithm for BSP: In each iteration:
    - Each vertex sets its value to the maximum between its current value and the values of its (first) neighbors.
    - Check end of iterations: if none vertex updated its value, then finish iterations



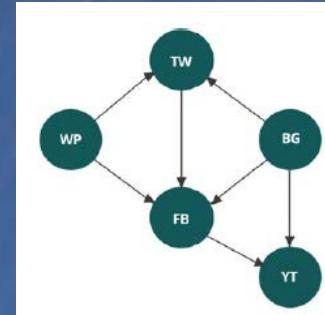
# BSP method. SSSP example

- SSSP algorithm for BSP.  
In each iteration:
  - Each vertex sets its distance value as the sum of its in-vertices + weight of the edge.
  - In case of several in-vertices, select the minimum
  - Check iterations end



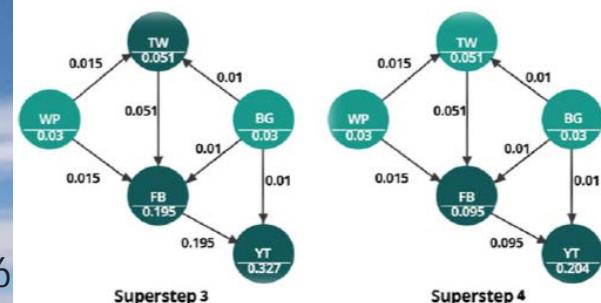
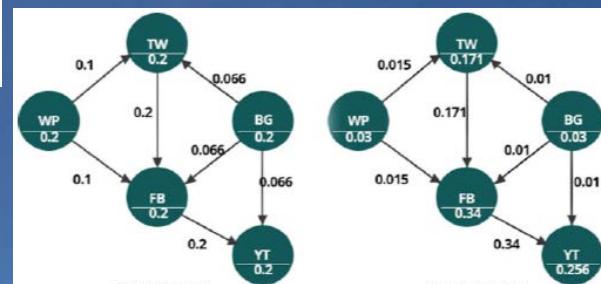
# BSP method. PageRank example

- PageRank algorithm for BSP.
  - Initialization
    - Assign  $0.2$  ( $1/\# \text{vertices}$ ) to each vertex.
    - Message sent:  $\text{val} / \# \text{outEdges}$
  - In each iteration
    - Each vertex sums the incoming messages ( $\text{im}$ ) \*  $0.85$  and sums  $0.15/\# \text{vertices}$  (that is, computes pageRank value)
    - Check end of iterations



Initial state

Evolution



Source: Large scale graph porcessing using Apache Giraph. Springer 2016

# BSP method. PageRank example

- Giraph Java code

[Github source](#)

```
public class PageRankVertexComputation extends
    BasicComputation<LongWritable, DoubleWritable, NullWritable, DoubleWritable> {

    public static final int MAX_SUPERSTEPS = 5; #ad hoc value for this example

    @Override
    public void compute(Vertex<LongWritable, DoubleWritable,
        NullWritable> vertex,
        Iterable<DoubleWritable> messages) throws IOException {
        if (getSuperstep() >= 1) {
            double sum = 0;
            for (DoubleWritable message : messages) {
                sum += message.get();
            }
            DoubleWritable vertexValue = new
                DoubleWritable( 0.15f / getTotalNumVertices() + 0.85f * sum);
            vertex.setValue(vertexValue);
        }
        if (getSuperstep() < MAX_SUPERSTEPS) {
            long edges = vertex.getNumEdges();
            sendMessageToAllEdges(vertex,
                new DoubleWritable(vertex.getValue().get() / edges));
        }
        else {
            vertex.voteToHalt();
        }
    }
}
```

# Algorithms complexity

- Comparison by tool

Table 4. Algorithmic Time Complexities

FEATURES	NETWORKX	IGRAPH	GEPHI	PAJEK
Isomorphism	$O(N^2)$	exp	Na	Na
CORE M M=No. Of Lines	$O(M)$	$O(M)$	$O(M)$	$O(M)$
Cliques	$O( V /(\log 2))$	$O(3 V /3)$	Na	$O(N)$
Shortest Path	$O( V . E )$	$O( V + E )$	$O( V + E )$	$O( V + E )$
Clustering	$O(V)$	Na	$O(V)$	Na
All Simple Path	$O( V + E )$	$O( V + E )$	Na	Na
Closeness Centrality	$O(N. E )$	$O(N. E )$	Na	Na
Density	$O(N^3)$	$O(1)$	Na	Na
MST	Na	$O( V + E )$	Na	Na
Cycles	$O(( V + E ).C+1)$	Na	Na	Na
PageRank	Na	$O(E)$	$O(E)$	Na
Betweenness	Na	$O( V . E )$	Na	Na
Eigenvector	Na	$O( V + E )$	Na	Na



Graph Theoretic Approaches for  
Analyzing Large-Scale Social  
Networks

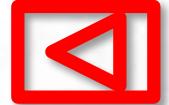
# Layouts

- Layouts supported comparison by tool

*Table 3. Comparison Based on Graph Layout Supported by Tools*

Layout	NETWORKX	IGRAPH	GEPHI	PAJEK
Circular Layout	Yes	Yes	Yes	Yes
Random Layout	Yes	Yes	Yes	No
Spectral Layout	Yes	No	No	No
Spring Layout	Yes	Yes	Yes	Yes
Graphviz Layout	Yes	No	No	No
Kamada Kawai	No	Yes	Yes	No
Fruchterman Reingold	No	Yes	Yes	No
Force Atlas Layout	No	No	Yes	No





Finally!: viewing visualizing

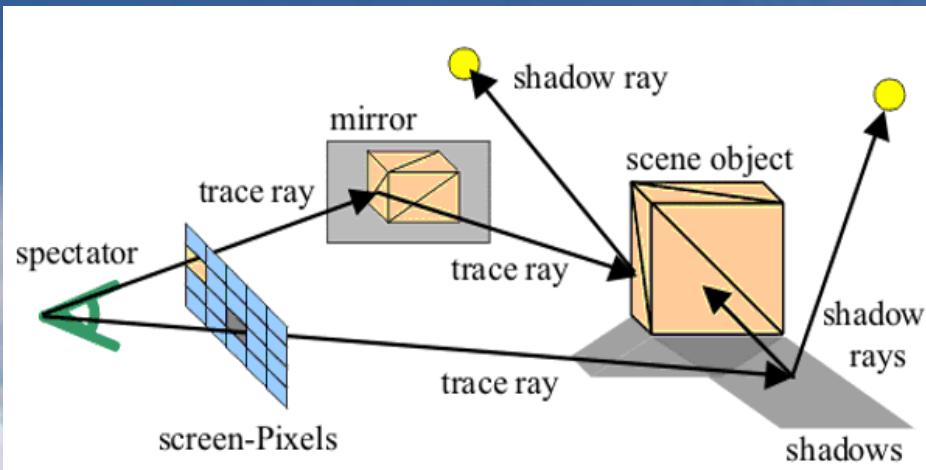
# VISUALIZING BIG GRAPHS

# Display limits

- Imagine a 100 billion star galaxy
- Can you see all the stars? [Look for “Andromeda”](#)
- Sure?
  - What is the screen resolution of your
    - Monitor? 4K = 8.3Mpixel in 16x9
    - Eyes? ~600Mpixel [Source](#)

# Display limits

- It is like ray-tracing
  - Rendering is not a big problem
  - The problem is to make all the computation

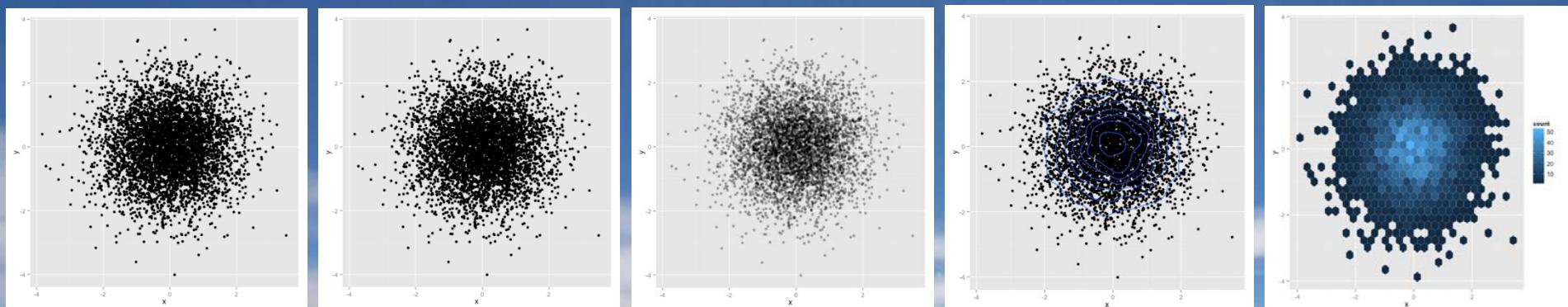


[Source](#)



# How to display

- Sometimes the problem is how to display the data
- Example: The “all black” case.
  - Use: jitter, alpha, density contour... hexbins!!



R Source (using ggplot2)

# How to display

- For Spark
  - GraphX
    - Pregel API
    - Fast, but non interactive
    - Doc
      - API for
        - » Java
        - » Scala
        - » Python
        - » R (paquetes [SparkR](#) (Apache), [sparklyr](#) (RStudio), o [graphframes](#) (extension de sparklyr de RStudio))

The screenshot shows the official Apache Spark GraphX website. At the top, there's a navigation bar with links for Download, Libraries, Documentation, Examples, Community, Developers, and Apache Software Foundation. Below the navigation is a main heading "APACHE Spark™ GraphX". A sub-section title "GraphX is Apache Spark's API for graphs and graph-parallel computation." is followed by a "Flexibility" section which says "Seamlessly work with both graphs and collections". Below this, a paragraph explains that GraphX unifies ETL, exploratory analysis, and iterative graph computation within a single system. It mentions viewing data as graphs and collections, transforming and joining graphs with RDDs, and writing iterative graph algorithms using the Pregel API. To the right of this text is a bar chart titled "Runtime (s)" comparing three systems: GraphLab, GraphX, and Graph. The chart shows GraphX with a runtime of 579 seconds, Graph with 1235 seconds, and GraphLab with 833 seconds. A snippet of Scala code is shown on the right side of the chart. On the far right, there's a "Latest News" sidebar with links to news items about Spark 2.1.2, Spark Summit Europe, Spark 2.2.0, and Spark 2.1.1, along with an "Archive" link. A large green "Download Spark" button is located at the bottom right.

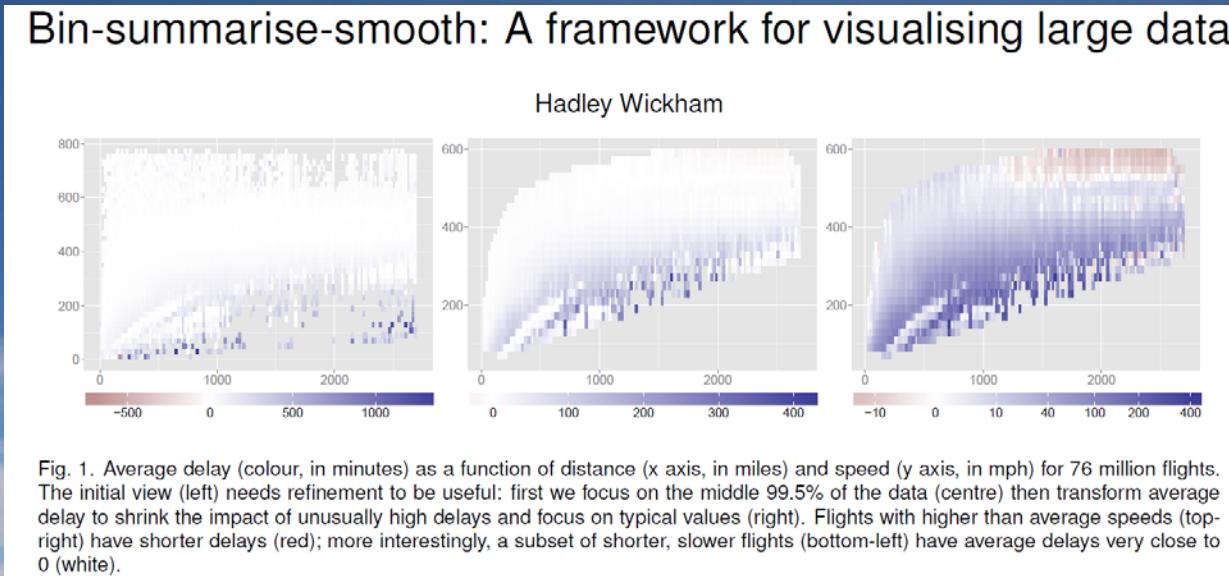
# How to display

- For R
  - bigvis package (by RStudio, not yet in CRAN despite started in 2013)
  - For datasets with 10-100 million observations (rows)
    - Create bins
    - Condense (manages data in bins)
    - Plot condense objects. Ubercool autoplot().
    - Smooth plot (to focus on main trends and hide outliers)
  - More info (pre-print paper)

# How to display

- For R
  - [bigvis package](#) (by RStudio, not yet in CRAN despite started in 2013)

Bin-summarise-smooth: A framework for visualising large data



# How to display networks

- For R
  - iGraph
    - Non interactive visualization
    - API for
      - C
      - Python
      - R
  - Have a look at the R Journal
    - E.g.: packages sna and networks (Butts 2014)



# How to display networks

- For R
  - ggplot2 style for networks
    - See [paper](#) (R Journal 2017)
    - Functions
      - [ggnet2](#) (in package GGally)
        - » Returns a ggplot2 object
        - » Different layers for labels, shapes, etc..
        - » Arguments
          - » an igraph/network
          - » a layout for the graph
          - » ggplot2 args

Functionality	ggnet2 (GGally)	geom_net (geomnet)	geom_nodes, geom_edges, etc (ggnetwork)
Data	object of class "network" or object easily converted to that class (i.e. incidence or adjacency matrices, edge list) or object of class "igraph"	a fortified "network", "igraph", "edgedf", or "adjmat" object OR one edge data frame and one node data frame to be merged internally	same as ggnet2
Naming conventions	node_., edge_., label_., edge.label_., for alpha, color, etc.	arguments identical to ggplot2 with exception of ecolor, ealpha	same as ggplot2
Layout package & default	sna, Fruchterman-Reingold	sna, Kamada-Kawai	sna, Fruchterman-Reingold
Aesthetic mappings to variables	all alpha, color, shape, size for nodes, edges, labels	colour, size, shape, x, y, linetype, linewidth, label, group, fontsize	same as ggplot2
Arrows	directed = TRUE, arrow.size, gap	arrowsize, gap, arrow = arrow() like ggplot2	specify arrows in geom_edge like in code-geom_segment, arrow.gap
Theme or palette changes	done in the function with arguments like .legend, .palette, etc. and adding ggplot2 elements	adding ggplot2 elements	adding ggplot2 elements
Creating small multiples	created separately, use grid.arrange from gridExtra	add group argument to fortify() and use facet_*() from ggplot2	use by argument in ggnetwork() and facet_*() from ggplot2
Edge labelling?	Yes	No	Yes
Draw self-loops?	No	Yes	No

Table 1: Comparing the three different package side-by-side.

# How to display networks

- For R
  - ggplot2 style for networks
    - See [paper](#) (R Journal 2017)
    - Packages
      - [geomnet](#)
        - » A single layer
      - [ggnetwork](#)

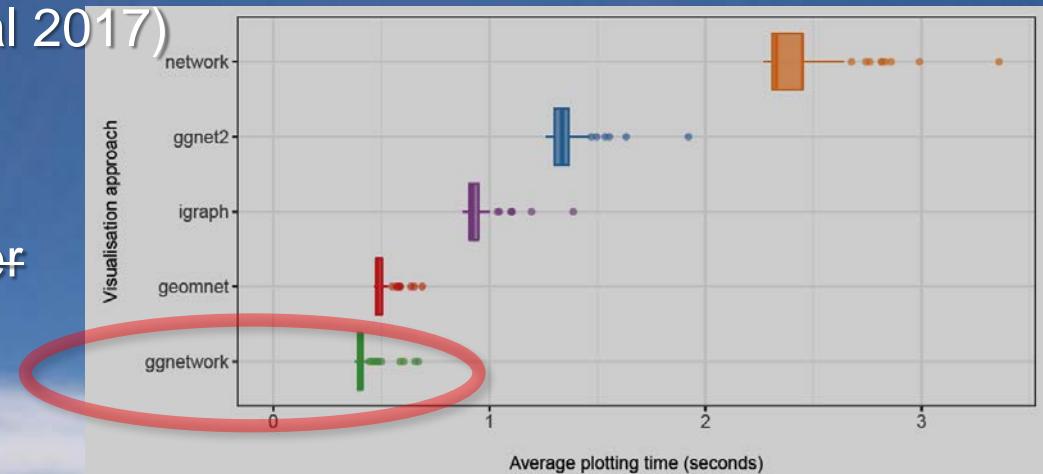
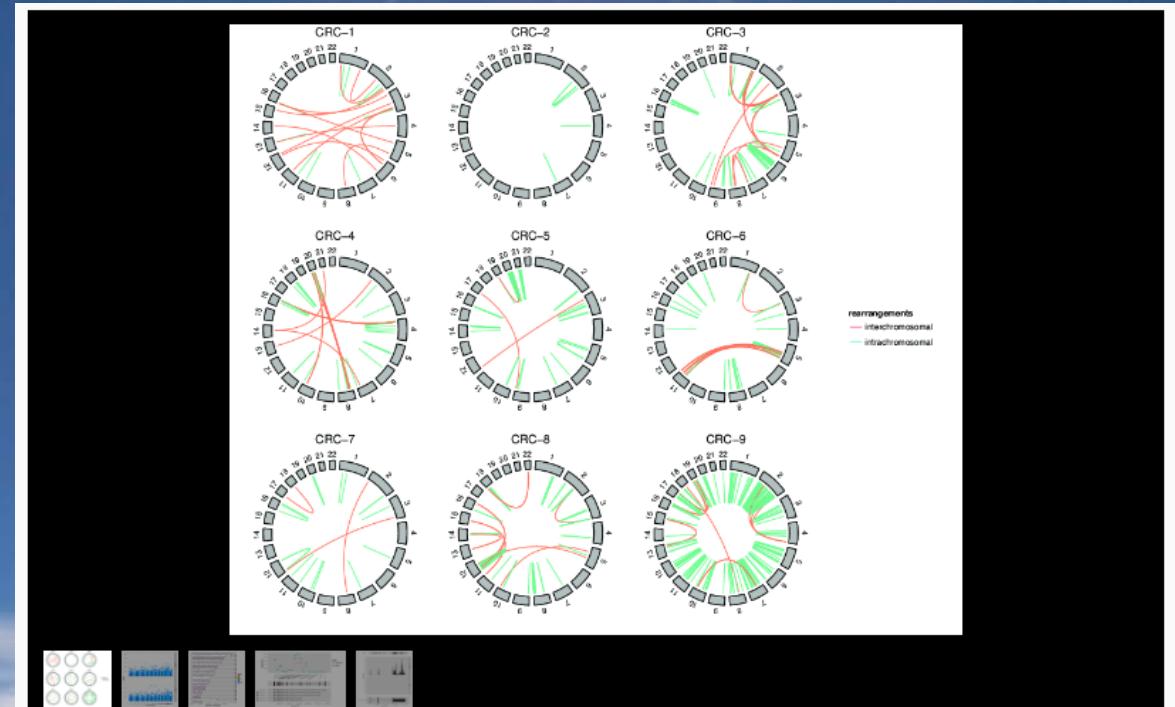


Figure 10: Comparison of the times needed for calculating and rendering the previously discussed protein interaction network in the three ggplot2 approaches and the standard plotting routines of the **network** and **igraph** packages based on 100 evaluations each.

# More ggplot2 style

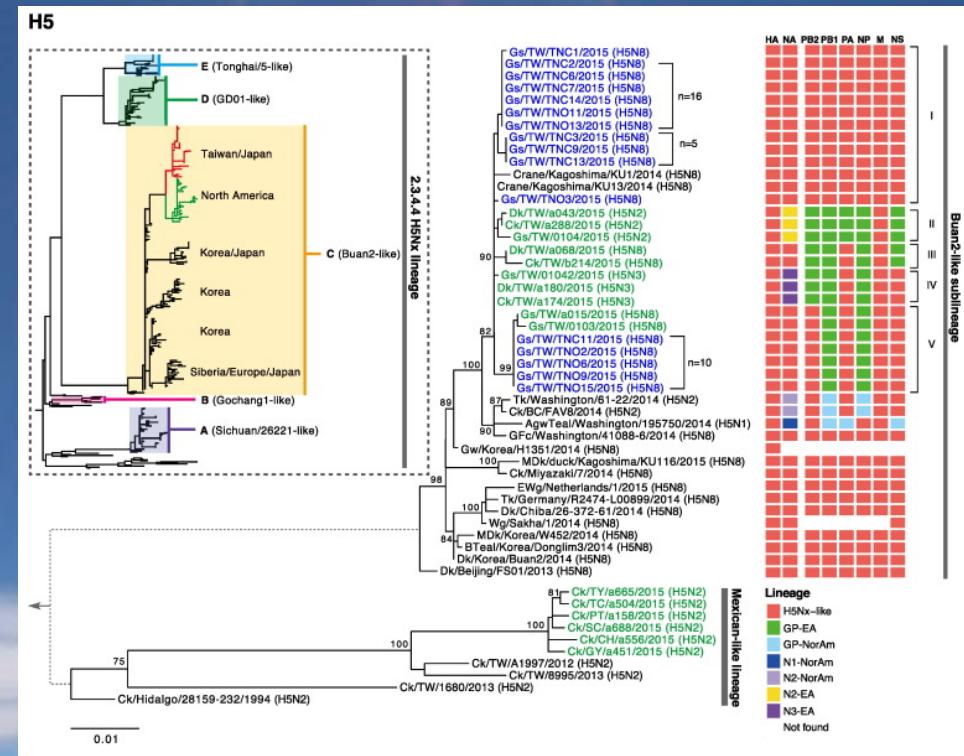
- ggbio



ggbio: An R implementation for extending the Grammar of Graphics for Genomic Data

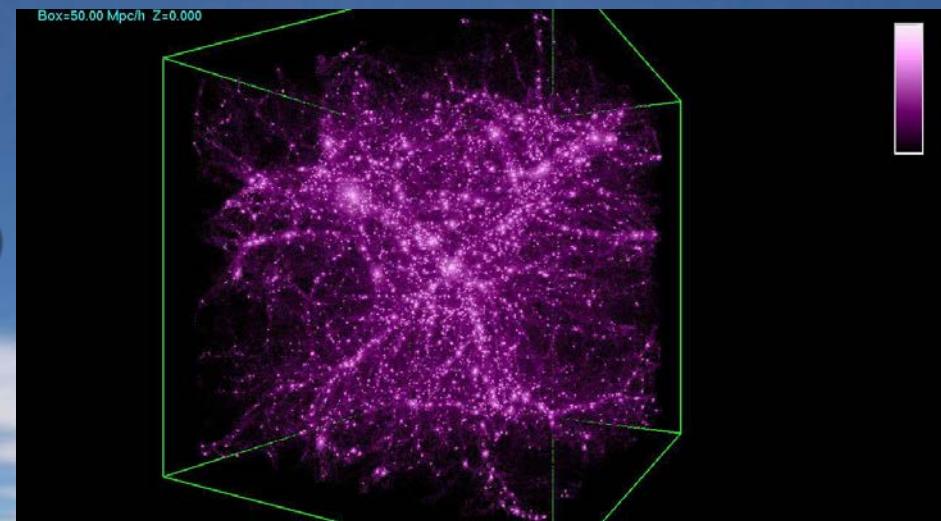
# More ggplot2 style

- ## • ggtree



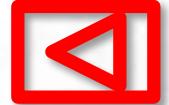
# Hardware limits

- Use GPUs
- Specific software
  - PMViewer rendering framework
    - Open source
    - 10 year old (aka “stable”)
    - Only for N-body simulations
    - See [the gallery](#)



# What can I do with my PC?

- Aggregate points up to your screen resolution
- Use a modest rendering software
  - Which one? Wait for a while and see ☺



How to explore big datasets

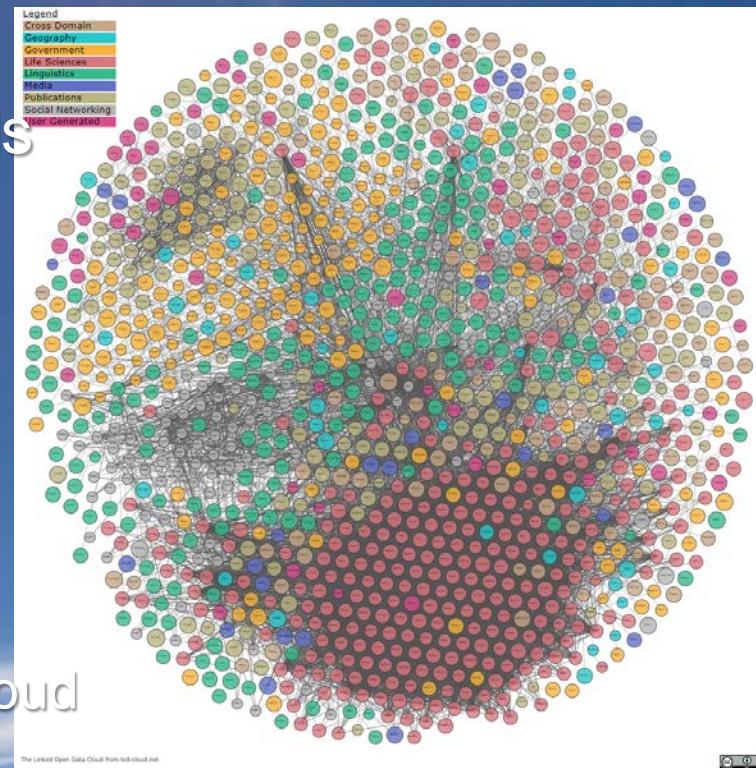
# BIG DATA USE CASE: LOD CLOUD



In the LOD galaxy...

# LOD galaxy

- Numbers
  - Number of datasets: thousands
  - Number of triples: billions
    - DBpedia: 38M [instances](#)
      - 2.7K dbo properties (much more dbp)
    - Wikidata: 100M [instances](#)
      - 10K [props](#)
- How do we explore it?
  - The [LOD](#) (Linked Open Data) Cloud
    - 1349 datasets as of Dec 2024
      - » 16,000+ links



A photograph of a group of hikers walking up a snowy mountain slope. They are wearing various colored jackets and backpacks, and some are using ski poles. The snow-covered ground has tracks from previous hikers.

Big VISIBLE & INTERACTIVE graphs with  
**GEPHI**

# Gephi

- Big Graphs (up to ~1M nodes)
- 100% Java app (Windows, Mac, Linux)
- Open, free
- Java API
- Multicore algorithms
- Extensible (plugins). E.g.:
  - Graph import from SPARQL queries results

- Installation
  - It is picky
    - Install Gephi version 0.10.1

Gephi  
Version 0.10.x (Apr 2022-today)  
untested on Windows 10 & iOS  
Doesn't support (yet)  
SemanticWebImport ☹



# Gephi

- Installation
  - It is picky
    - Install Gephi version 0.82 to support the WebImport plugin (latest version, 0.9, does NOT support it yet on all platforms)
      - Configure Gephi to use JDK (no JRE) SE 7 (Java 8 not supported yet). Oracle ended support for Java 7 (You must be registered to download [from here](#))
        - » {gephiDir}/etc/gephi.conf is within a “read only” folder.
        - » Modify the file, save it to Desktop, move it to folder

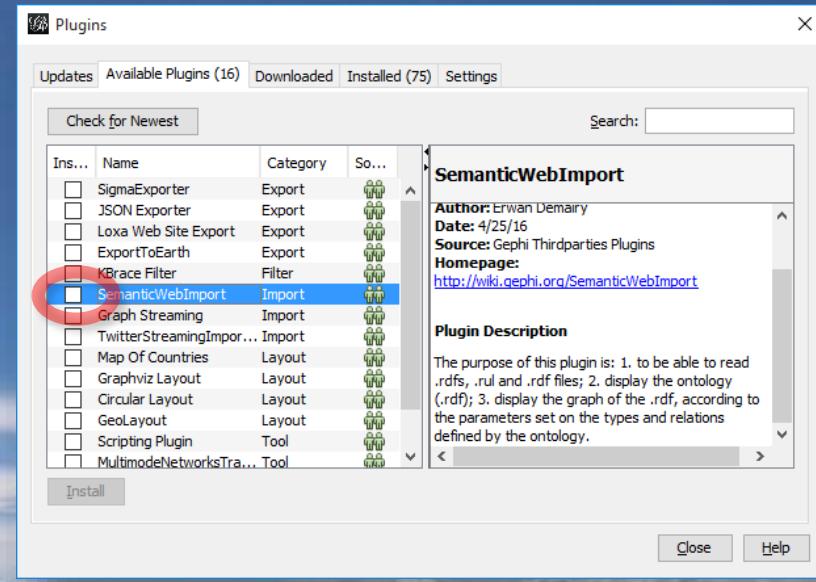
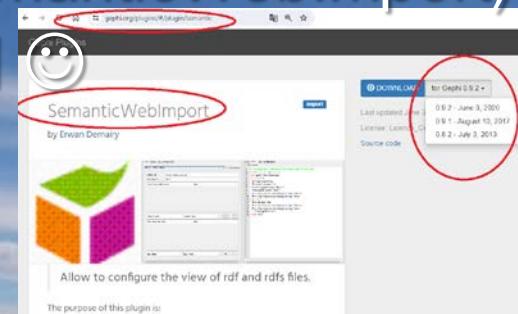
Java SE Development Kit 7 Downloads

End of Public Updates for Oracle JDK 7

This release will be the last Oracle JDK 7 publicly available update. For more information, and details on how to receive longer term support for Oracle JDK 7, please see the Oracle Java SE Support Roadmap.

# Gephi

- 0.9.2 version Installation on Windows
  - Can run without **uninstalling** previous versions
  - The WebImport plugin (now SemanticWebImport) is listed ☺

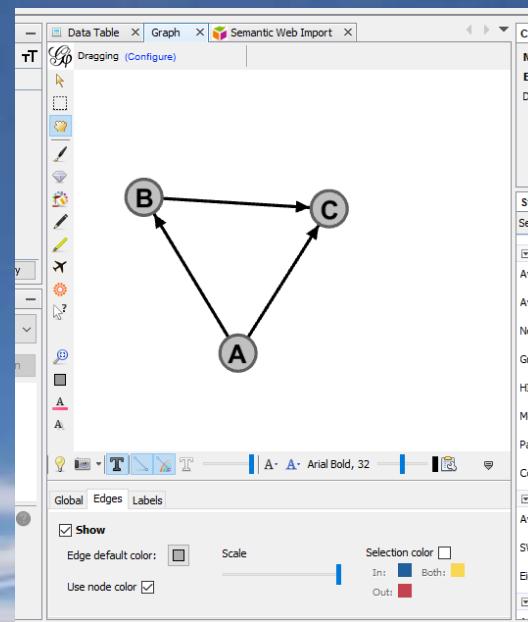


# Gephi

- 0.92 version Installation on Mac (iOS)
  - It works. If you have issues, please, tell me ☺
- 0.91 version Installation on Mac (iOS)
  - You can install the plugin but  
IT DOES NOT WORK ☹

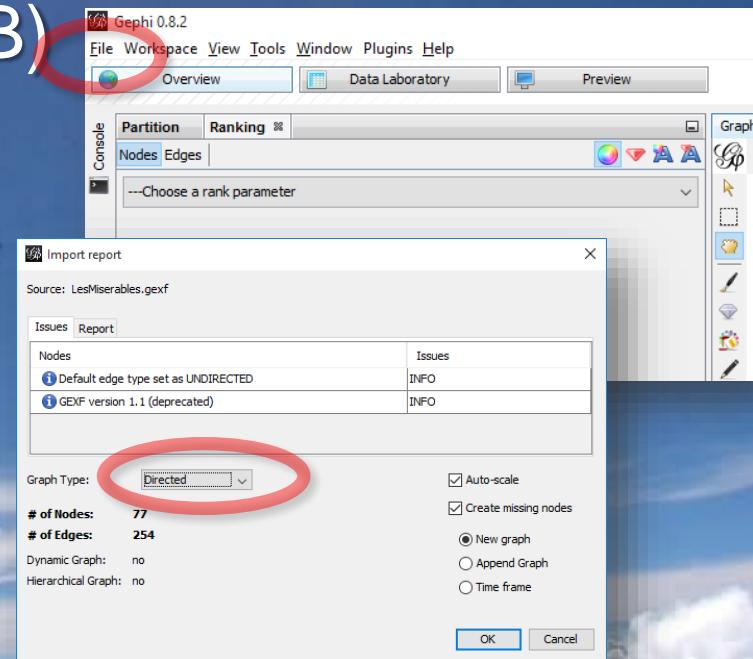
# Hands on

- Type data manually in Gephi's table
  - Node A
  - Node B
  - Node C
  - Directed Edge A-B
  - Directed Edge A-C
  - Directed Edge B-C
- See the resulting graph
  - Zoom in/out using
    - “Two fingers” in touch pad
    - Mouse wheel



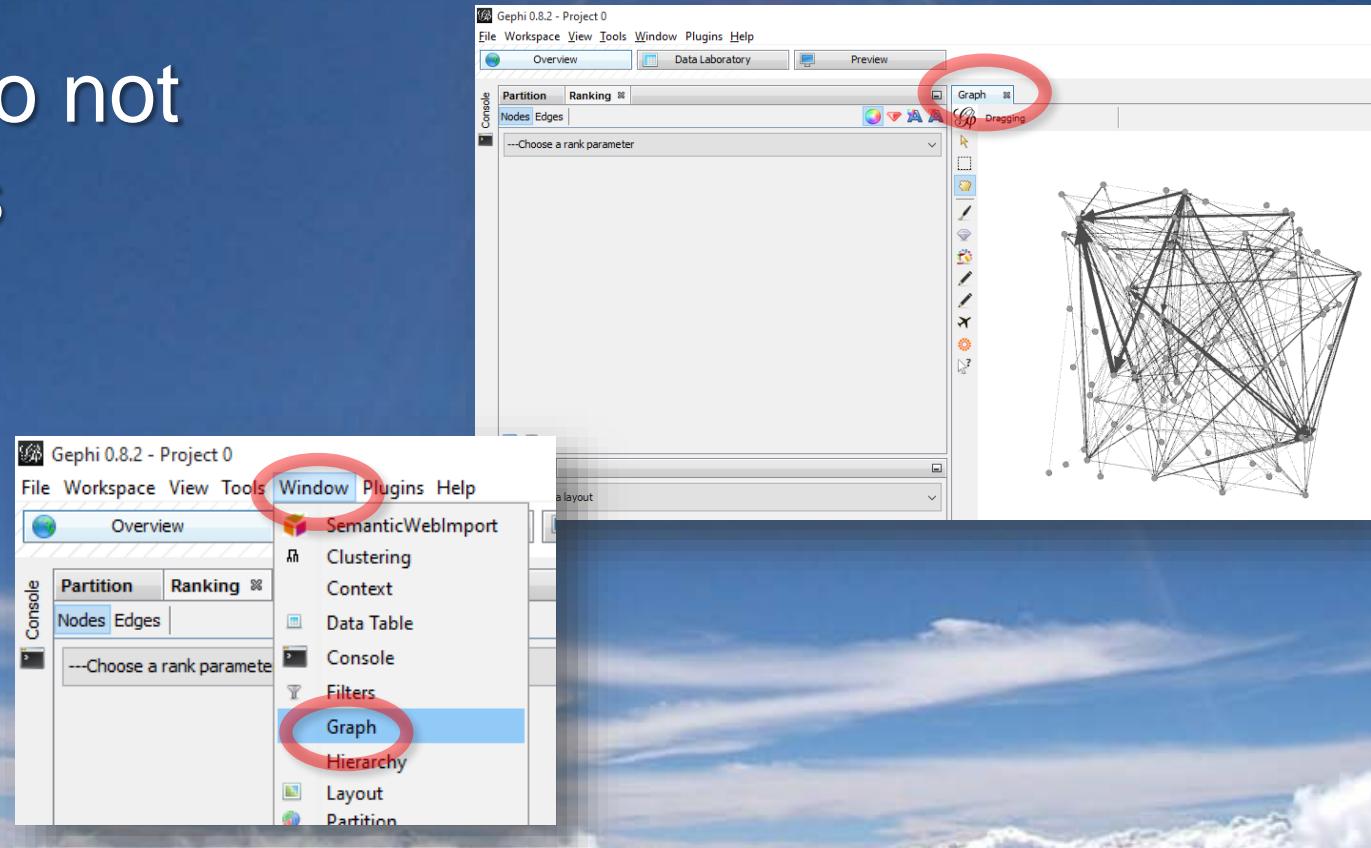
# A not so quick tutorial

- Download a “gephi graph file”
  - [LesMiserables.gexf](#) (17 KB)
- Open Gephi
- Load LesMiserables.gexf
  - File → Open...
  - Use “undirected”



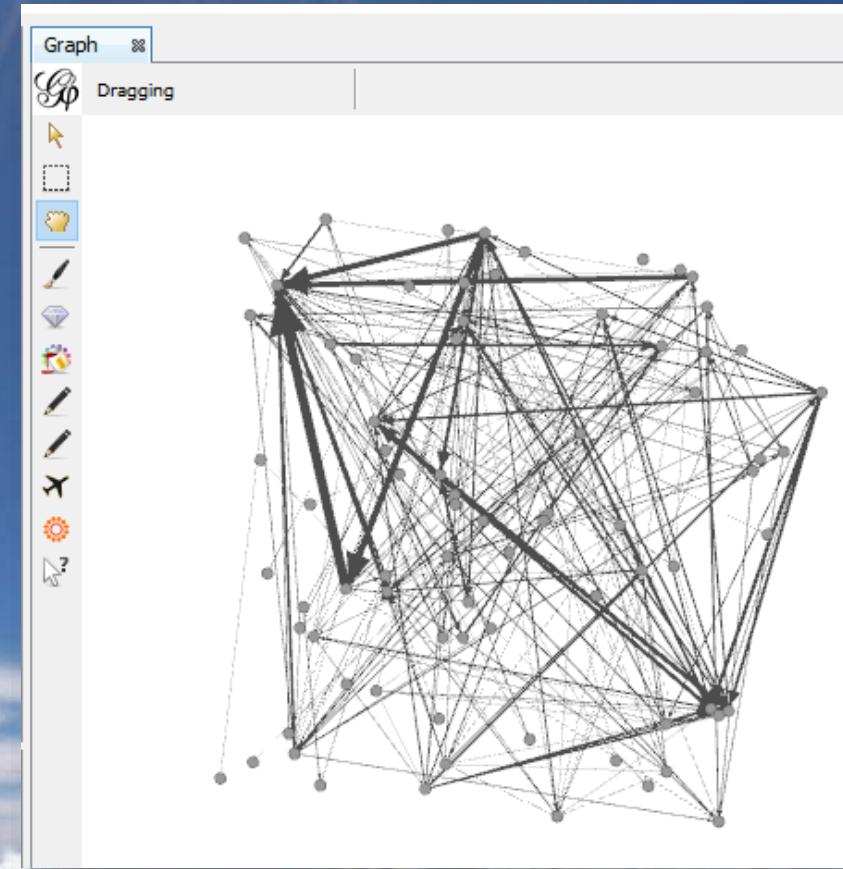
# Quick tutorial

- If you do not see this
- Try this



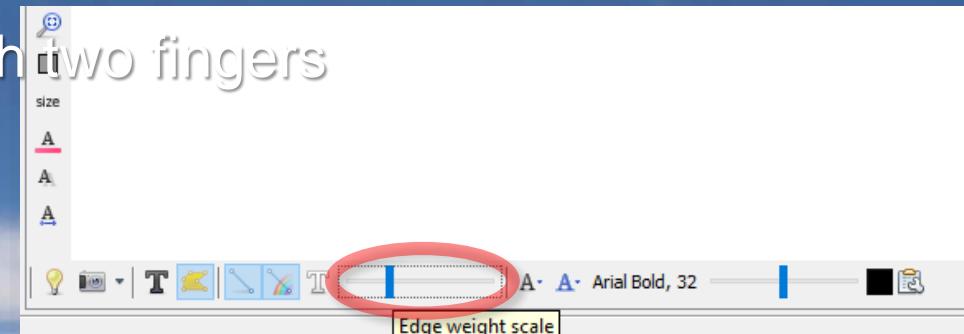
# Quick tutorial

- Node positions are random.  
You will see a different graph
- Each node is a character of  
Victor Hugo's novel
- This graph is a  
“Co-appearance weighted  
network”, that is,
  - Two vertices are adjacent if the  
corresponding characters  
encounter each other,  
in selected chapters of the novel



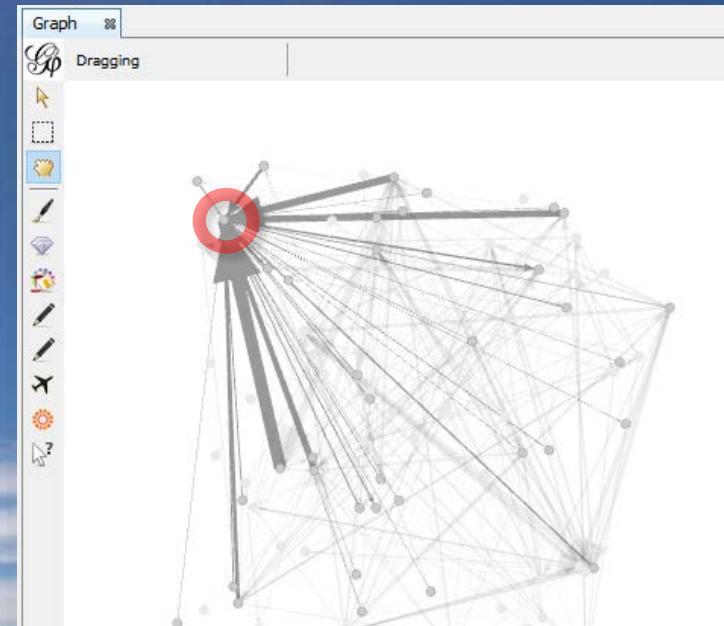
# Quick tutorial

- Basics
  - Move the whole graph
    - Drag the mouse while **right** button is pressed
  - Zoom in/out
    - Use mouse wheel
    - or drag touchpad with **two fingers**
  - Width of edges (arcs)
    - Slider (no “text size” slider)



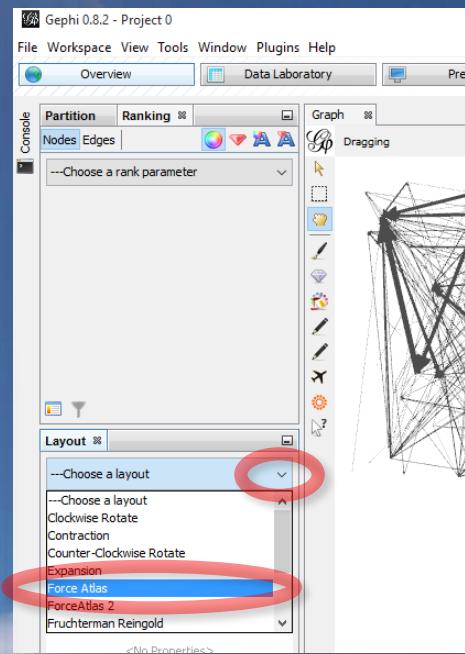
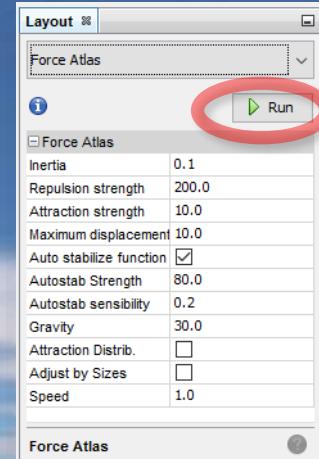
# Quick tutorial

- Basics (cont.)
  - Hover (locate mouse pointer) over a node shows the links to/from that node
    - “Fading out” the rest of the graph



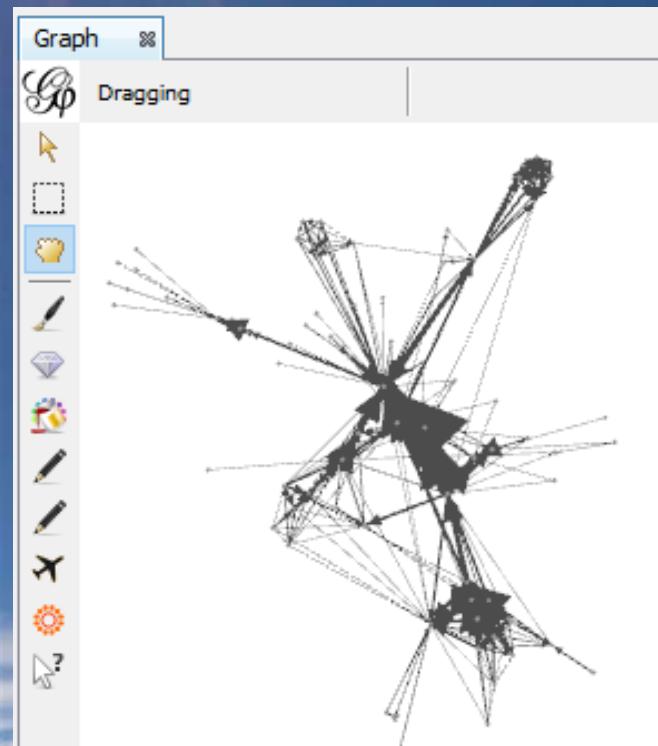
# Quick tutorial

- Layout
  - Click on “Force Atlas”
    - This is a layout algorithm
      - Linked nodes “attract”, non-linked nodes are “repelled” (pushed apart)
    - Do not change default parameters
    - Press “Run” button



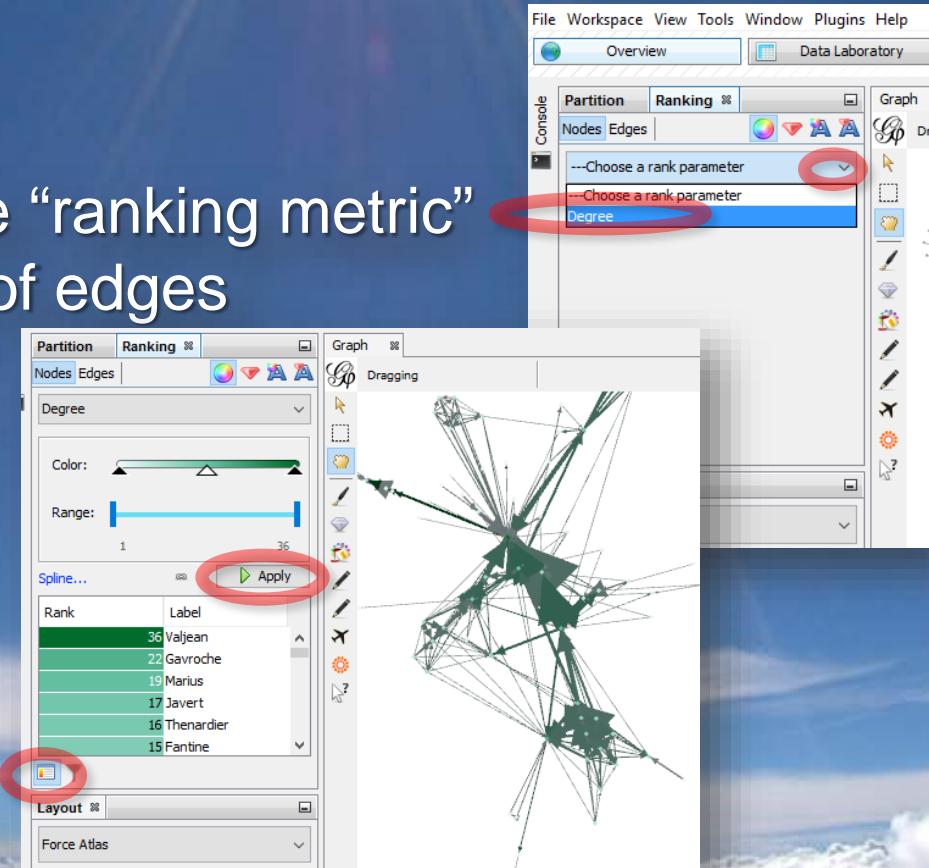
# Quick tutorial

- Layout (cont.)
  - Press “Stop” button
  - Zoom in
  - Drag the graph
- You should get (something like) this



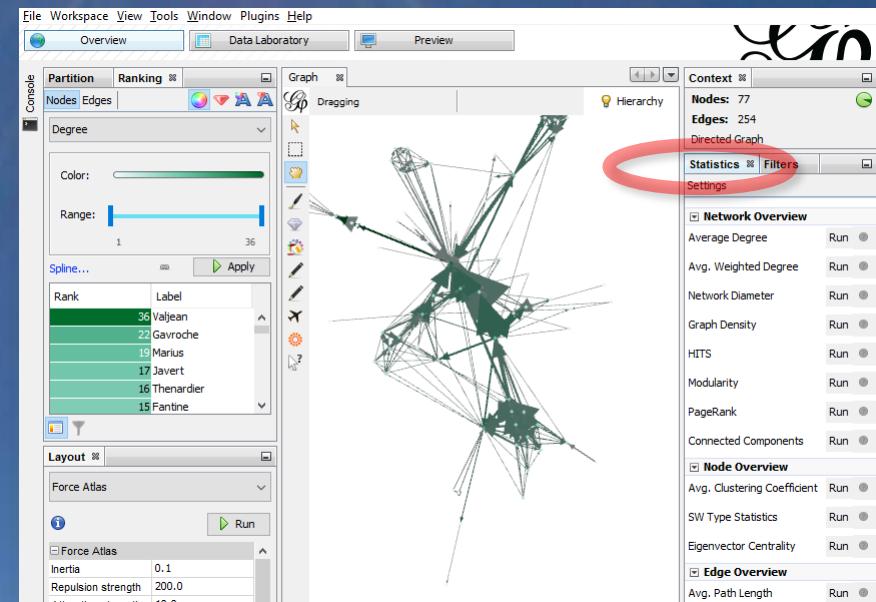
# Quick tutorial

- Nodes' color
  - By default, the unique “ranking metric” is “Degree” (number of edges of each node)
    - You will get more metrics later on
  - Select “Degree”
  - Press “Apply”
  - Show rank values



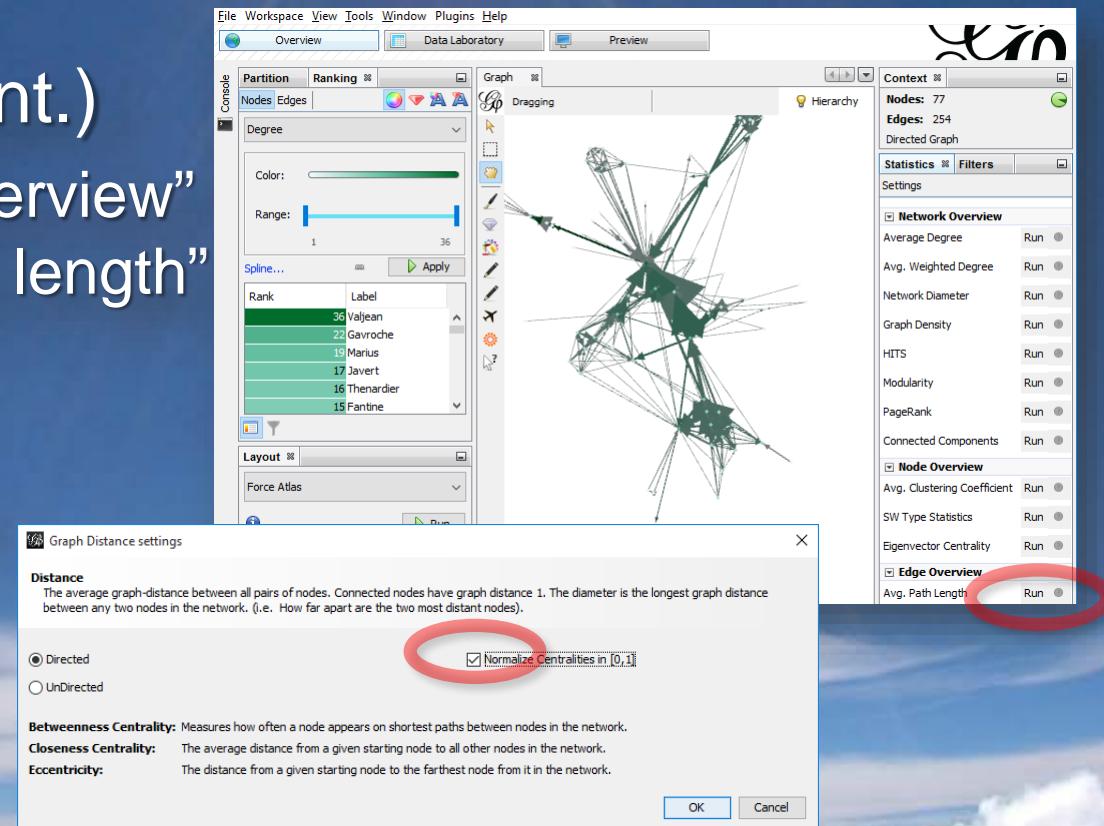
# Quick tutorial

- Graph metrics
  - Many metrics (Statistics tab on the right side)
    - In 3 groups
      - Network overview
        - » Average Degree
        - » ...
        - » Page Rank
      - Node overview
      - Edge overview



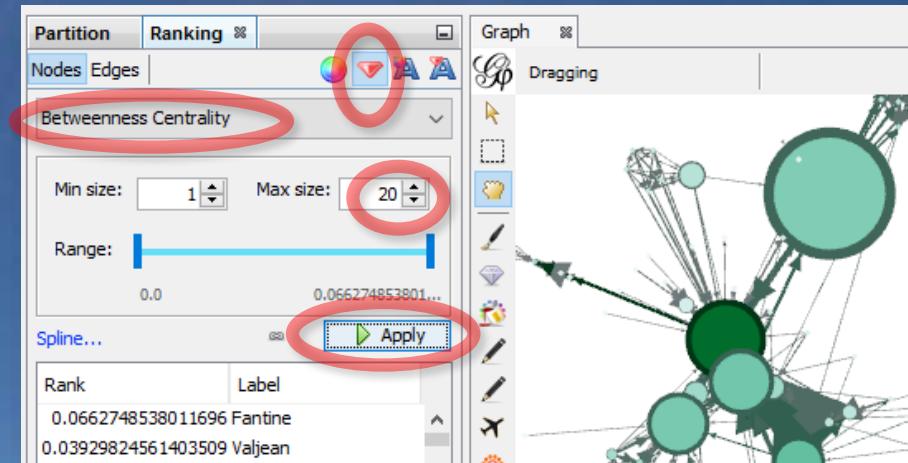
# Quick tutorial

- Graph metrics (cont.)
  - In group “Edge overview” run the “Avg. Path length” method (press “Run” button)
  - 3 new metrics are computed:
    - Betweenness Centrality
    - Closeness Centrality
    - Eccentricity



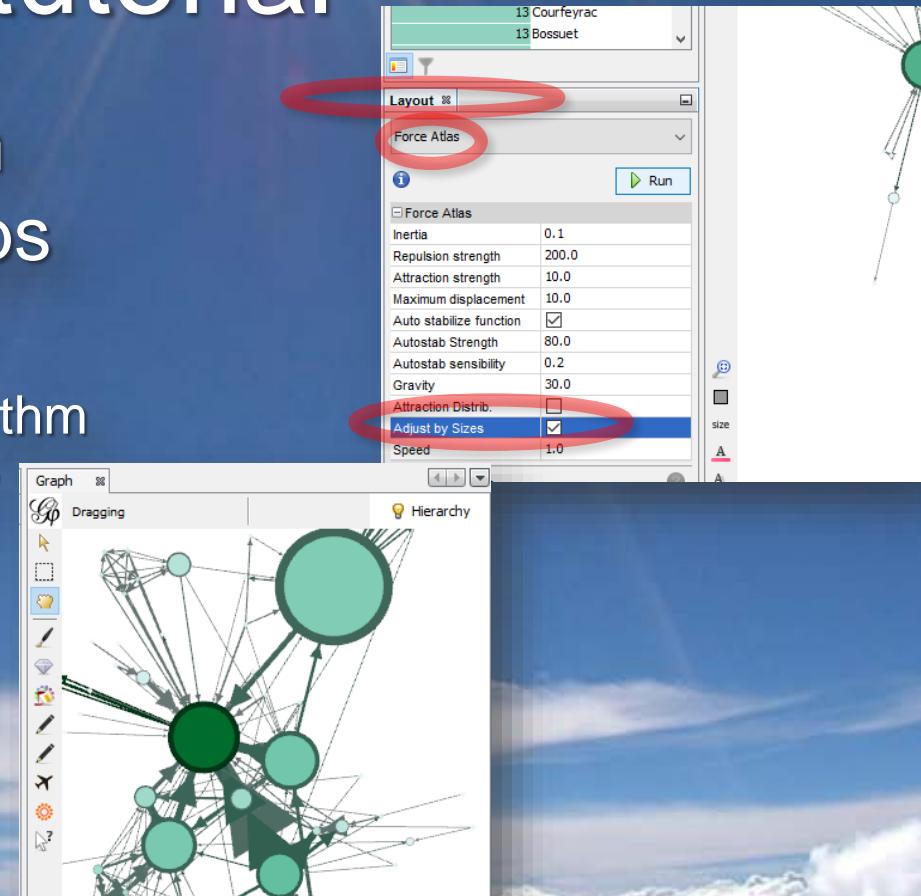
# Quick tutorial

- Graph metrics (cont.)
  - Now you have more metrics in the “rank parameter”
  - Click node size icon 
  - Select “Betweenness Centrality”
    - Higher values mean more influential (nodes); i.e. “bridges” between relevant nodes.
  - Apply (change “Max size”) to apply this metric to nodes’ size. Notice that node’s color used “Degree” rank



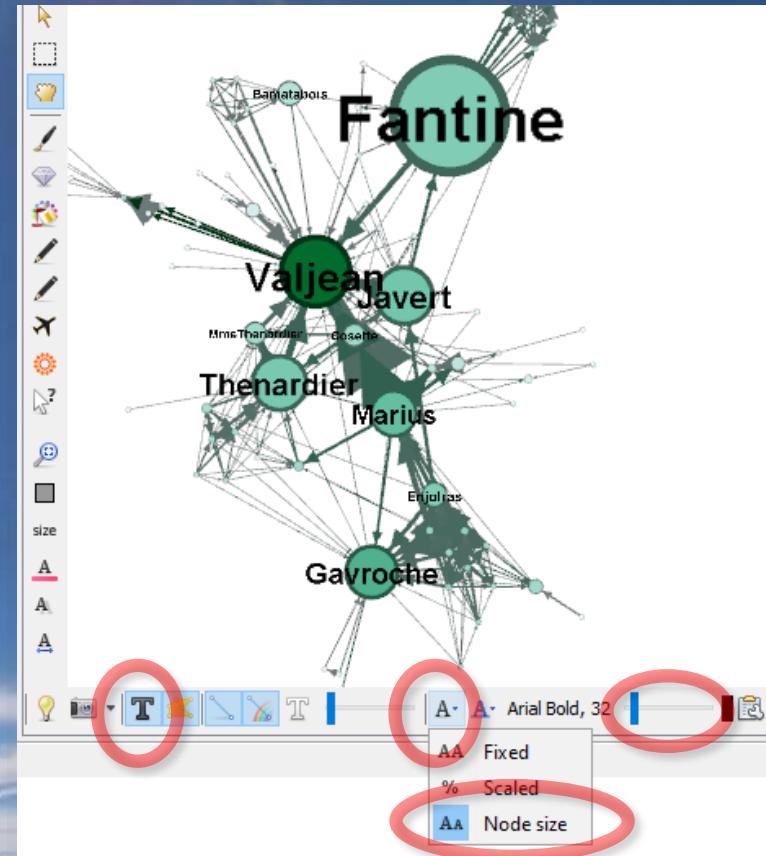
# Quick tutorial

- Re-run layout algorithm avoiding nodes' overlaps
  - In “layout” tab
    - select “Force Atlas” algorithm
    - Click on “Adjust by Sizes”
    - Run for a while
  - You get (something like) this



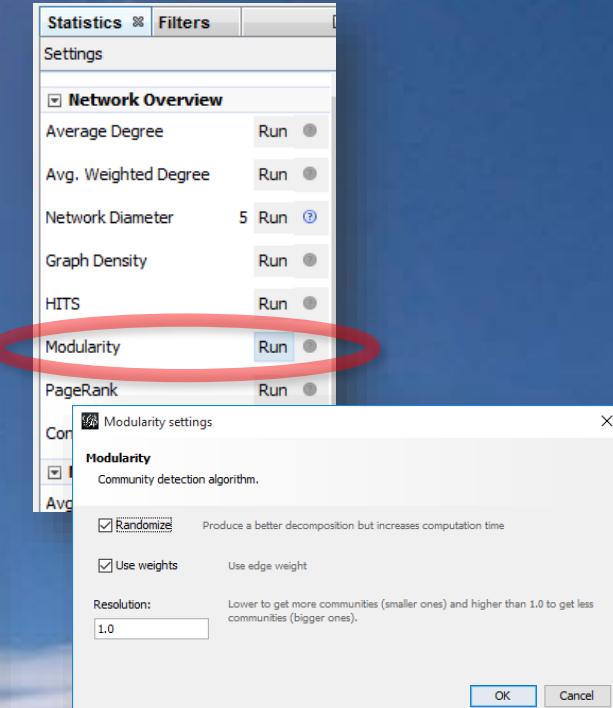
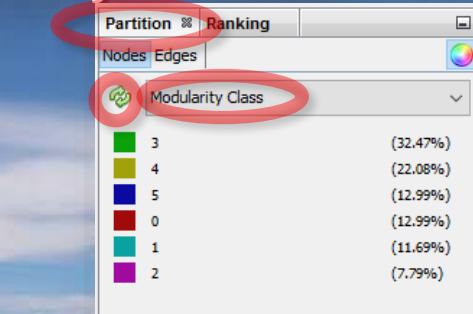
# Quick tutorial

- Show Nodes' labels
  - Click on the  icon
  - Select “Node size” in the “size mode” selector
  - Move the “text size” slider
- You should get (something like) this



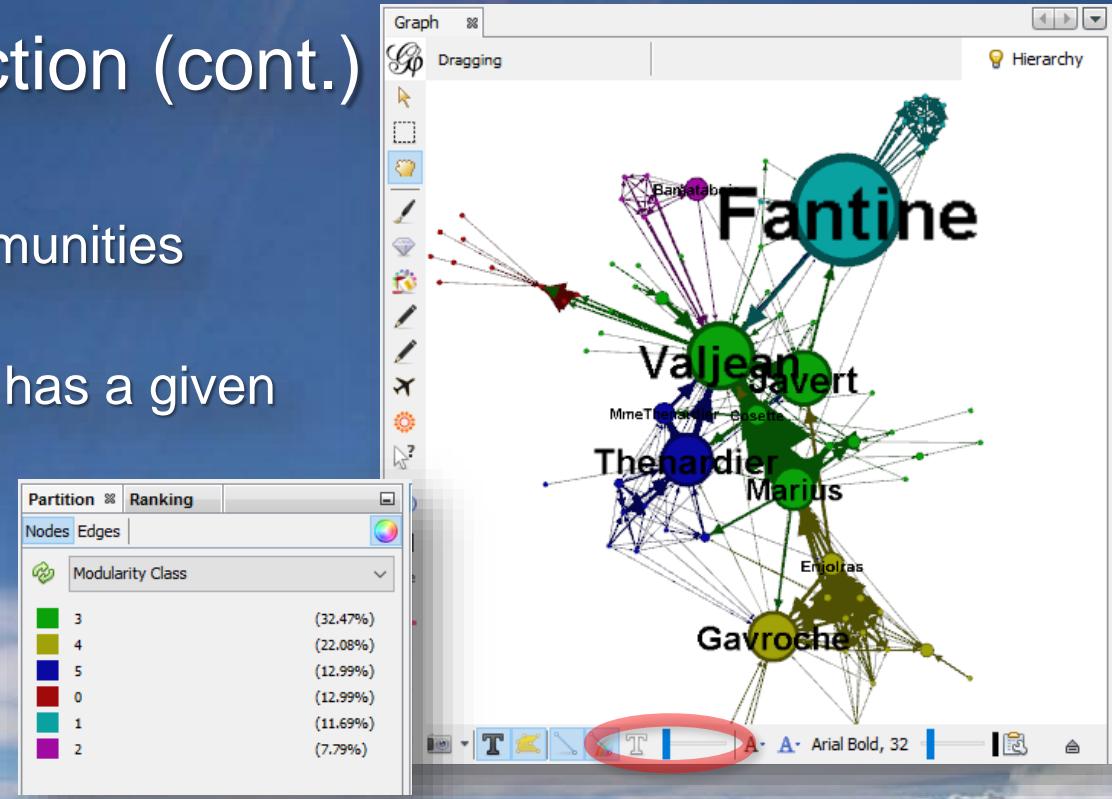
# Quick tutorial

- Community detection
  - Run “modularity” in Statistics tab
    - With default parameters
  - Select “Partition” tab
  - Select “Modularity Class”
    - Refresh



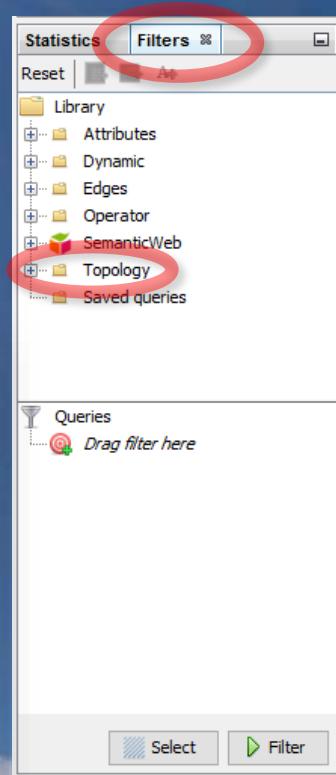
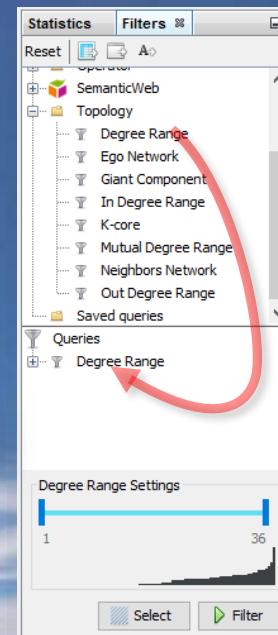
# Quick tutorial

- Community detection (cont.)
  - You get this
    - There are 6 communities (clusters)
    - Each community has a given node/edge color
  - Reduce the edge weight scale
    - To get thinner edges



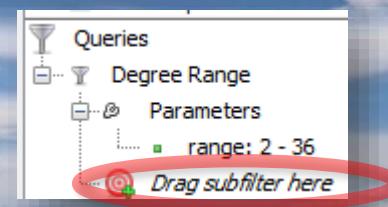
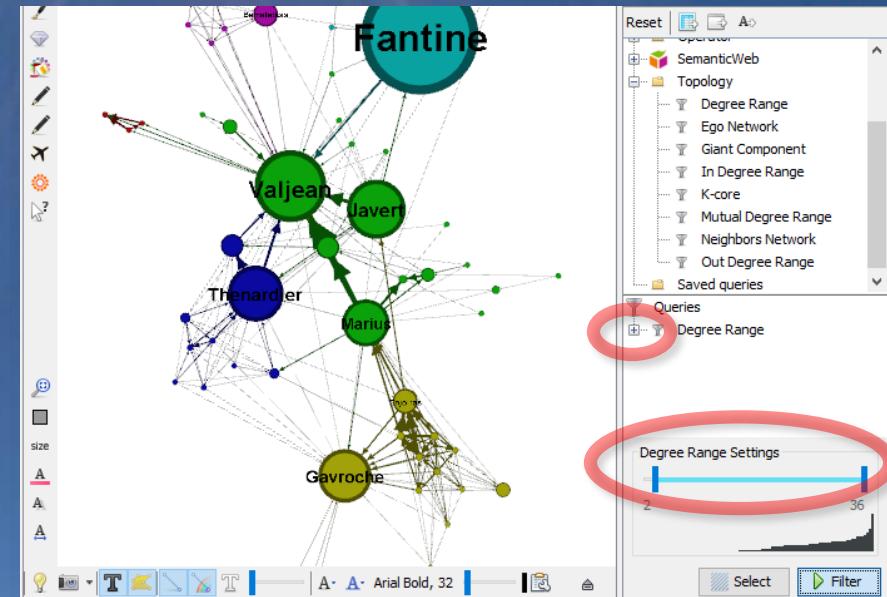
# Quick tutorial

- Filters (cleaning up the graph)
  - Click on the “Filters” tab
  - Expand the “Topology” folder
  - Drag the “Degree Range” to the target icon



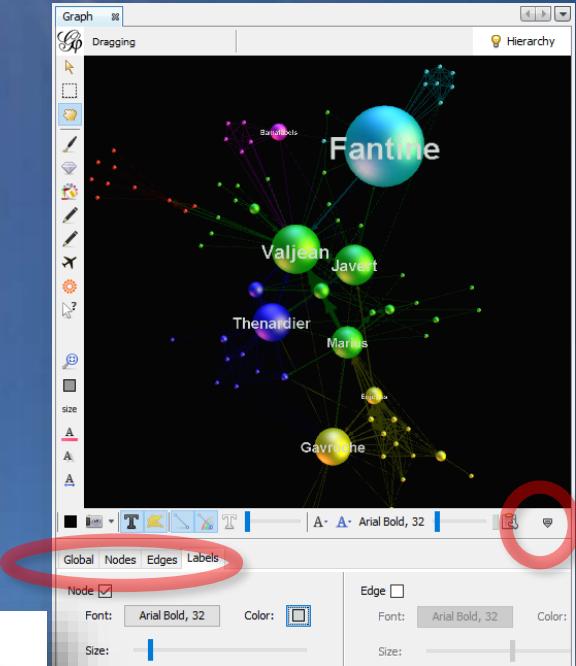
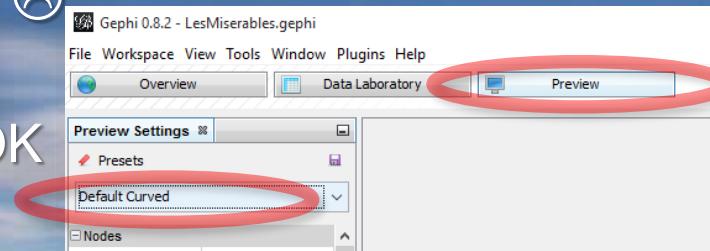
# Quick tutorial

- Filters (cont.)
  - Move the slider to remove nodes with a degree lower than 2
  - Apply the filter
    - You get a cleaner graph
  - This query can be refined (expand the filter) with subfilters



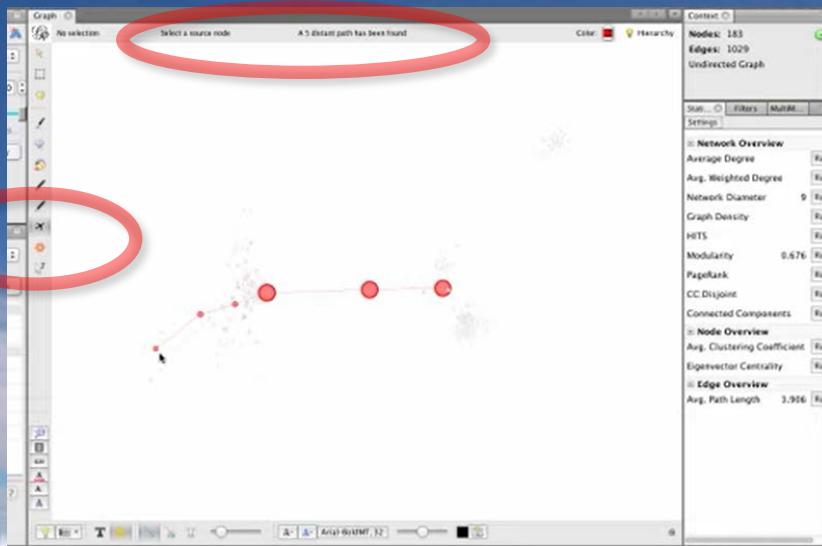
# Quick tutorial

- Final output
  - Expand the “details” icon 
  - Assign values for bg color, etc.
  - Click the “Preview” window
    - I can not make it work on Windows 😞
    - But the file is OK



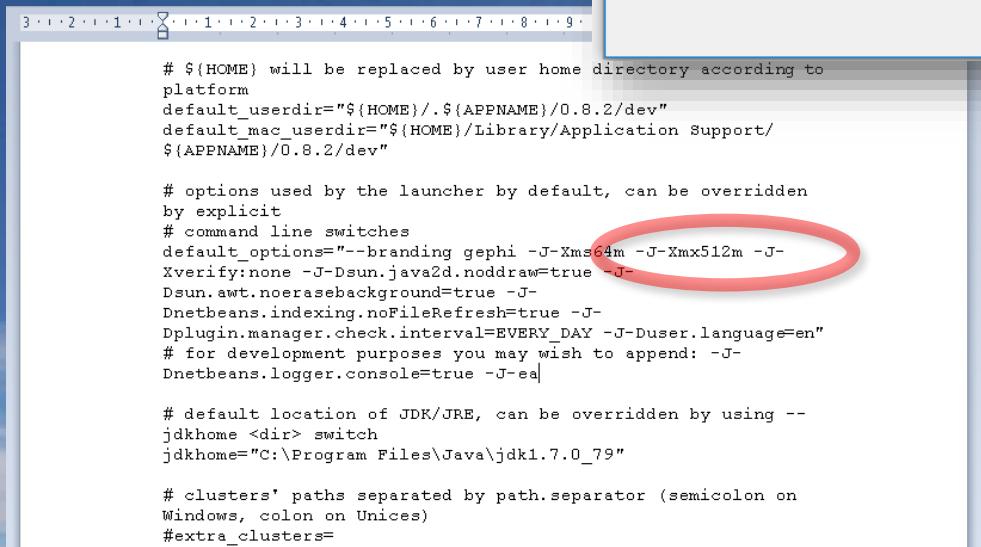
# Extras

- Getting the shortest path in a graph
  - Use the “airplane icon”. Video [here](#).



# Gephi

- Out of memory
  - Edit {gephiDir}/etc/gephi.conf

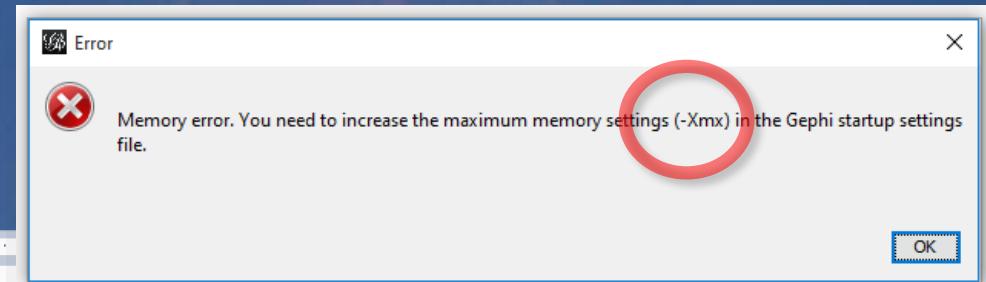


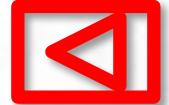
```
# ${HOME} will be replaced by user home directory according to
platform
default_userdir="${HOME}/.${APPNAME}/0.8.2/dev"
default_mac_userdir="${HOME}/Library/Application Support/
${APPNAME}/0.8.2/dev"

# options used by the launcher by default, can be overridden
by explicit
# command line switches
default_options="--branding gephi -J-Xms64m -J-Xmx512m -J-
Xverify:none -J-Dsun.java2d.nodraw=true -J-
Dsun.awt.noerasebackground=true -J-
Dnetbeans.indexing.noFileRefresh=true -J-
Dplugin.manager.check.interval=EVERY_DAY -J-Duser.language=en"
# for development purposes you may wish to append: -J-
Dnetbeans.logger.console=true -J-ea]

# default location of JDK/JRE, can be overridden by using --
jdkhome <dir> switch
jdkhome="C:\Program Files\Java\jdk1.7.0_79"

# clusters' paths separated by path.separator (semicolon on
Windows, colon on Unices)
#extra_clusters=
```



A photograph of a group of hikers walking in a single file line across a snowy mountain slope. They are wearing various colored jackets and backpacks. The background shows a vast, snow-covered landscape under a bright sky.

# Big Linked Data graphs with **GEPHI LAYOUTS**

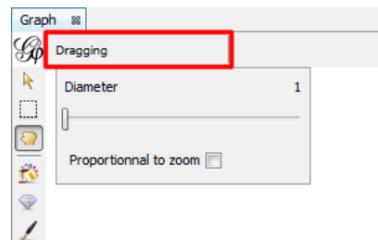
# Gephi layouts

- See documentation at  
<https://gephi.org/tutorials/gephi-tutorial-layouts.pdf>
- Here I will show you a summary

# Gephi layouts

## Play against the algorithm!

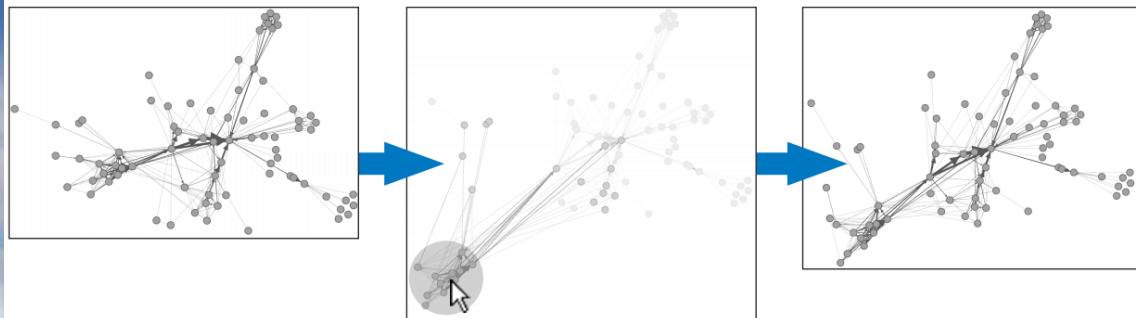
Run the layout again and drag the nodes to stress it.



- Locate Dragging action, in the top left of the Visualization panel.
- Adjust the selection diameter in the panel or by using the shortcut “Ctrl + Mouse Wheel”.

Increase the “Autostab strength” in the Layout Properties to 100 000, then drag the nodes. The graph becomes less deformed.

- And now Stop the algorithm.

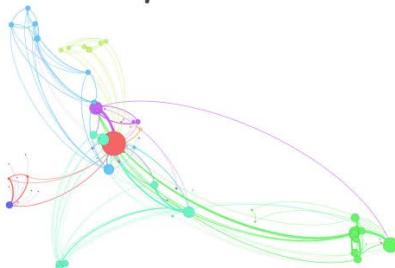


# Gephi layouts

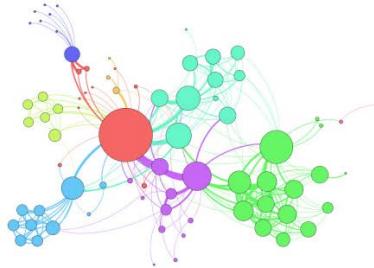
Various layouts exist

Gephi implements various layout algorithms. They set the shape of the graph.

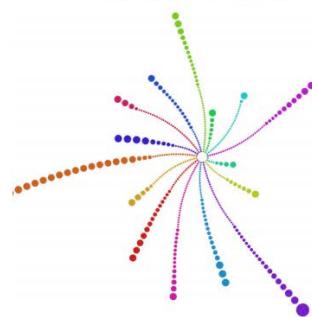
*OpenOrd*



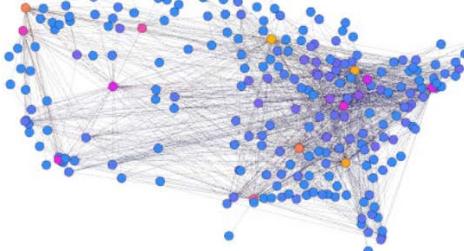
*ForceAtlas 2*



*Radial Axis*



*GeoLayout*



Airlines sample dataset: <http://gephi.org/datasets/airlines-sample.gexf>

# Gephi layouts

So how to choose a layout?

In general, select one according to the feature of the topology you want to highlight:

emphasis  
**DIVISIONS**

*OpenOrd*

emphasis  
**COMPLEMENTARITIES**

*ForceAtlas, Yifan Hu,  
Frushterman-Reingold*

emphasis  
**RANKING**

*Circular, Radial Axis*

emphasis  
**GEOGRAPHIC  
REPARTITION**

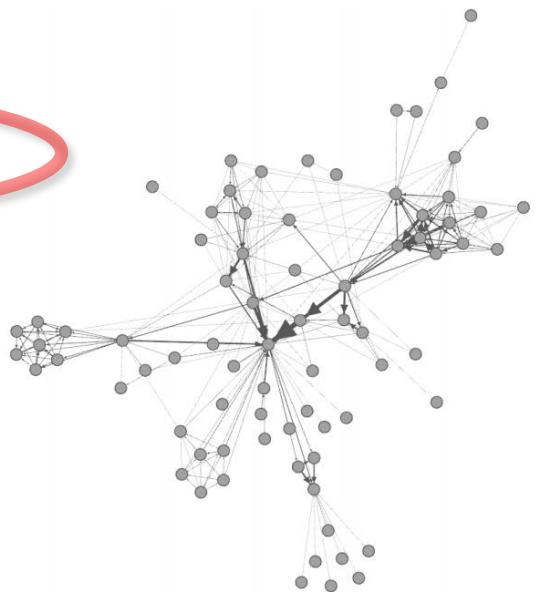
*GeoLayout*

# Gephi layouts

## ForceAtlas layout

Home-brew layout of Gephi, it is made to spatialize Small-World / Scale-free networks. It is focused on quality (meaning “being useful to explore real data”) to allow a rigorous interpretation of the graph (e.g. in SNA) with the fewest biases possible, and a good readability even if it is slow.

Author:	Mathieu Jacomy
Date:	2007
Kind:	Force-directed
Complexity:	$O(N^2)$
Graph size:	1 to 10 000 nodes
Use edge weight:	Yes



# Gephi layouts

## ForceAtlas layout

Home-brew layout of Gephi, it is made to spatialize Small-World / Scale-free networks. It is focused on quality (meaning “being useful to explore real data”) to allow a rigorous interpretation of the graph (e.g. in SNA) with the fewest biases possible, and a good readability even if it is slow.

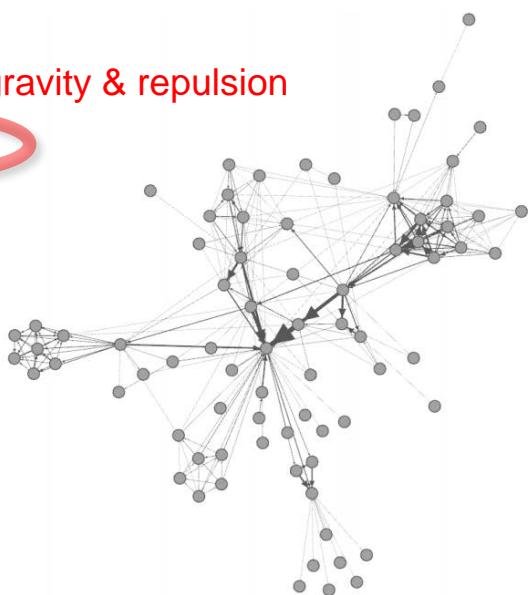
Author:	Mathieu Jacomy
Date:	2007
Kind:	Force-directed: gravity & repulsion
Complexity:	$O(N^2)$
Graph size:	1 to 10 000 nodes
Use edge weight:	Yes

### Run ForceAtlas

 run the layout by applying the following settings step by step:

- Autostab strength = 2 000 Increase to move the nodes slowly.
- Repulsion strength = 1 000 How strongly does each node reject others.
- Attraction strength = 1 How strongly each pair of connected nodes attract each other.
- Gravity = 100 Attract all nodes to the center to avoid dispersion of disconnected components.
- Attraction Distrib. = checked Push hubs (high number of output links) at the periphery and put authorities (high number of input links) more central.

And now  Stop the algorithm.



# Gephi layouts

## Fruchterman-Reingold layout

It simulates the graph as a system of mass particles. The nodes are the mass particles and the edges are springs between the particles. The algorithms try to minimize the energy of this physical system. It has become a standard but remains very slow.

Author:  
Date:  
Kind:  
Complexity:  
Graph size:  
Use edge weight:

Thomas Fruchterman & Edward Reingold<sup>1</sup>

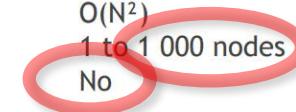
1991

Force-directed: **springs**

$O(N^2)$

1 to 1 000 nodes

No



### Run Fruchterman-Reingold

the layout by applying the following settings step by step:

- Area = 100
- Area = 100 000
- Gravity = 1 000
- Gravity = 100

Graph size area.

Attract all nodes to the center to avoid dispersion of disconnected components.

And now  the algorithm.

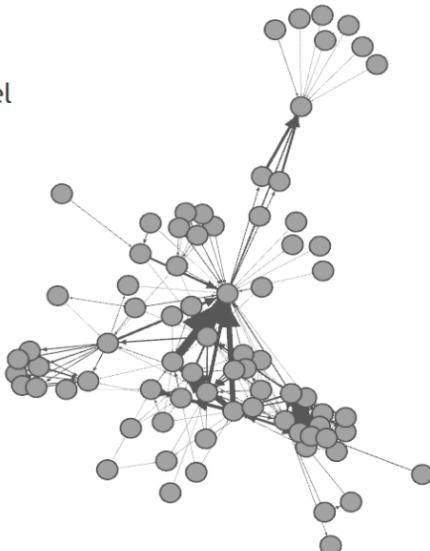


# Gephi layouts

## Yifan Hu Multilevel layout

It is a very fast algorithm with a good quality on large graphs. It combines a force-directed model with a graph coarsening technique (multilevel algorithm) to reduce the complexity. The repulsive forces on one node from a cluster of distant nodes are approximated by a Barnes-Hut calculation, which treats them as one super-node. It stops automatically.

Author:	Yifan Hu <sup>1</sup>
Date:	2005
Kind:	Force-directed + multilevel
Complexity:	$O(N \cdot \log(N))$
Graph size:	100 to 100 000 nodes
Use edge weight:	No



### Run Yifan Hu Multilevel

Launch the layout by applying the following settings step by step:

- Step ratio = 0.99 Run Ratio used to update the step size. Increase it for a better quality (vs speed).
- Optimal distance = 200 Run Natural length of the springs. Increase it to place nodes farther apart.
- Theta = 1.0 Run Approximation for Barnes-Hut calculation. Smaller values mean more accuracy.

# Gephi layouts

## OpenOrd layout

It expects undirected weighted graphs and aims to better distinguish **clusters**. It can be run in parallel to speed up computing, and stops automatically. The algorithm is originally based on Fruchterman-Reingold and works with a fixed number of iterations controlled via a simulated annealing type schedule (liquid, expansion, cool-down, crunch, and simmer). Long edges are cut to allow clusters to separate.

Author:	S. Martin, W. M. Brown, R. Klavans, and K. Boyack <sup>1</sup>
Date:	2010 (VxOrd)
Kind:	Force-directed + simulated annealing
Complexity:	$O(N^* \log(N))$
Graph size:	100 to 1 000 000 nodes
Use edge weight:	Yes

### Run OpenOrd

Launch the layout by applying the following settings step by step:

- Edge cut = 0.95

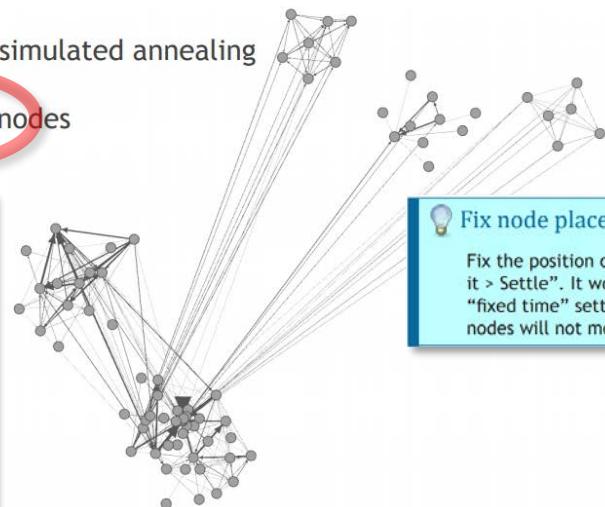
 From 0 (standard Fruchterman-Reingold) to 1. Percentage of the greatest distance between two nodes in the drawing. A higher cutting means a more clustered result.

- Num iterations = 100
- Num iterations = 850

 Contract the clusters.  
 Expand the clusters.

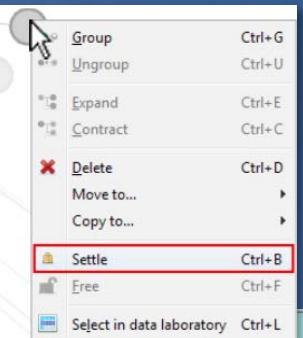
- Random seed = -6308261588084905834

 Use this value to produce exactly the same shape as shown before.



### Fix node placement

Fix the position of a node (or a group of selected nodes) by "Right-click on it > Settle". It works for all layouts except Yifan Hu. For OpenOrd, use the "fixed time" setting on the Layout panel to configure the time the fixed nodes will not move.



# Gephi layouts

## ForceAtlas 2 layout

Improved version of the Force Atlas to handle large networks while keeping a very good quality. Nodes repulsion is approximated with a Barnes-Hut calculation, which therefore reduces the algorithm complexity. Replace the “attraction” and “repulsion” forces by a “scaling” parameter.

Author: Mathieu Jacomy<sup>1</sup>  
Date: 2011  
Kind: Force-directed  
Complexity:  $O(N^* \log(N))$   
Graph size: 1 to 1 000 000 nodes  
Use edge weight: Yes

the layout by applying the following settings step by step:

- LinLog mode = checked
- LinLog mode = unchecked
- Scaling = 100
- Edge weight influence = 0

Linear attraction & logarithmic repulsion (lin-lin by default), makes clusters tighter.  
Increase to make the graph sparser.  
From 0 (no influence) to 1 (normal). Set 0 to calculate forces without edge weight.

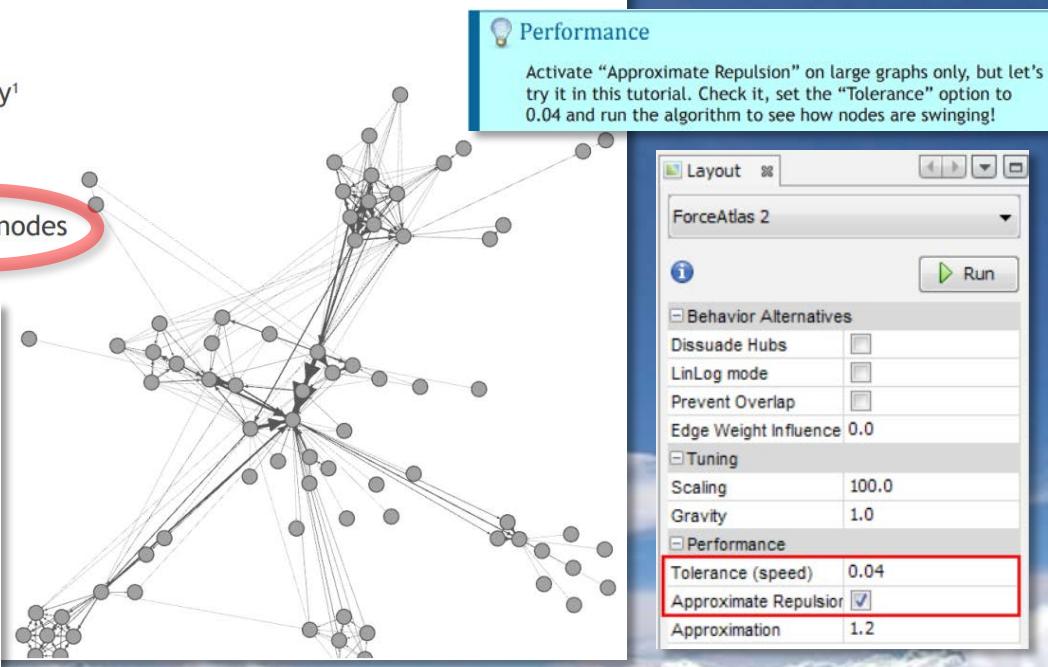
### Run ForceAtlas 2

 Run the layout by applying the following settings step by step:

- LinLog mode = checked
- LinLog mode = unchecked
- Scaling = 100
- Edge weight influence = 0

Linear attraction & logarithmic repulsion (lin-lin by default), makes clusters tighter.  
Increase to make the graph sparser.  
From 0 (no influence) to 1 (normal). Set 0 to calculate forces without edge weight.

And now  Stop the algorithm.



# Gephi layouts

## Detect communities

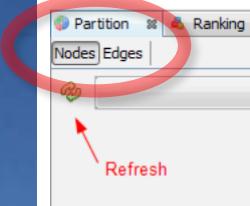
We now want to study the community structure in this network: does it divide naturally into groups of nodes with dense connections within groups and sparser connections between groups?

In the  Statistics panel, click on  Run near the “Modularity”<sup>1</sup> line.



The community detection algorithm created a “Modularity Class” value for each node. The partition module can use this new data to colorize communities.

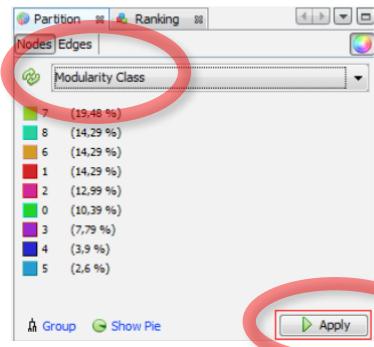
- Locate the  Partition module on the left panel.
- Click on the “Refresh” button to populate the partition list.



- Select “Modularity Class” in the partition list.

You can see that 9 communities were found, could be different for you. A random color has been set for each community identifier.

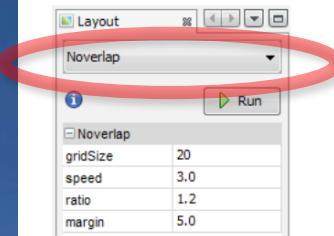
- Click on  to colorize nodes.



# Gephi layouts

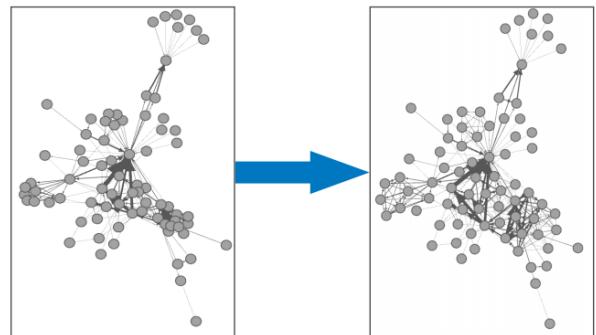
## Nooverlap layout

Use it after any layout to prevent node overlap while keeping the shape of the graph. It is optimized for big graphs.



- First, run the “YifanHu” layout.
- Select the “Nooverlap” algorithm and run it until it stops.
- Reduce the “speed” setting to 0.1 to increase quality.
- Increase the “ratio” at 2 and “margin” at 10 for more spacing around nodes.

You can see nodes are not overlapping anymore.



# Gephi layouts

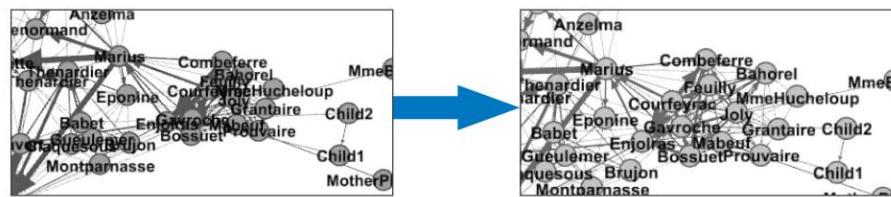
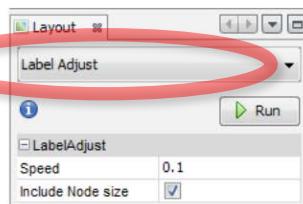
## Label Adjust layout

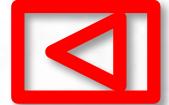
It works on text size to repulse nodes and therefore makes every label readable. It only runs on the visible nodes in the Visualization panel.

- Locate the Visualization settings.
- Click on to activate text display.
- Increase the text size to the maximum.



- Go to the Layout panel.
- Select the “Label Adjust” algorithm and run it until it stops.





Big Linked Data graphs with  
**GEPHI + DBPEDIA** (OR ANY OTHER SPARQL EP)

# Gephi

- Plugins mechanish
  - SemanticWebImport (for Gephi 0.9.2) not yet for versión 0.9.5 or 0.10
  - ~~Virtuoso Importer (for Gephi 0.8.2)~~
  - Gephi Graph Streaming
  - Neo4J Graph Database Support

# Gephi

- Plugins

Gephi Plugins

Gephi Plugins

SEE ALL

Search

DOWNLOAD GEPHI  
for WINDOWS

### Latest Plugins

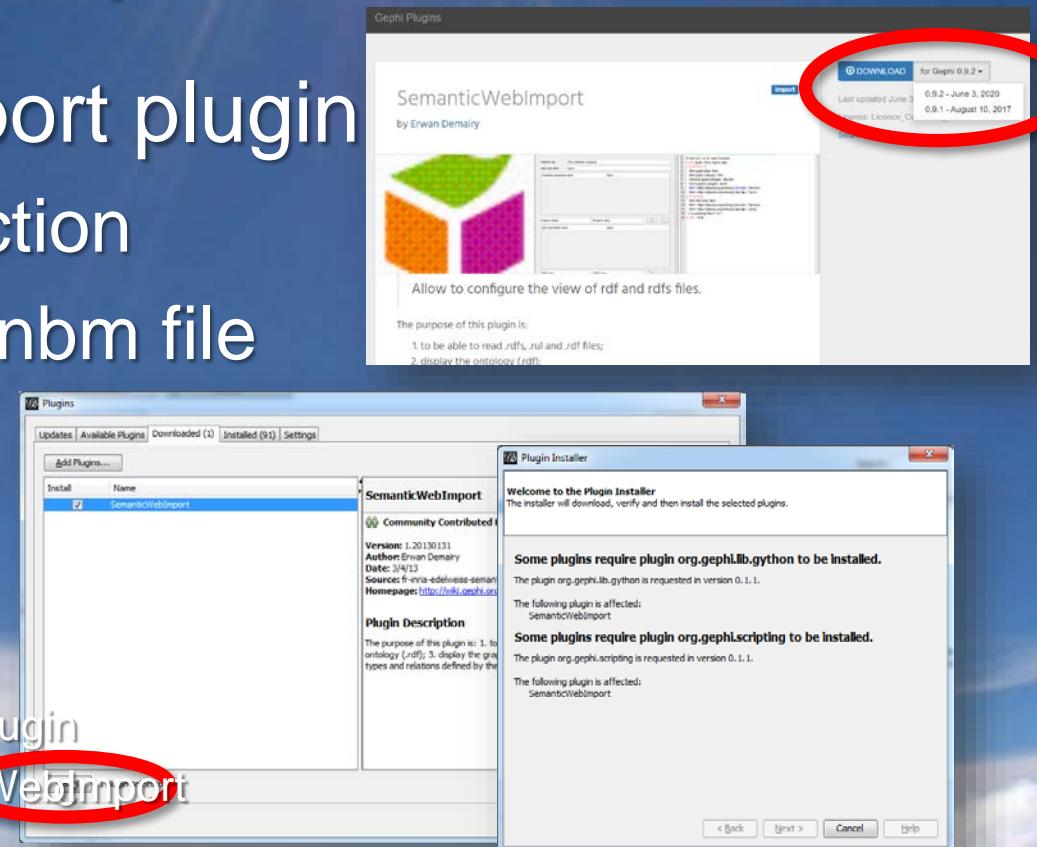
 GeoLayout 3 weeks ago A layout to display geocoded data	 Linkfluence Plugin 3 weeks ago Public plugin for Linkfluence	 ExportToEarth 3 weeks ago Export networks with geographic attributes to	 KBrace Filter 3 weeks ago K-brace filter which removes less embedded
---	---	---	--

### Browse by category

TYPE	LAYOUT	GEPHI VERSION
		0.9.3
TOOL		0.9.2
CLUSTERING		0.9.1
DATA LABORATORY		0.9.0
EXPORT		0.8.2
IMPORT		
FILTER		
GENERATOR		

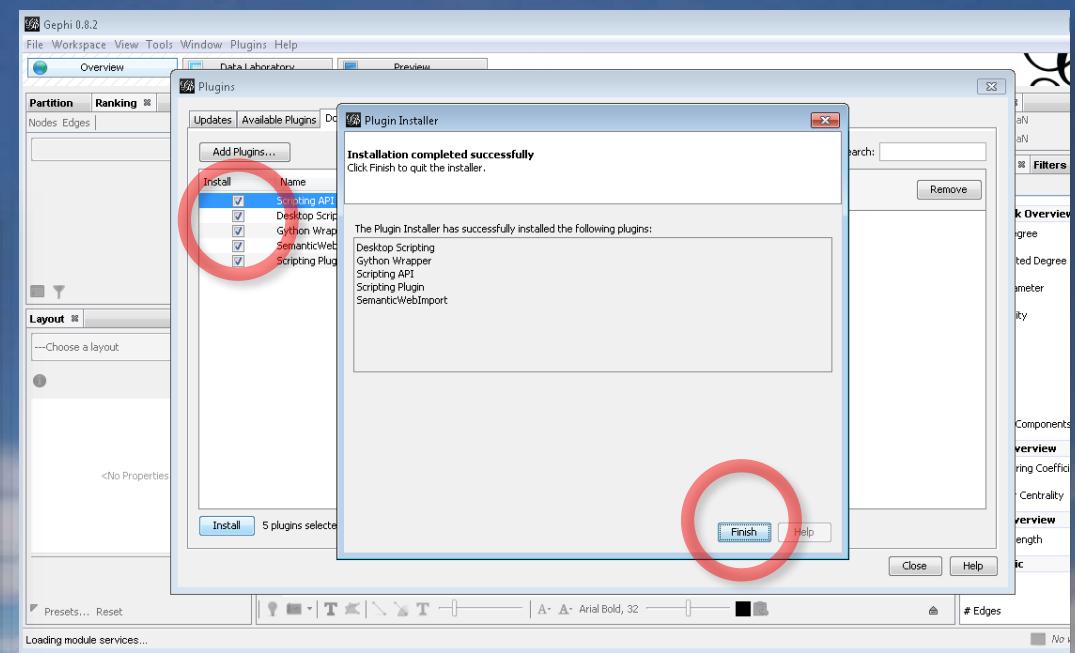
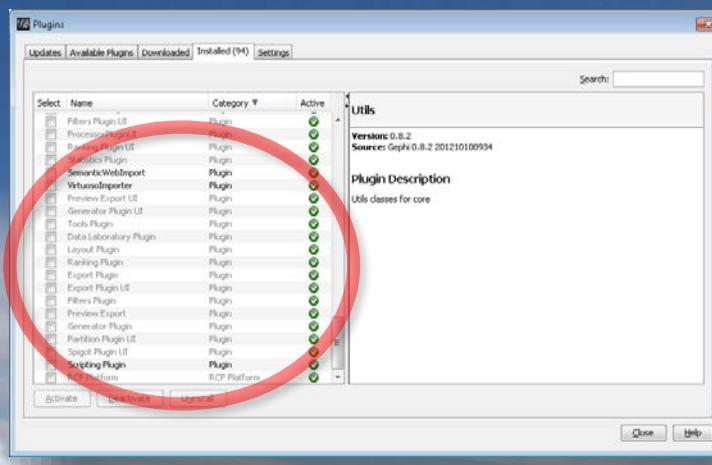
# Gephi

- SemanticWebImport plugin
  - In the Plugins section
  - Download it as a nbm file
  - Install it in Gephi
    - Tools→plugins→Downloaded → Add Plugins
      - ??Firstly: Scripting plugin
      - Secondly: SemanticWebImport



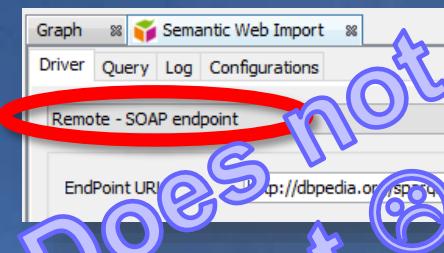
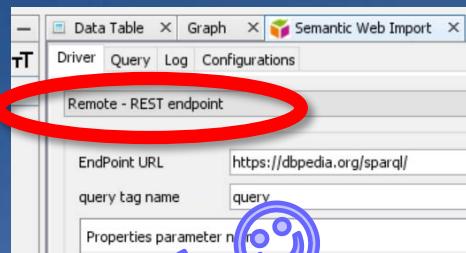
# Gephi

- Loading the Scripting Plugin
  - zip file → 4 nbm files



# Gephi 0.9.2

- Importing data from DBpedia
  - 1) Point to (Driver tab) <https://dbpedia.org/sparql/>



The screenshot shows the Gephi interface with the 'Semantic Web Import' tab selected. In the 'Query' tab, a SPARQL query is displayed in the 'SPARQL Query' section:

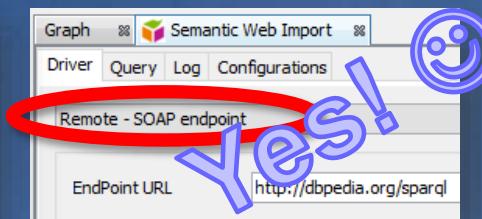
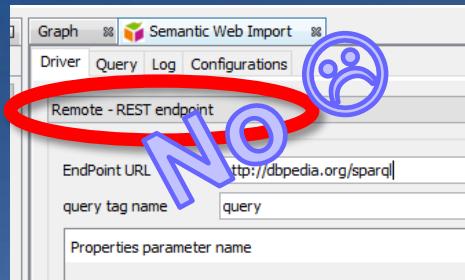
```
1 prefix gephi:<http://gephi.org/>
2 prefix foaf: <http://xmlns.com/foaf/0.1/>
3 CONSTRUCT{
4 ?philosopher gephi:label ?philosopherName .
5 ?influence gephi:label ?influenceName .
6 ?philosopher <http://dbpedia.org/ontology/influencedBy> ?influence
7 } WHERE {
8 ?philosopher a <http://dbpedia.org/ontology/Philosopher> .
9 ?philosopher <http://dbpedia.org/ontology/influencedBy> ?influence.
10 ?philosopher foaf:name ?philosopherName.
11 ?influence foaf:name ?influenceName.
12 }
```

Pay  
attention  
to the  
ending /

- 2) Make a  
SPARQL **CONSTRUCT** query to  
relate gephi vars with SPARQL vars

# Gephi 0.8.2

- Importing data from DBpedia
  - 0) Create a new (empty) workspace/project. **IMPORTANT!!**



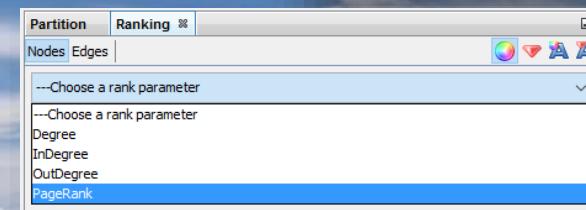
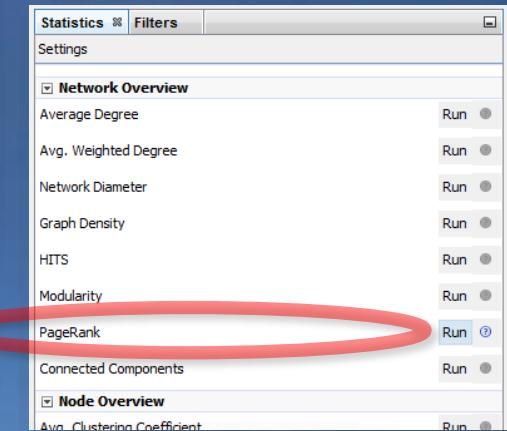
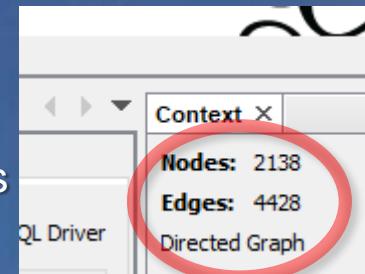
```
prefix gephi:<http://gephi.org/>
prefix foaf: <http://xmlns.com/foaf/0.1/>
CONSTRUCT{
  ?philosopher gephi:label ?philosopherName .
  ?influence gephi:label ?influenceName .
  ?philosopher <http://dbpedia.org/ontology/influencedBy> ?influence
} WHERE
  ?philosopher a <http://dbpedia.org/ontology/Philosopher> .
  ?philosopher <http://dbpedia.org/ontology/influencedBy> ?influence.
  ?philosopher foaf:name ?philosopherName.
  ?influence foaf:name ?influenceName.
```

- 1) Point to (Driver tab)  
[httpS://dbpedia.org/sparql/](http://dbpedia.org/sparql/)
- 2) Make a  
SPARQL **CONSTRUCT** query

# Gephi

- Importing data from DBpedia

- 3) Run!
  - You get 2138 nodes and 4428 edges.
- 4) Adjust visualization parameters
  - Run Statistics → PageRank
    - Default parameters



- Select Appearance → Nodes → Ranking  
Rank parameter = PageRank → Apply

# Gephi

- You should get something like this in the “log” tab

The screenshot shows the Gephi interface with the "Log" tab selected. The log window displays the following text:

```
ENT task finished -- INFO: Finished starting CreateGraphs. Time elapsed = 32,737 milliseconds
[agent startCreateGraphs -- INFO: Entering startCreateGraphs --
agent startCreateGraphs -- INFO: Starting the RDF importer for Gephi --
RDFGraph -- INFO: resetWorkspace = false --
RDFGraph -- INFO: postProcessing = false --
Loading the implementation relationships graph. --
y -- INFO: fr.inria.edelweiss.sparql.restdriver.SparqlRestEndPointDriver executing request: query=prefix+gephi%3A%3Chttp%3A%2F
y -- INFO: Result request: --
y -- INFO: <?xml version="1.0" encoding="utf-8" ?> --
y -- INFO: <rdf:RDF --
y -- INFO:     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" --
y -- INFO:     xmlns:dfs="http://www.w3.org/2000/01/rdf-schema#" --
y -- INFO:     xmlns:dbo="http://dbpedia.org/ontology/" --
y -- INFO:     xmlns:ns3="http://gephi.org/" > --
y -- INFO:     <rdf:Description rdf:about="http://dbpedia.org/resource/Adolf_Loos"> --
y -- INFO:     <ns3:label xml:lang="en">Adolf Loos</ns3:label> --
y -- INFO:     </rdf:Description> --
y -- INFO:     <rdf:Description rdf:about="http://dbpedia.org/resource/Adrienne_Rich"> --
y -- INFO: Result contains 11,260 lines. --
```

The first two lines of the log are circled in red. The entire log output is circled in red.

# Gephi

- If you cannot make it run

- See Log tab

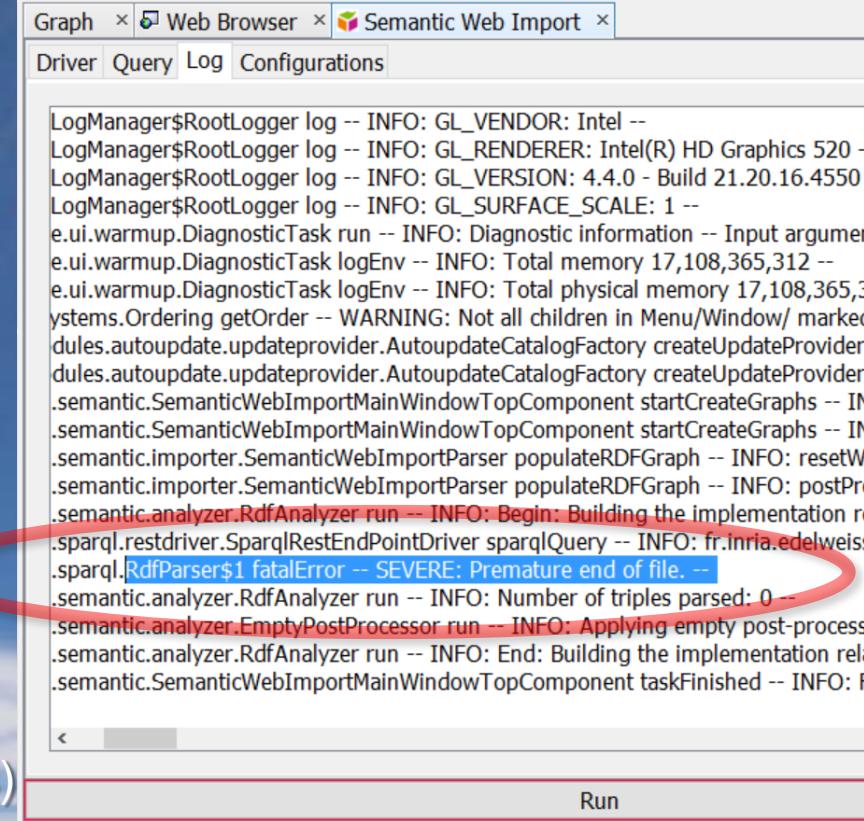
- “fatalError --

SEVERE:Premature  
end of file. --”

May be you forgot the trailing / in the URL  
of the SPARQL EP

Look for invalid chars in the query (e.g. tabs)

May be you have to update the plugins



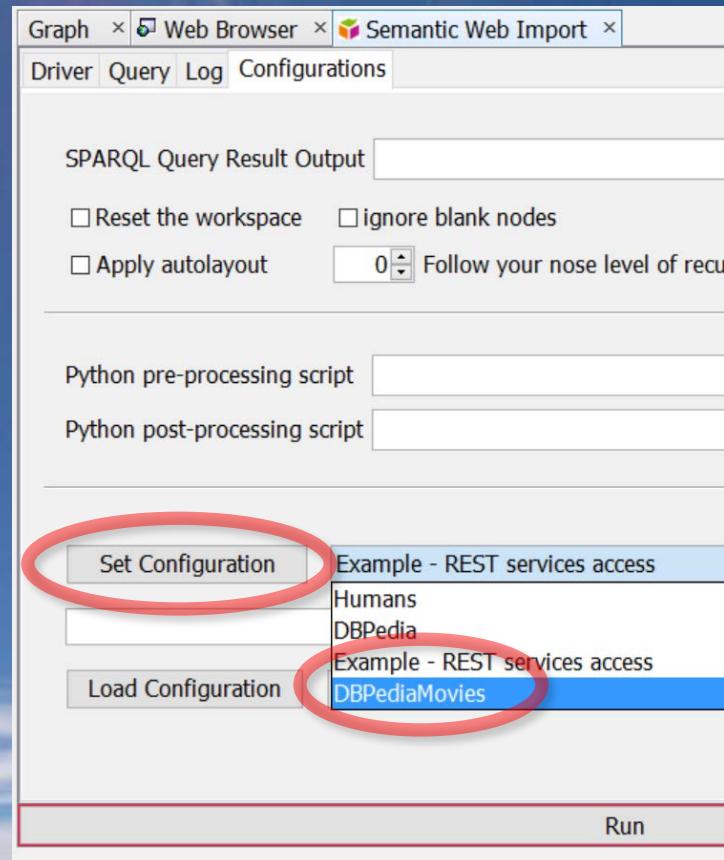
The screenshot shows the Gephi interface with the "Log" tab selected in the top navigation bar. The log window displays several lines of log output, with the last few lines circled in red. The circled text reads:

```
LogManager$RootLogger log -- INFO: GL_VENDOR: Intel --
LogManager$RootLogger log -- INFO: GL_RENDERER: Intel(R) HD Graphics 520 -
LogManager$RootLogger log -- INFO: GL_VERSION: 4.4.0 - Build 21.20.16.4550
LogManager$RootLogger log -- INFO: GL_SURFACE_SCALE: 1 --
e.ui.warmup.DiagnosticTask run -- INFO: Diagnostic information -- Input argumen
e.ui.warmup.DiagnosticTask logEnv -- INFO: Total memory 17,108,365,312 --
e.ui.warmup.DiagnosticTask logEnv -- INFO: Total physical memory 17,108,365,3
systems.Ordering getOrder -- WARNING: Not all children in Menu/Window/ marked
dules.autoupdate.updateprovider.AutoupdateCatalogFactory createUpdateProvider
dules.autoupdate.updateprovider.AutoupdateCatalogFactory createUpdateProvider
.semantic.SemanticWebImportMainWindowTopComponent startCreateGraphs -- IN
.semantic.SemanticWebImportMainWindowTopComponent startCreateGraphs -- IN
.semantic.importer.SemanticWebImportParser populateRDFGraph -- INFO: resetW
.semantic.importer.SemanticWebImportParser populateRDFGraph -- INFO: postPro
.semantic.analyzer.RdfAnalyzer run -- INFO: Begin: Building the implementation re
.sparql.restdriver.SparqlRestEndPointDriver sparqlQuery -- INFO: fr.inria.edelweiss
.sparql.RdfParser$1 fatalError -- SEVERE: Premature end of file. --
.semantic.analyzer.RdfAnalyzer run -- INFO: Number of triples parsed: 0 --
.semantic.analyzer.EmptyPostProcessor run -- INFO: Applying empty post-process
.semantic.analyzer.RdfAnalyzer run -- INFO: End: Building the implementation rela
.semantic.SemanticWebImportMainWindowTopComponent taskFinished -- INFO: F
```

A red box highlights the "Run" button at the bottom of the log window.

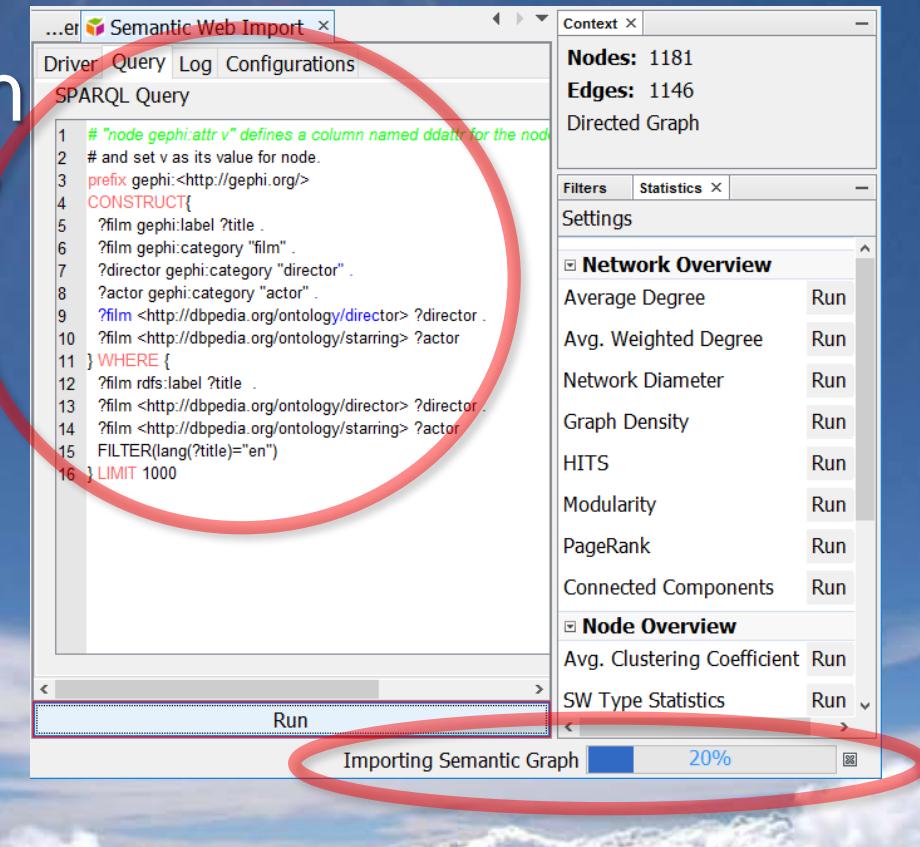
# Gephi

- If you cannot make it run
  - Use a “pre configuration”
    - Select DBPediaMovies
    - Press button “Set configuration” (you can see the CONSTRUCT query in the “Query” tab)
    - Change http → https
    - Check the trailing /



# Gephi

- If you cannot make it run
  - Use a “pre configuration”
    - Select DBPediaMovies
    - Press button “Set configuration” (you can see the CONSTRUCT query in the “Query” tab)
    - Change http → https
    - Check the trailing /

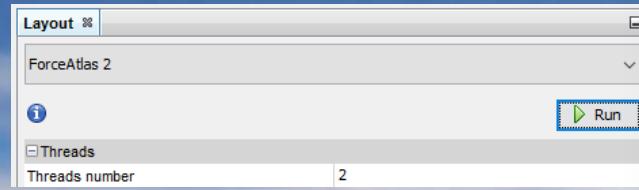


# Gephi

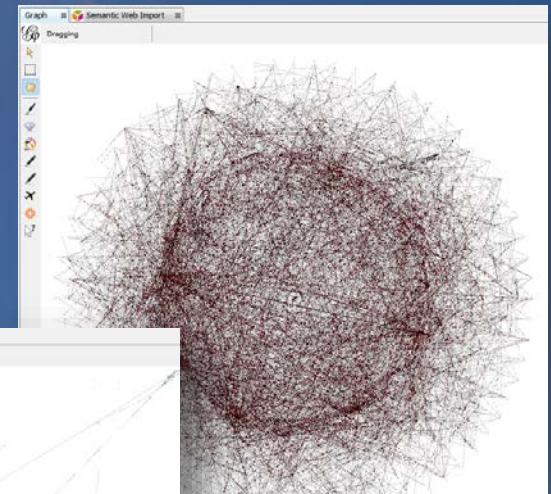
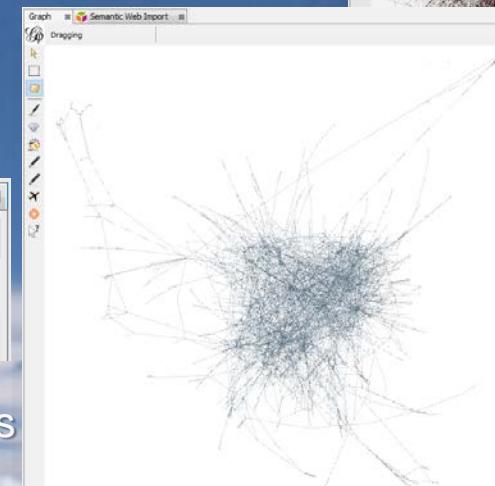
- Importing data from DBpedia

- Check that you have something like this  
(Show/Hide the Graph tab with Window → Graph)

- Layout → Force Atlas 2 → Run (~10 seconds)



- You should get something like this

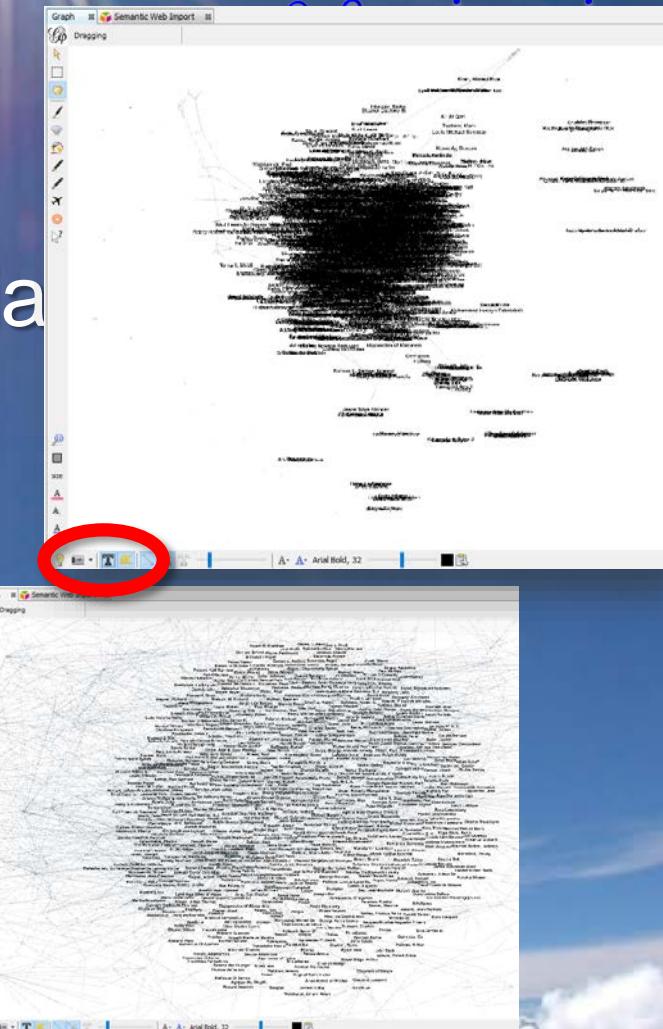


# Gephi

- Importing data from DBpedia

- Show node labels  
(Clic icon  in the bottom of the Graph tab)

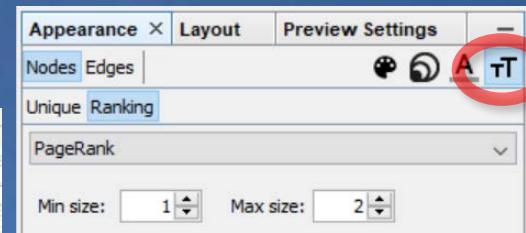
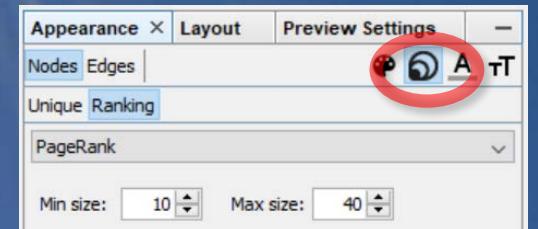
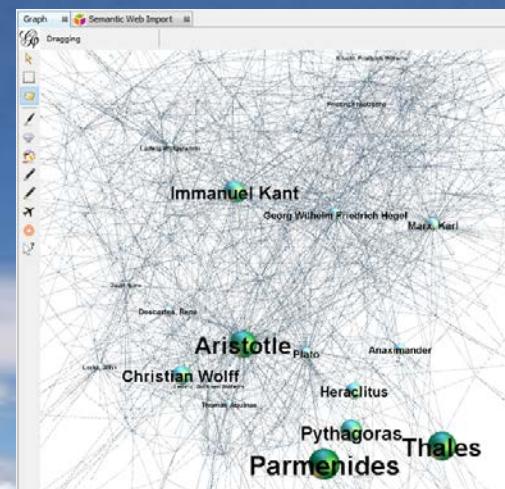
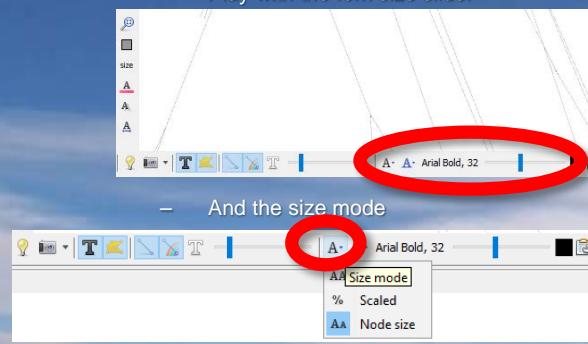
- Layout → Expansion → Run three times
- Layout → Label Adjust → Run ~10 seconds
- You should get some thing like this



# Gephi

- Importing data from DBpedia

- Node size proportional to ranking  
(Ranking→Nodes→ clic icon  → set min & max → Apply)
- Node label size proportional to ranking  
(Ranking→Nodes→clic icon  → set min & max → Apply)
- You should get some thing like this
  - Play with the font size slider





# Gephi

- Importing data from Wikidata

- 1) Use the Wikidata Query Service (SPARQL queries)
  - Select some example
  - Run the query

The screenshot shows the Wikidata Query Service interface. In the search bar, the word "marvel" is typed. Below the search bar, a red circle highlights the result "Fictional subjects of the Marvel Universe". At the bottom of the interface, another red circle highlights the "Run" button.

The screenshot shows the results of the SPARQL query "SELECT ?char ?charName ?types WHERE { ?char wdt:P1800 wd:Q22287; wdt:P117 ?type; wdt:P1889 ?universe . SERVICE wikibase:label { bd:serviceParam wikibase:language \"[AUTO\_LANGUAGE]\" } ; ?char rdfs:label ?charName ; ?universe rdfs:label ?universeLabel . ?type rdfs:label ?typeLabel } GROUP BY ?char ?charName". The results table has columns for char, charName, types, and universes. A red circle highlights the "2000 results in 10.7M ms" message at the bottom right of the results table.

char	charName	types	universes
Q1257894	Doctor Octopus	mutant, superhero film character	Marvel Universe, Earth-616
Q12578279	Firestar	mutant	Marvel Universe
Q12578371	Dark Avengers	group of fictional characters, fictional organization	Marvel Universe
Q12580374	David North	mutant, animated character, superhero film character	Marvel Universe
Q12580322	Many MacPharlane	comic book character	Marvel Universe
Q12584985	Human Torch	mutant, fictional human, animated character, film character	Marvel Universe, Earth-616

# Gephi

- Importing data from Wikidata

2) Create a SPARQL query with CONSTRUCT to create gephi data

Notice: many prefixes known by the Wikidata EP

3) Point to the Wikidata endpoint <https://query.wikidata.org/sparql/>

Based on: [sbalci repo at github](#)

The screenshot shows the Gephi interface with the "Semantic Web Import" tab selected. On the left, a code editor displays a SPARQL query:

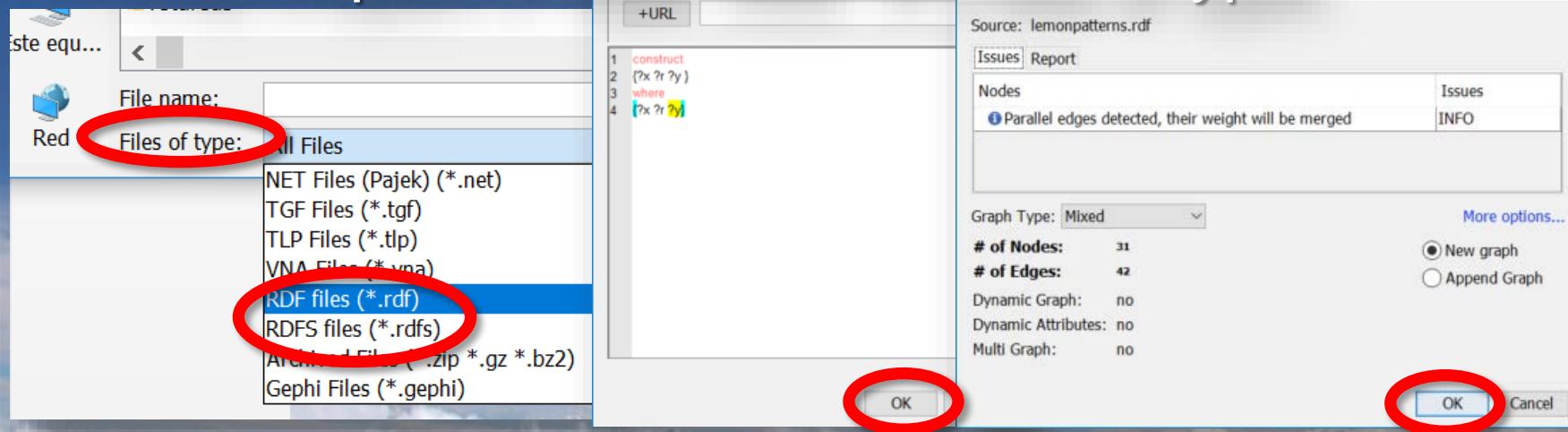
```
1 #Groups of characters in the Marvel universe
2 PREFIX geph: <http://gephi.org/> #This line is missing in sbalci post!!!
3 CONSTRUCT {
4 ?group geph:label ?groupLabel .
5 ?group geph:category "group" .
6 ?char geph:label ?charLabel .
7 ?char geph:category "character" .
8 ?char wdt:P463 ?group .
9 } WHERE {
10 ?group wdt:P31 wd:Q14514600 ; # group of fictional characters
11 wdt:P1080 wd:Q931597. # from Marvel universe
12 ?char wdt:P463 ?group # member of group
13 SERVICE wikibase:label { bd:serviceParam wikibase:language "en".}
14 }
```

The code editor has several red highlights: a red circle around the line "#This line is missing in sbalci post!!!", a red box around the CONSTRUCT block, and a red box around the WHERE block. At the bottom right of the interface, there is a progress bar labeled "Importing Semantic Graph" with a value of "20%".



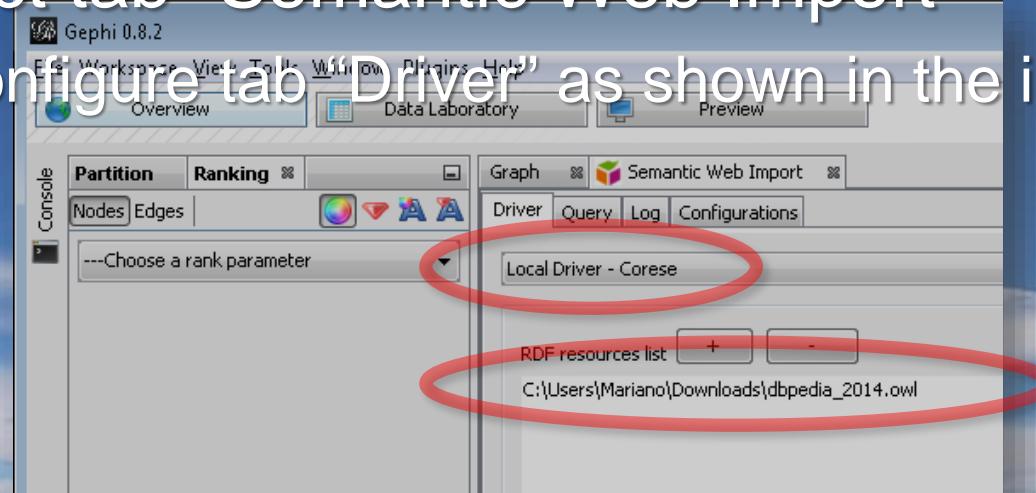
# Gephi

- Gephi can import RDF (not .owl, see later)
  - Requires the plugin “Semantic Web Import”
  - File → Open → "RDF files" in Files of type:



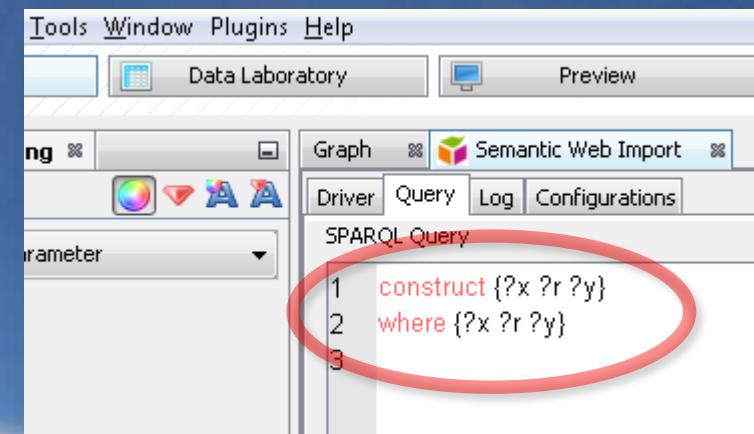
# Gephi

- Other example: DBpedia ontology 2014
  - Load local file (dbpedia\_2014.owl)
  - Select tab “Semantic Web Import”
    - Configure tab “Driver” as shown in the image



# Gephi

- Load local file (dbpedia\_2014.owl)
  - 2) Click tab “Query”
    - Remove the triples limit (100 by default)



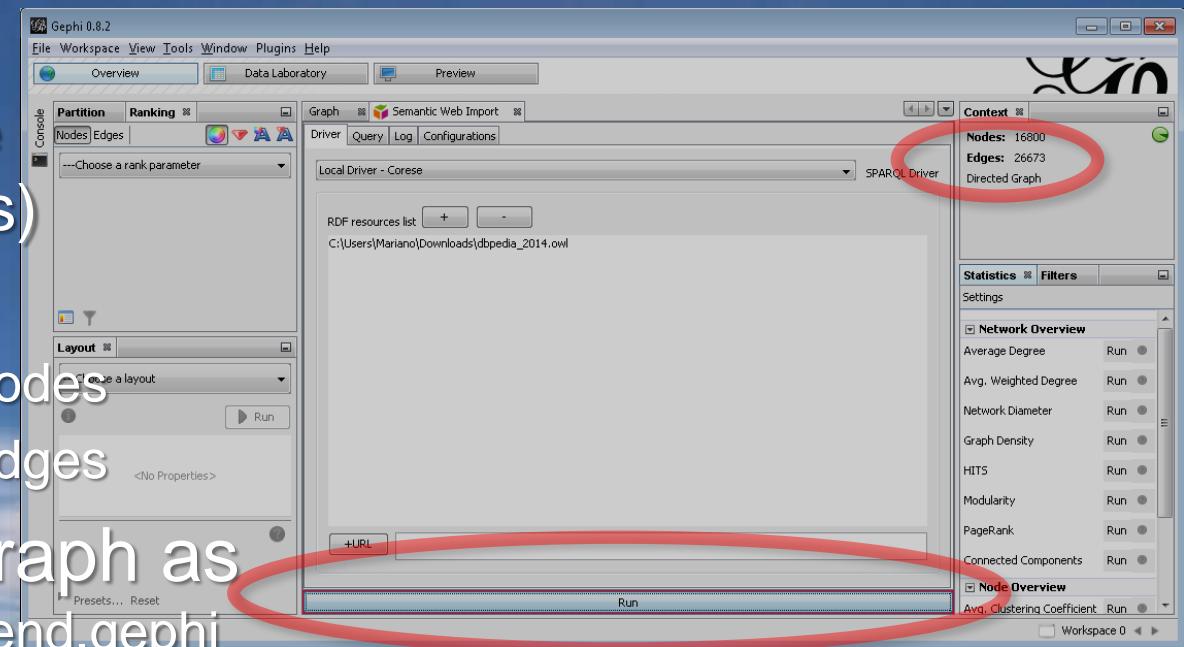
# Gephi

- Load local file (dbpedia\_2014.owl)

## 3) Run

- Takes time  
(a few mins)
- You get
  - 16,800 nodes
  - 26,673 edges

## 4) Save the graph as dbpedia\_2014.end.gephi

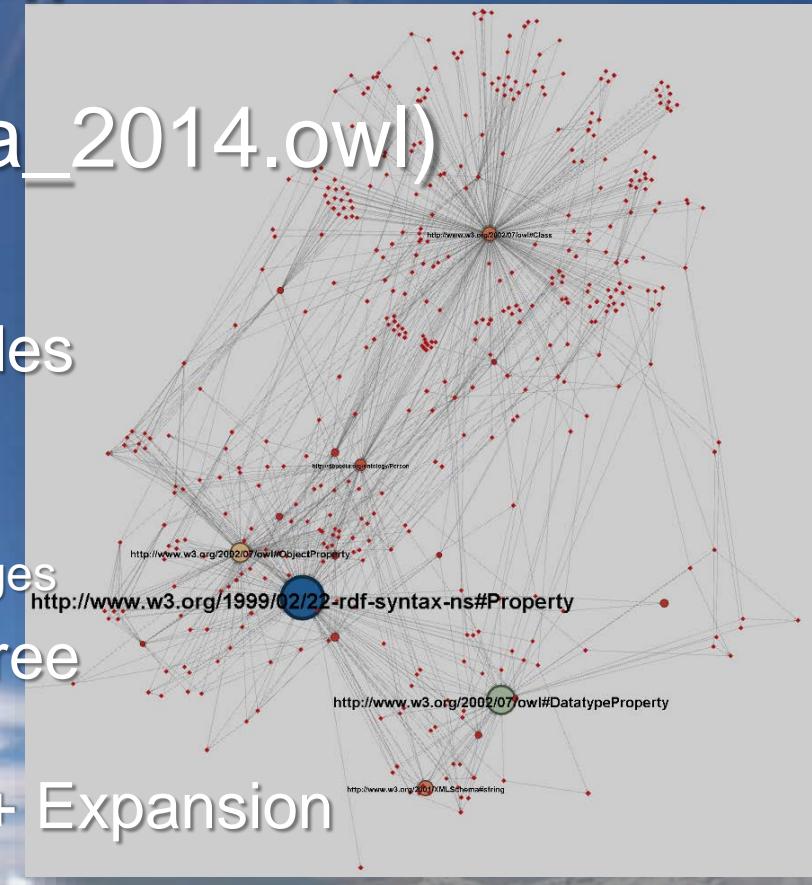


# Gephi

- Load local file (dbpedia\_2014.owl)

Result after

- Filter nodes. “Remove” nodes with degree < 10
  - Makes visible only 2.76% nodes and 4.5 edges
- Node size and color ~ Degree (min size = 5 , max size = 50)
- Fuchterman + Force atlas + Expansion

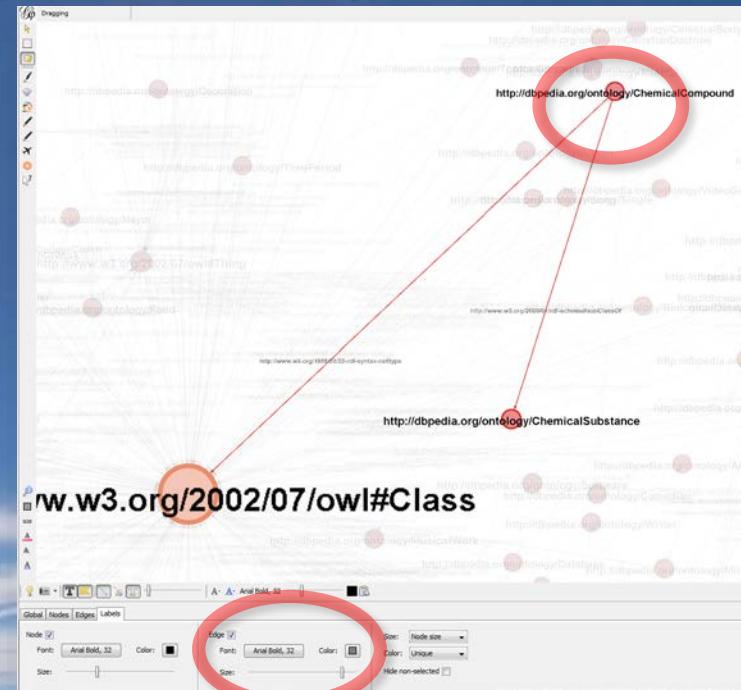


# Gephi

- Load local file (dbpedia\_2014.owl)

## Result analysis

- Zoom in into classes
  - Show edge labels
  - Hover a class  
(ChemicalCompound in the figure)

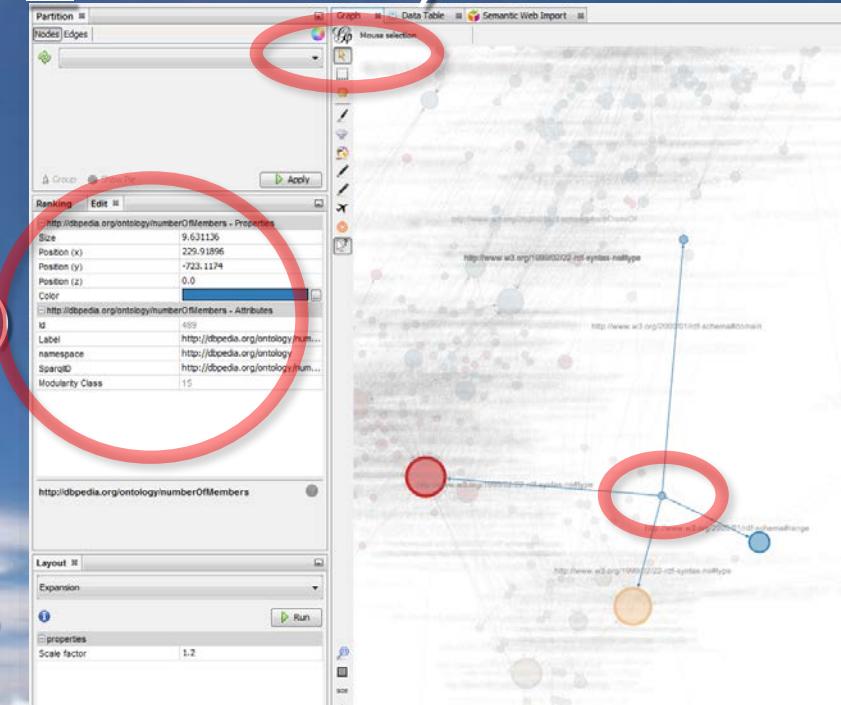


# Gephi

- Load local file (dbpedia\_2014.owl)

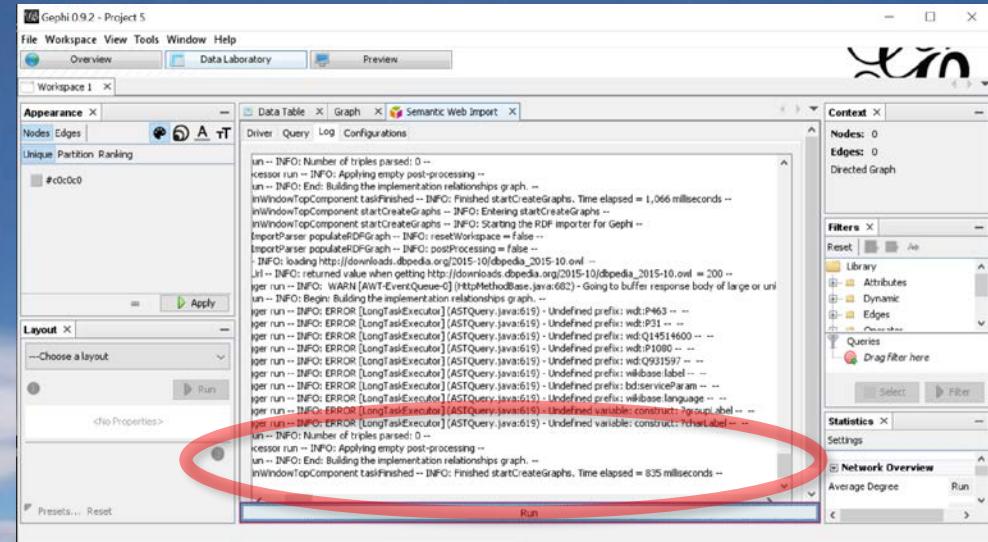
## Result analysis

- Use the selection tool 
- Click on a node to see its properties in a tab (“Edit tab”)
- Hide node labels
- Increase edges’ label size
- Now you see the node’s name in the “Edit tab” and its edges labels



# Gephi

- Load online file
  - e.g. [http://downloads.dbpedia.org/2015-10/dbpedia\\_2015-10.owl](http://downloads.dbpedia.org/2015-10/dbpedia_2015-10.owl)
- Execution error
  - At least on Windows with Gephi 0.9.2 ☹



# Gephi

- Load local owl file
  - e.g. [http://downloads.dbpedia.org/2016-10/dbpedia\\_2016-10.owl](http://downloads.dbpedia.org/2016-10/dbpedia_2016-10.owl)
- Goes fine
  - At least on Windows ☺  
although it takes ~10 min to load ☹

# Gephi. Load a csv file (1/3)

- e.g. DBpedia classes  
(with format subclass, class)
  - Use File→Open
  - Select the csv file
  - (opens a wizard)

The screenshot shows the Gephi CSV import wizard. At the top, a preview window displays the contents of 'DBclasses.csv' with 10 rows of data:

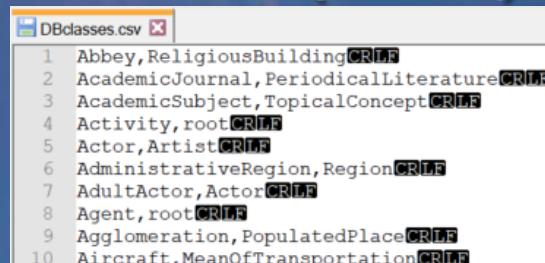
1	Abbey, ReligiousBuilding	CRLF
2	AcademicJournal, PeriodicalLiterature	CRLF
3	AcademicSubject, TopicalConcept	CRLF
4	Activity, root	CRLF
5	Actor, Artist	CRLF
6	AdministrativeRegion, Region	CRLF
7	AdultActor, Actor	CRLF
8	Agent, root	CRLF
9	Agglomeration, PopulatedPlace	CRLF
10	Aircraft, MeanOfTransportation	CRLF

Below the preview, the 'General CSV options (1 of 2)' step is selected. It includes fields for 'CSV file to import' (set to 'C:\Users\...'), 'Separator' (set to 'Comma'), 'Import as' (set to 'Adjacency list'), and 'Charset' (set to 'UTF-8'). A preview table shows the first few rows of the imported data.

At the bottom, there are navigation buttons: < Back, Next >, Finish, Cancel, and Help.

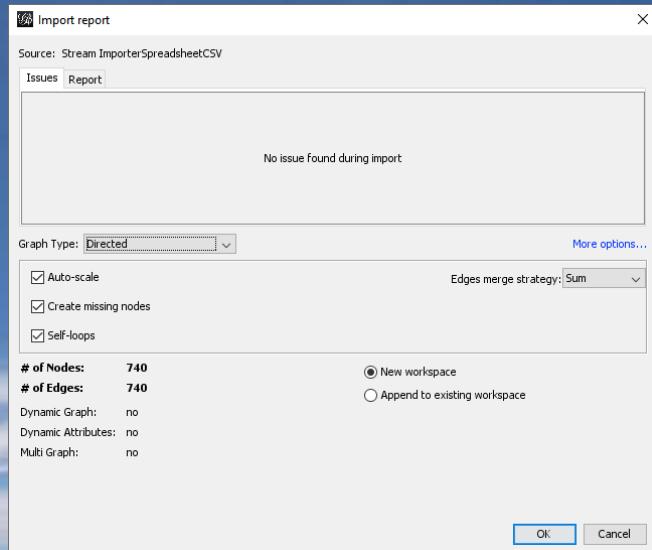
# Gephi. Load a csv file (2/3)

- e.g. DBpedia classes  
(with format subclass, class)
  - Check the “import report”



DBclasses.csv

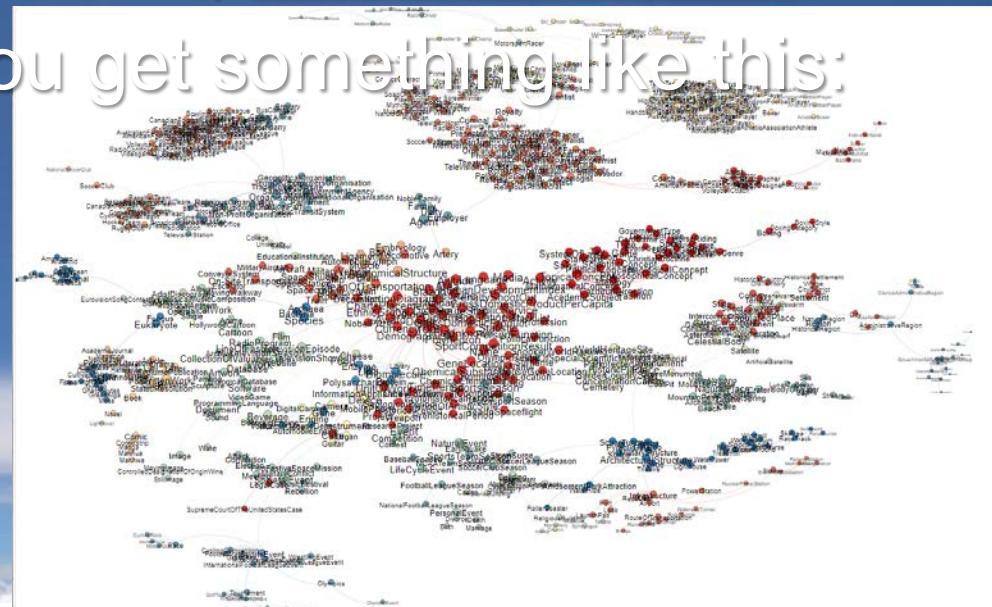
```
1 Abbey,ReligiousBuilding
2 AcademicJournal,PeriodicalLiterature
3 AcademicSubject,TopicalConcept
4 Activity,root
5 Actor,Artist
6 AdministrativeRegion,Region
7 AdultActor,Actor
8 Agent,root
9 Agglomeration,PopulatedPlace
10 Aircraft,MeanOfTransportation
```



# Gephi. Load a csv file (3/3)

- e.g. DBpedia classes  
(with format subclass, class)
  - After some work you get something like this:

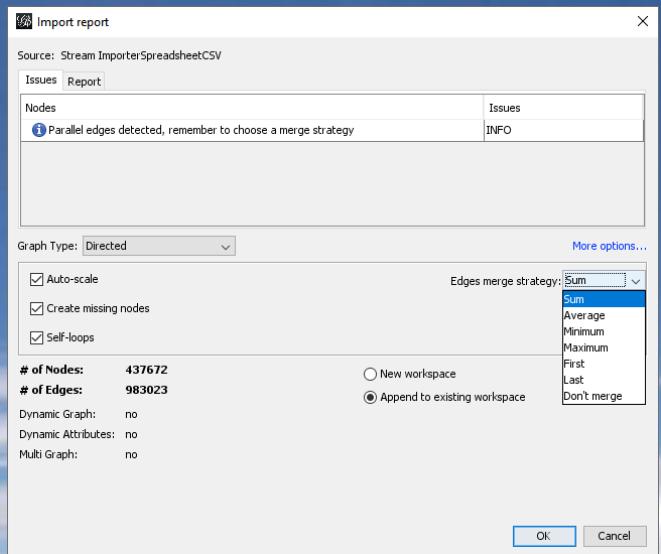
DBclasses.csv	x
1	Abbey, ReligiousBuilding
2	AcademicJournal, PeriodicalLiterature
3	AcademicSubject, TopicalConcept
4	Activity, root
5	Actor, Artist
6	AdministrativeRegion, Region
7	AdultActor, Actor
8	Agent, root
9	Agglomeration, PopulatedPlace
10	Aircraft, MeanOfTransportation



# Gephi. Load a csv file

- e.g. DBpedia categories  
(with format `subcat;cat`)
  - Graph  $10^3$  bigger than classes graph
    - Warning!!
      - Parallel edges detected!!

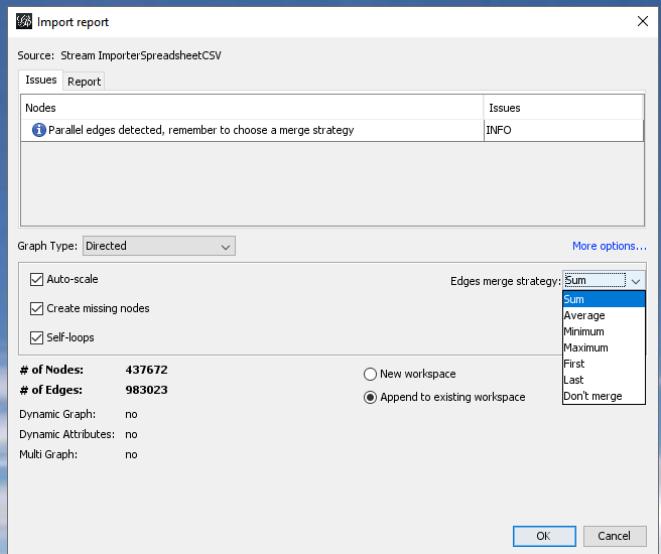
```
1 'N_Sync;Categorías_de_grupos_musicales_de_Estados_ UnidosLF
2 'Ndrangheta;Historia_de_CalabriaIT
3 'Ndrangheta;Organizaciones_delictivas_de_ItaliaLF
4 'Ndrangheta;Sociedades_secretas_criminalesIT
5 (G) I-dle;Categorías_de_grupos_musicales_de_Corea_del_SurLF
6 (G) I-dle;Grupos_de_pop_de_Corea_del_SurLF
7 .hack;Categorías_de_series_de_anime_y_mangaLF
8 .hack;Franquicias_de_videojuegosLF
9 .hack;Realidad_simulada_en_la_ficciónLF
10 .hack;Realidad_virtual_en_ficciónIT
11 .hack;Series_de_anime_de_Bandai_VisualIT
12 1. Bundesliga;Campeonatos_de_fútbol_entre_clubes_de_AlemaniaLF
```



# Gephi. Load a csv file

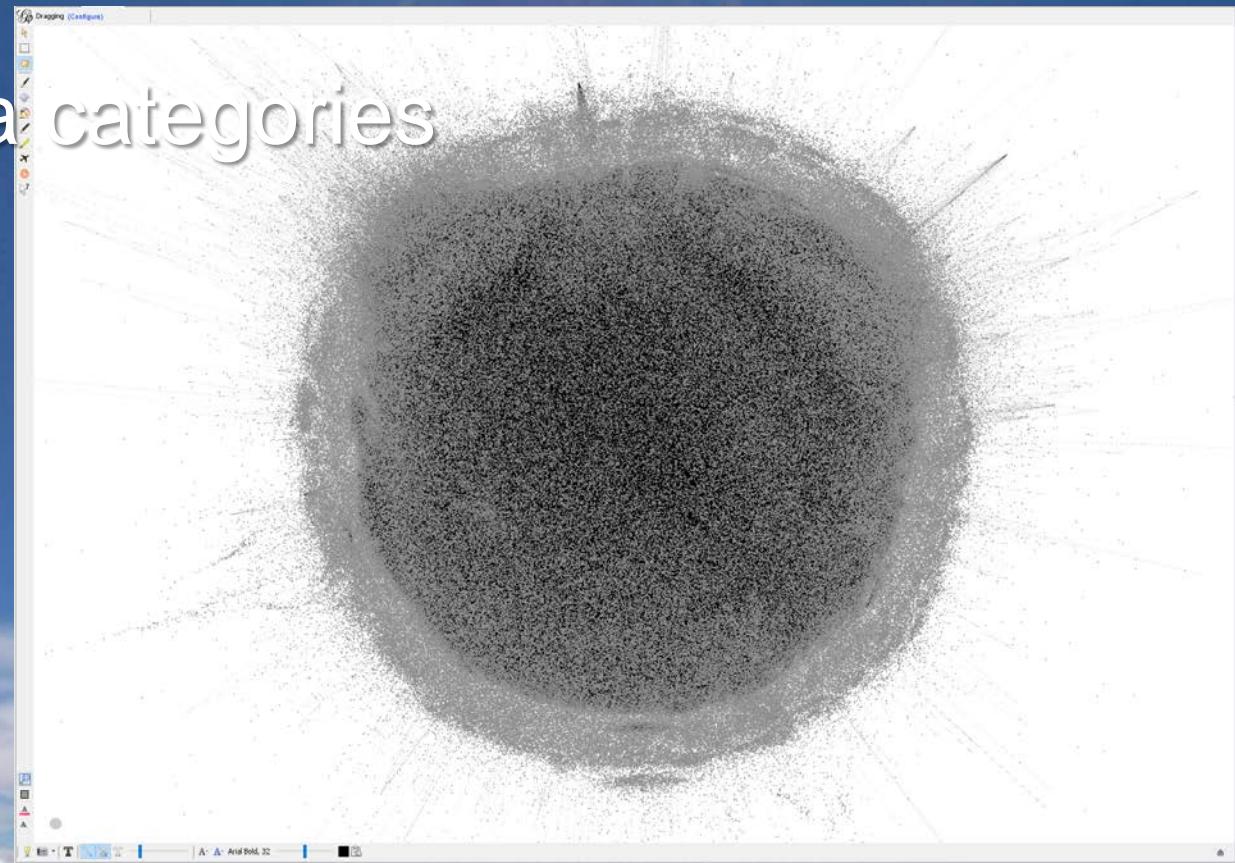
- e.g. DBpedia categories  
(with format `subcat;cat`)
  - Graph  $10^3$  bigger than classes graph
    - Warning!!
      - Parallel edges detected!!

```
1 'N_Sync;Categorías_de_grupos_musicales_de_Estados_ UnidosLF
2 'Ndrangheta;Historia_de_CalabriaIT
3 'Ndrangheta;Organizaciones_delictivas_de_ItaliaLF
4 'Ndrangheta;Sociedades_secretas_criminalesIT
5 (G) I-dle;Categorías_de_grupos_musicales_de_Corea_del_SurLF
6 (G) I-dle;Grupos_de_pop_de_Corea_del_SurLF
7 .hack;Categorías_de_series_de_anime_y_mangaLF
8 .hack;Franquicias_de_videojuegosLF
9 .hack;Realidad_simulada_en_la_ficciónLF
10 .hack;Realidad_virtual_en_ficciónIT
11 .hack;Series_de_anime_de_Bandai_VisualIT
12 1. Bundesliga;Campeonatos_de_fútbol_entre_clubes_de_AlemaniaLF
```



# Gephi. Load a csv file

- e.g. DBpedia categories



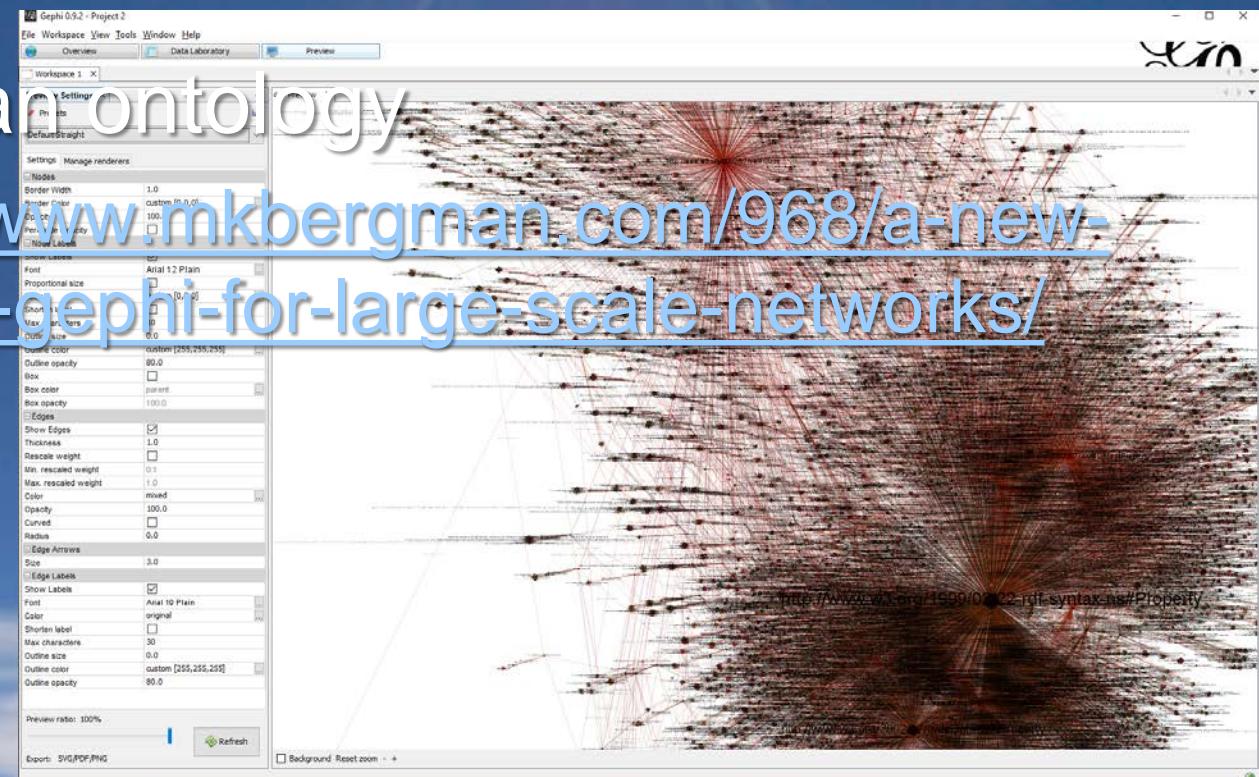
# Gephi. Load a csv file

- e.g. DBpedia categories



# Gephi

- Displaying a ontology
  - See <http://www.mkbergman.com/968/a-new-best-friend-gephi-for-large-scale-networks/>



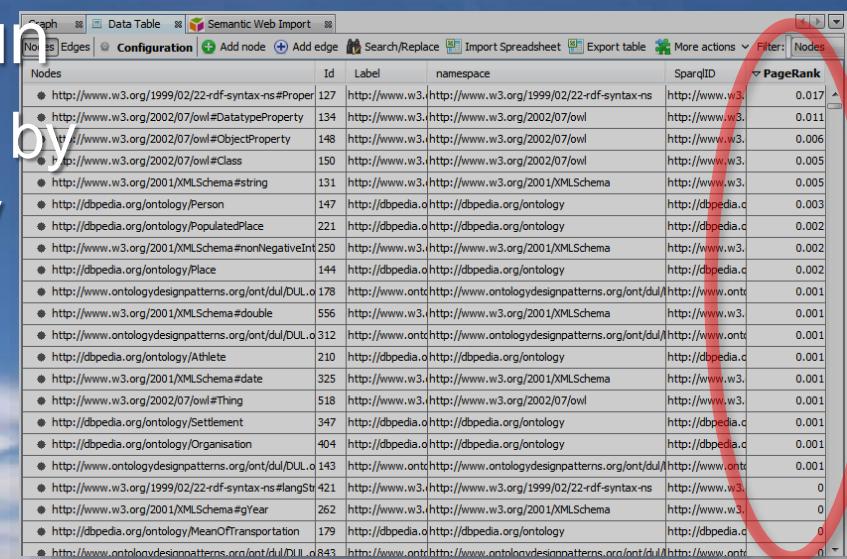
# Gephi

- The DBpedia Ontology use case
  - Load the file dbpedia\_2014.init.gephi
    - It is a directed graph with nodes, edges and labels
    - Statistics → PageRank → Run
      - See the data table, sort by PageRank
      - Notice that only 19 nodes have a value != 0 →  
**Not a valid measure**



# Gephi

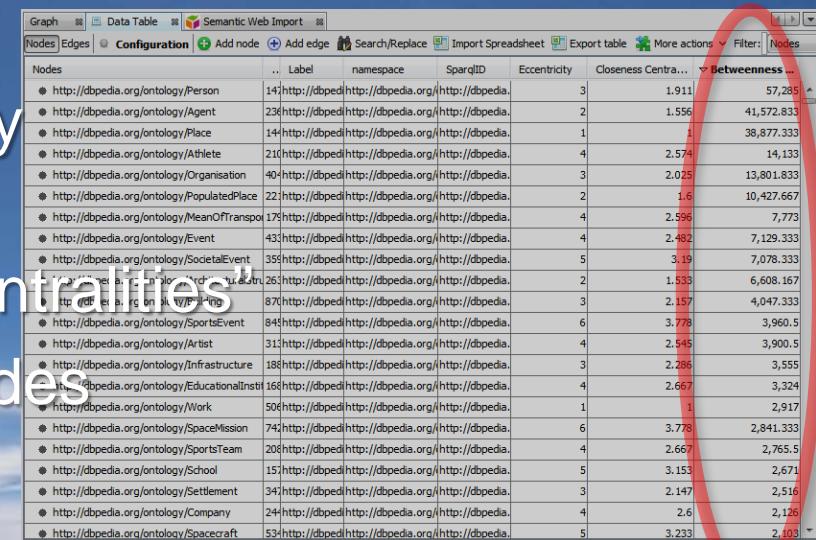
- The DBpedia Ontology use case
  - Statistics → Edge Overview → Avg. Path Length → Run
    - See the data table, sort by Betweenness Centrality
    - Tic normalize
    - Notice that ALL nodes have a value == 0 → Not a valid measure



Nodes	Id	Label	namespace	SparqlID	PageRank
● http://www.w3.org/1999/02/22-rdf-syntax-ns#Property	127	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property	http://www.w3.org/1999/02/22-rdf-syntax-ns#	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property	0.017
● http://www.w3.org/2002/07/owl#DatatypeProperty	134	http://www.w3.org/2002/07/owl#DatatypeProperty	http://www.w3.org/2002/07/owl#	http://www.w3.org/2002/07/owl#DatatypeProperty	0.011
● http://www.w3.org/2002/07/owl#ObjectProperty	148	http://www.w3.org/2002/07/owl#ObjectProperty	http://www.w3.org/2002/07/owl#	http://www.w3.org/2002/07/owl#ObjectProperty	0.006
● http://www.w3.org/2002/07/owl#Class	150	http://www.w3.org/2002/07/owl#Class	http://www.w3.org/2002/07/owl#	http://www.w3.org/2002/07/owl#Class	0.005
● http://www.w3.org/2001/XMLSchema#string	131	http://www.w3.org/2001/XMLSchema#string	http://www.w3.org/2001/XMLSchema#	http://www.w3.org/2001/XMLSchema#string	0.005
● http://dbpedia.org/ontology/Person	147	http://dbpedia.org/ontology/Person	http://dbpedia.org/ontology	http://dbpedia.org/ontology/Person	0.003
● http://dbpedia.org/ontology/PopulatedPlace	221	http://dbpedia.org/ontology/PopulatedPlace	http://dbpedia.org/ontology	http://dbpedia.org/ontology/PopulatedPlace	0.002
● http://www.w3.org/2001/XMLSchema#nonNegativeInteger	250	http://www.w3.org/2001/XMLSchema#nonNegativeInteger	http://www.w3.org/2001/XMLSchema#	http://www.w3.org/2001/XMLSchema#nonNegativeInteger	0.002
● http://dbpedia.org/ontology/Place	144	http://dbpedia.org/ontology/Place	http://dbpedia.org/ontology	http://dbpedia.org/ontology/Place	0.002
● http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	178	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	http://www.ontologydesignpatterns.org/ont/dul/	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	0.001
● http://www.w3.org/2001/XMLSchema#double	556	http://www.w3.org/2001/XMLSchema#double	http://www.w3.org/2001/XMLSchema#	http://www.w3.org/2001/XMLSchema#double	0.001
● http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	312	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	http://www.ontologydesignpatterns.org/ont/dul/	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	0.001
● http://dbpedia.org/ontology/Athlete	210	http://dbpedia.org/ontology/Athlete	http://dbpedia.org/ontology	http://dbpedia.org/ontology/Athlete	0.001
● http://www.w3.org/2001/XMLSchema#date	325	http://www.w3.org/2001/XMLSchema#date	http://www.w3.org/2001/XMLSchema#	http://www.w3.org/2001/XMLSchema#date	0.001
● http://www.w3.org/2002/07/owl#Thing	518	http://www.w3.org/2002/07/owl#Thing	http://www.w3.org/2002/07/owl#	http://www.w3.org/2002/07/owl#Thing	0.001
● http://dbpedia.org/ontology/Settlement	347	http://dbpedia.org/ontology/Settlement	http://dbpedia.org/ontology	http://dbpedia.org/ontology/Settlement	0.001
● http://dbpedia.org/ontology/Organisation	404	http://dbpedia.org/ontology/Organisation	http://dbpedia.org/ontology	http://dbpedia.org/ontology/Organisation	0.001
● http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	143	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	http://www.ontologydesignpatterns.org/ont/dul/	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	0.001
● http://www.w3.org/1999/02/22-rdf-syntax-ns#langString	421	http://www.w3.org/1999/02/22-rdf-syntax-ns#langString	http://www.w3.org/1999/02/22-rdf-syntax-ns#	http://www.w3.org/1999/02/22-rdf-syntax-ns#langString	0
● http://www.w3.org/2001/XMLSchema#gYear	262	http://www.w3.org/2001/XMLSchema#gYear	http://www.w3.org/2001/XMLSchema#	http://www.w3.org/2001/XMLSchema#gYear	0
● http://dbpedia.org/ontology/MeanOfTransportation	179	http://dbpedia.org/ontology/MeanOfTransportation	http://dbpedia.org/ontology	http://dbpedia.org/ontology/MeanOfTransportation	0
● http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	843	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	http://www.ontologydesignpatterns.org/ont/dul/	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	0

# Gephi

- The DBpedia Ontology use case
  - Statistics → Edge Overview → Avg. Path Length → Run
    - See the data table, sort by Betweenness Centrality
    - Do NOT tick “normalize centralities”
    - Notice that now, more nodes have a value != 0 → **Do NOT normalize!!**



Nodes	... Label	namespace	SparqlID	Eccentricity	Closeness Centra...	Betweenness ...
● http://dbpedia.org/ontology/Person	14	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	3	1.911	57,285
● http://dbpedia.org/ontology/Agent	23	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	2	1.556	41,572,833
● http://dbpedia.org/ontology/Place	14	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	1	1	38,877,333
● http://dbpedia.org/ontology/Athlete	210	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	4	2.574	14,133
● http://dbpedia.org/ontology/Organisation	40	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	3	2.025	13,801,833
● http://dbpedia.org/ontology/PopulatedPlace	22	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	2	1.6	10,427,667
● http://dbpedia.org/ontology/MeansOfTranspo	179	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	4	2.596	7,773
● http://dbpedia.org/ontology/Event	43	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	4	2.482	7,129,333
● http://dbpedia.org/ontology/SocialEvent	359	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	5	3.19	7,078,333
● http://dbpedia.org/ontology/GeographicEntity	14	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	2	1.533	6,608,167
● http://dbpedia.org/ontology/Thing	1,436	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	3	2.157	4,047,333
● http://dbpedia.org/ontology/VIPListing	870	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	3	2.157	4,047,333
● http://dbpedia.org/ontology/SportsEvent	843	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	6	3.778	3,960,5
● http://dbpedia.org/ontology/Artist	313	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	4	2.545	3,900,5
● http://dbpedia.org/ontology/Infrastructure	188	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	3	2.286	3,555
● http://dbpedia.org/ontology/EducationalInstitution	158	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	4	2.667	3,324
● http://dbpedia.org/ontology/Work	506	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	1	1	2,917
● http://dbpedia.org/ontology/SpaceMission	742	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	6	3.778	2,841,333
● http://dbpedia.org/ontology/SportsTeam	208	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	4	2.667	2,765,5
● http://dbpedia.org/ontology/School	157	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	5	3.153	2,671
● http://dbpedia.org/ontology/Settlement	347	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	3	2.147	2,516
● http://dbpedia.org/ontology/Company	244	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	4	2.6	2,126
● http://dbpedia.org/ontology/Spacecraft	534	http://dbpedi.../http://dbpedia.org/	http://dbpedia...	5	3,233	2,103

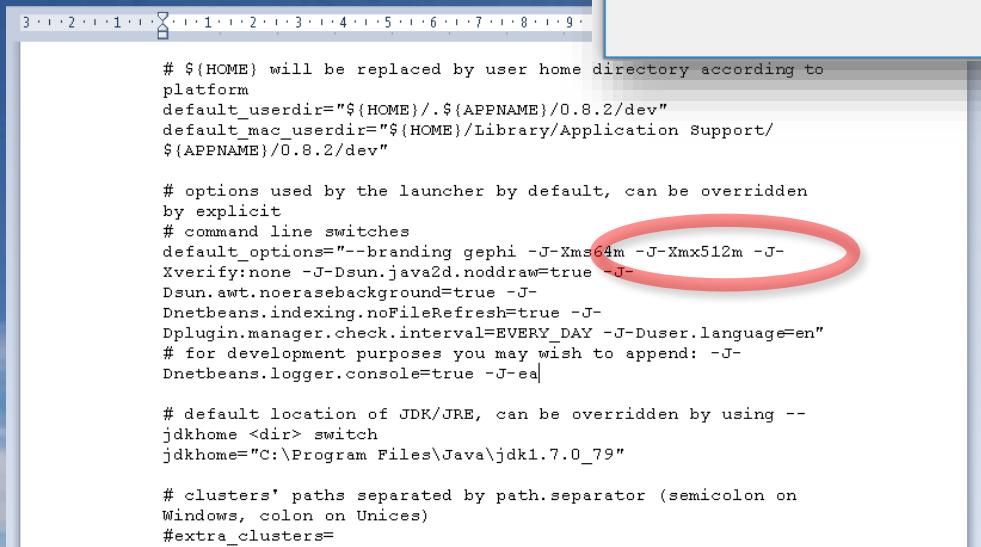
# Gephi

- The DBpedia Ontology use case
  - Statistics → Network Overview → Average Degree → Run
    - See the data table, with:  
In-Degree, Out-Degree, Degree
    - Degree runs parallel to  
PageRank
      - Notice Degree has more  
nodes with value != 0 ☺

Nodes	... Edges	... Configuration	Add node	Add edge	Search/Replace	Import Spreadsheet	Export table	More actions
● http://www.w3.org/1.127 http://http	0	0	0	0.017	2702	0	2702	
● http://www.w3.org/1.134 http://http	0	0	0	0.011	1716	0	1716	
● http://www.w3.org/1.148 http://http	0	0	0	0.006	1079	0	1079	
● http://www.w3.org/1.150 http://http	0	0	0	0.005	683	0	683	
● http://www.w3.org/1.131 http://http	0	0	0	0.005	777	0	777	
● http://dbpedia.org/147 http://http	3	1.911	57,285	0.003	499	19	518	
● http://dbpedia.org/221 http://http	2	1.6	10,427.667	0.002	261	10	271	
● http://www.w3.org/2.250 http://http	0	0	0	0.002	264	0	264	
● http://www.w3.org/1.188.5.107 144 http://http	1	1	38,877.333	0.002	224	17	241	
● http://www.ontology.178 http://http	0	0	0	0.001	226	0	226	
● http://www.w3.org/2.556 http://http	0	0	0	0.001	180	0	180	
● http://www.ontology.312 http://http	0	0	0	0.001	172	0	172	
● http://dbpedia.org/2.10 http://http	4	2.574	14,133	0.001	147	10	157	
● http://www.w3.org/2.325 http://http	0	0	0	0.001	149	0	149	
● http://www.w3.org/2.518 http://http	0	0	0	0.001	57	0	57	
● http://dbpedia.org/347 http://http	3	2.147	2,516	0.001	77	10	87	
● http://dbpedia.org/404 http://http	3	2.025	13,801.833	0.001	66	14	80	
● http://www.ontology.143 http://http	0	0	0	0.001	88	0	88	
● http://www.w3.org/1.421 http://http	0	0	0	0	67	0	67	
● http://www.w3.org/2.262 http://http	0	0	0	0	61	0	61	
● http://dbpedia.org/179 http://http	4	2.596	7,773	0	51	9	60	
● http://www.ontology.843 http://http	0	0	0	0	63	0	63	
● http://www.w3.org/2.228 http://http	0	0	0	0	50	0	50	
● http://dbpedia.org/1.747 http://http	6	3.778	2,841.333	0	46	10	56	

# Gephi limitations

- Out of memory
  - Edit {gephiDir}/etc/gephi.conf



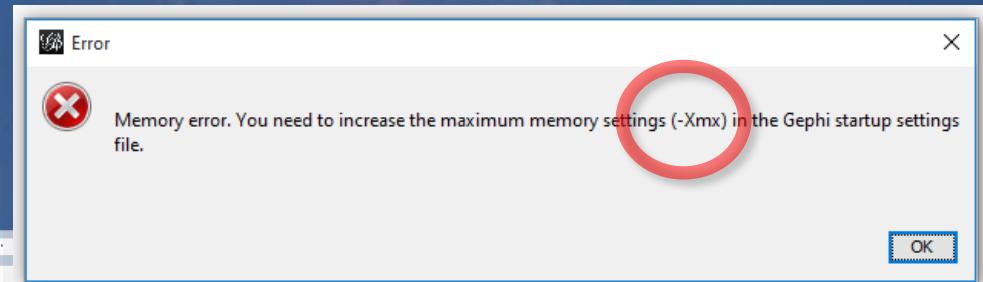
A screenshot of a code editor showing the contents of the Gephi configuration file, `gephi.conf`. The file contains various Java command-line options for the Gephi application. A red circle highlights the line `-J-Xmx512m`, which specifies the maximum memory allocation for the Java Virtual Machine.

```
# ${HOME} will be replaced by user home directory according to
platform
default_userdir="${HOME}/.${APPNAME}/0.8.2/dev"
default_mac_userdir="${HOME}/Library/Application Support/
${APPNAME}/0.8.2/dev"

# options used by the launcher by default, can be overridden
by explicit
# command line switches
default_options="--branding gephi -J-Xms64m -J-Xmx512m -J-
Xverify:none -J-Dsun.java2d.nodraw=true -J-
Dsun.awt.noerasebackground=true -J-
Dnetbeans.indexing.noFileRefresh=true -J-
Dplugin.manager.check.interval=EVERY_DAY -J-Duser.language=en"
# for development purposes you may wish to append: -J-
Dnetbeans.logger.console=true -J-ea|"

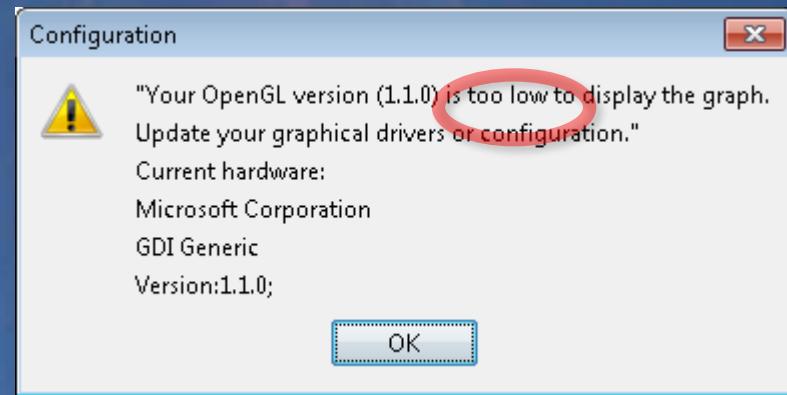
# default location of JDK/JRE, can be overridden by using --
jdkhome <dir> switch
jdkhome="C:\Program Files\Java\jdk1.7.0_79"

# clusters' paths separated by path.separator (semicolon on
Windows, colon on Unices)
#extra_clusters=
```



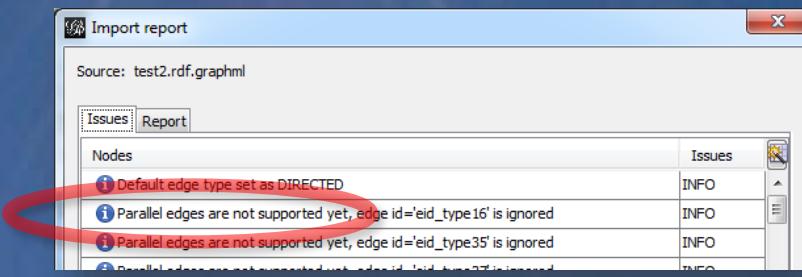
# Gephi limitations

- OpenGL
  - Update your graph drivers
    - For Intel go [here](#)



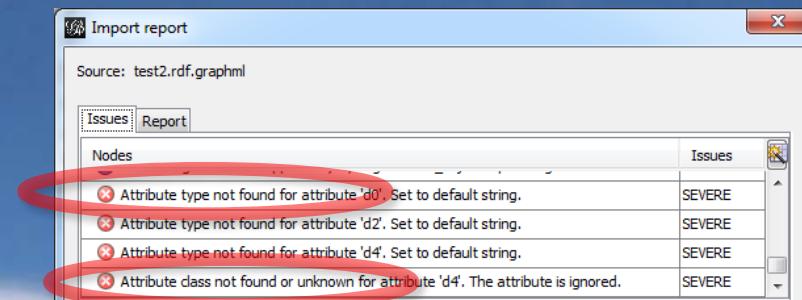
# Gephi limitations

- Logical
  - Not supported (yet)
    - Parallel edges  
(neither direct and reverse)



The screenshot shows the 'Import report' dialog box from Gephi. The source is 'test2.rdf.graphml'. The 'Issues' tab is selected. There are four entries in the table:

Nodes	Issues
Default edge type set as DIRECTED	INFO
Parallel edges are not supported yet, edge id='eid_type16' is ignored	INFO
Parallel edges are not supported yet, edge id='eid_type35' is ignored	INFO
Parallel edges are not supported yet, edge id='eid_type27' is ignored	INFO

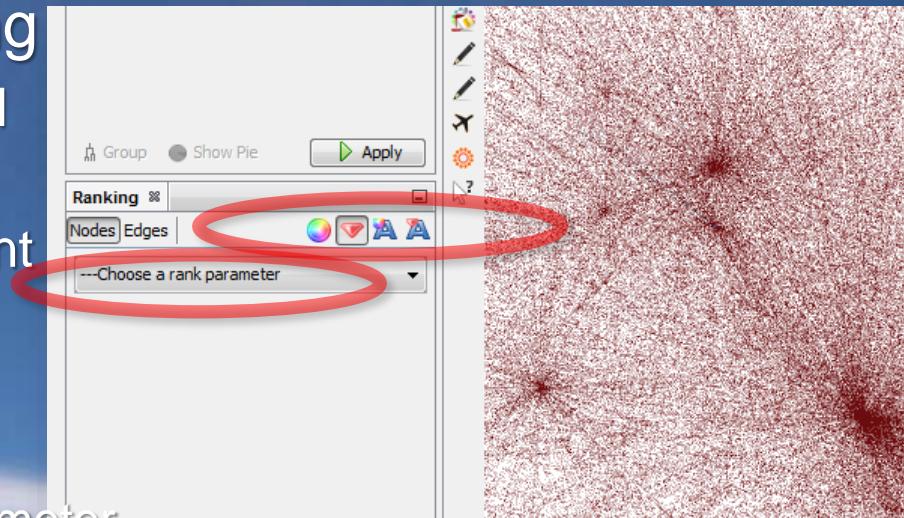


The screenshot shows the 'Import report' dialog box from Gephi. The source is 'test2.rdf.graphml'. The 'Issues' tab is selected. There are four entries in the table:

Nodes	Issues
Attribute type not found for attribute 'd0'. Set to default string.	SEVERE
Attribute type not found for attribute 'd2'. Set to default string.	SEVERE
Attribute type not found for attribute 'd4'. Set to default string.	SEVERE
Attribute class not found or unknown for attribute 'd4'. The attribute is ignored.	SEVERE

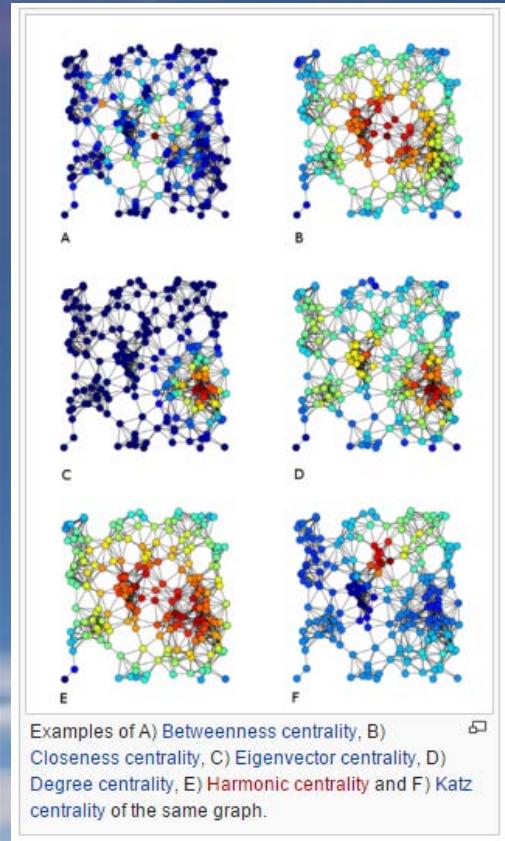
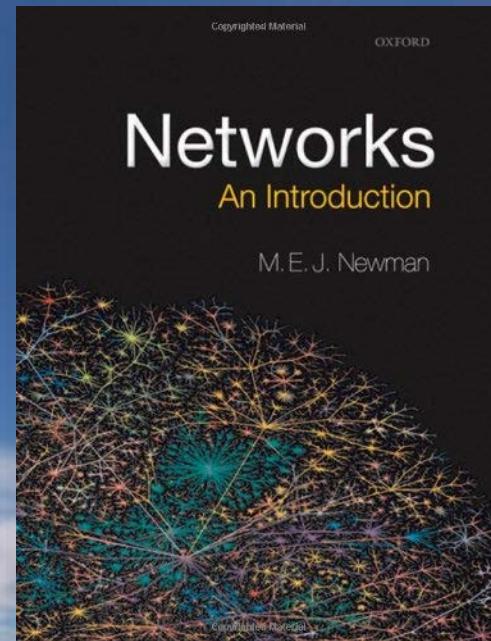
# Gephi limitations

- Usability
  - UI selectors are confusing
    - Always show the last used option
    - They do not use the current parameter ☹
    - E.g. load a graph
      - In the “Ranking” tab you cannot know which parameter has been used for node size, color, etc.



# More on graph theory

- Centrality
  - [See on wikipedia](#)
- Graphs
  - Newman's book, 2010.



# More on graph theory

- Classic papers on Wikipedia category graph analysis (and its applications)
  - [Identifying document topics using the Wikipedia category network](#) (2009)
  - [Decoding Wikipedia Categories for Knowledge Acquisition](#) (2008)
  - [Analysis of the Wikipedia Category Graph for NLP Applications](#) (2007)

# More on graph theory

- Classic papers on Wikipedia category graph analysis (and its applications)
  - Identifying document topics using the Wikipedia category network (2009)

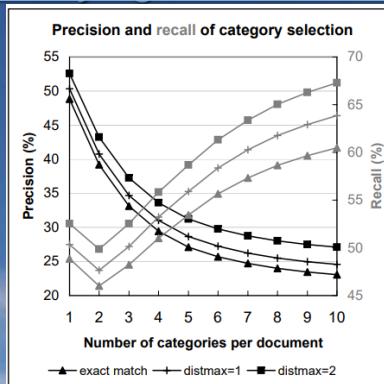


Figure 7. Black: percentage of Wikipedia categories which were correct in the top  $n$  categories. Gray: percentage of official Wikipedia categories among the top  $n$  categories. In both cases,  $n$  is shown on the  $x$  axis, values are averaged over processed documents.

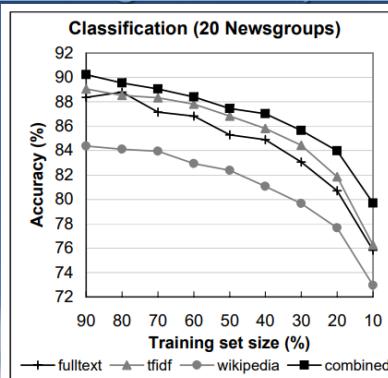
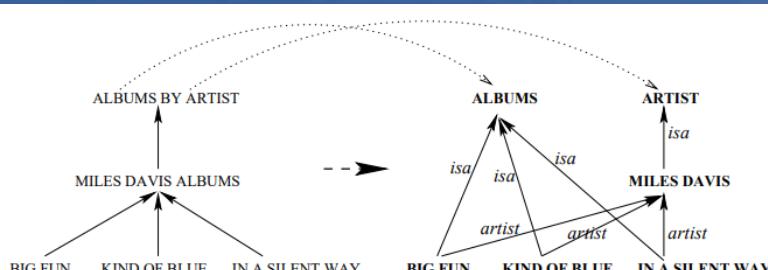


Figure 8. Classification accuracy in 20 Newsgroups at various training set sizes, when documents were represented by full text ("fulltext"), the 20 words with highest  $tf \times idf$  ("tfidf"), top 20 categories ("wikipedia"), or combination of the latter two ("combined").

# More on graph theory

- Classic papers on Wikipedia category graph analysis (and its applications)
  - [Identifying document topics using the Wikipedia category network](#) (2009)
  - [Decoding Wikipedia Categories for Knowledge Acquisition](#) (2008)



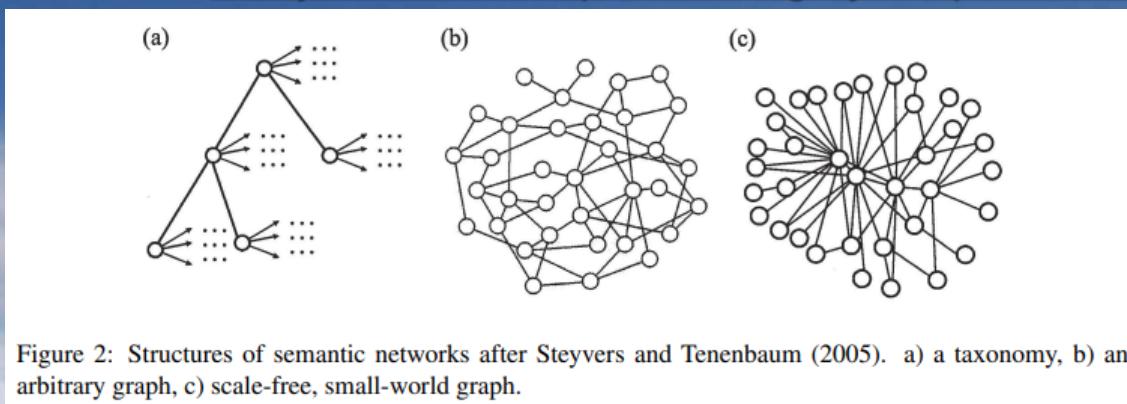
Category type	# categories	# relations extracted	Evaluation		
			P	manual $\cap$	manual $\cup$
explicit relations	3,450	86,649			
<code>caused_by, based_in, written_by, ...</code>	2,152	43,938	-	94.37%	96.38%
<code>member_of</code>	1,298	42,711	24% (25)	95.56%	97.17%
partly explicit and implicit relation categories	98,855	9,751,748			
<code>isa</code>	3,400,243	44.57% (6,250)	76.4%	84%	
<code>spatial</code>	3,201,125	39.69% (1,325)	87.09%	97.98%	

Table 2: Extracted relations and evaluation results

Figure 3: Relations inferred from "by" categories

# More on graph theory

- Classic papers on Wikipedia category graph analysis (and its applications)
  - [Identifying document topics using the Wikipedia category network](#) (2009)
  - [Decoding Wikipedia Categories for Knowledge Acquisition](#) (2008)
  - [Analysis of the Wikipedia Category Graph for NLP Applications](#) (2007)



# More on graph theory

- Less classic papers on Wikipedia category graph analysis (and its applications)
  - Uncovering the Semantics of Wikipedia Categories (2019)

**Table 2.** Total number of axioms/assertions and precision scores, based on the crowd-sourced evaluation. Numbers in parentheses denote the *total* number of assertions generated (including those already existing in DBpedia), as well as the precision estimation of those total numbers. The latter were derived as a weighted average from the human annotations and the overall correctness of existing assertions in DBpedia according to [3].

Approach	Count		Precision [%]	
	Relation axioms		Type axioms	
Cat2Ax	272,707	95.6	430,405	96.8
C-DF	143,850	83.6	28,247	92.0
Catriple	306,177	87.2	–	–
Relation assertions			Type assertions	
Cat2Ax	4,424,785 (7,554,980)	87.2 (92.1)	3,342,057 (12,111,194)	90.8 (95.7)
C-DF	766,921 (2,856,592)	78.4 (93.4)	198,485 (2,352,474)	76.8 (97.1)
Catriple	6,260,972 (6,836,924)	74.4 (76.5)	–	–

# More on graph theory

- Less classic papers on Wikipedia category graph analysis (and its applications)
  - [Cat2Type: Wikipedia Category Embeddings for Entity Typing in Knowledge Graphs](#) (2021)

Table 2: Results on DBpedia splits and FIGER

Models	DB1		DB2		DB3		FIGER	
	$Ma - F_1$	$Mi - F_1$	$Ma - F_1$	$Mi - F_1$	$Ma - F_1$	$Mi - F_1$	$Ma - F_1$	$Mi - F_1$
CUTE [32]	0.679	0.702	0.681	0.713	0.685	0.717	0.743	0.782
MuLR [34]	0.748	0.771	0.757	0.784	0.752	0.775	0.776	0.812
FIGMENT [33]	0.740	0.766	0.738	0.765	0.745	0.769	0.785	0.819
APE [16]	0.758	0.784	0.761	0.785	0.760	0.782	0.722	0.756
HMGCN [17]	0.785	0.812	0.794	0.820	0.791	0.817	0.789	0.827
Cat2Type-node2vec	<b>0.950</b>	<b>0.948</b>	<b>0.948</b>	<b>0.946</b>	<b>0.948</b>	<b>0.946</b>	0.683	<b>0.84</b>
Cat2Type-word2vec	<b>0.876</b>	<b>0.876</b>	<b>0.723</b>	<b>0.738</b>	<b>0.723</b>	<b>0.742</b>	0.502	<b>0.726</b>
Cat2Type-GloVe	<b>0.883</b>	<b>0.884</b>	<b>0.728</b>	<b>0.742</b>	<b>0.731</b>	<b>0.746</b>	0.501	<b>0.726</b>
Cat2Type-Wikipedia2Vec	<b>0.897</b>	<b>0.897</b>	<b>0.733</b>	<b>0.749</b>	<b>0.739</b>	<b>0.754</b>	0.522	<b>0.737</b>
Cat2Type-BERT	<b>0.983</b>	<b>0.984</b>	<b>0.983</b>	<b>0.983</b>	<b>0.985</b>	<b>0.985</b>	0.764	<b>0.881</b>

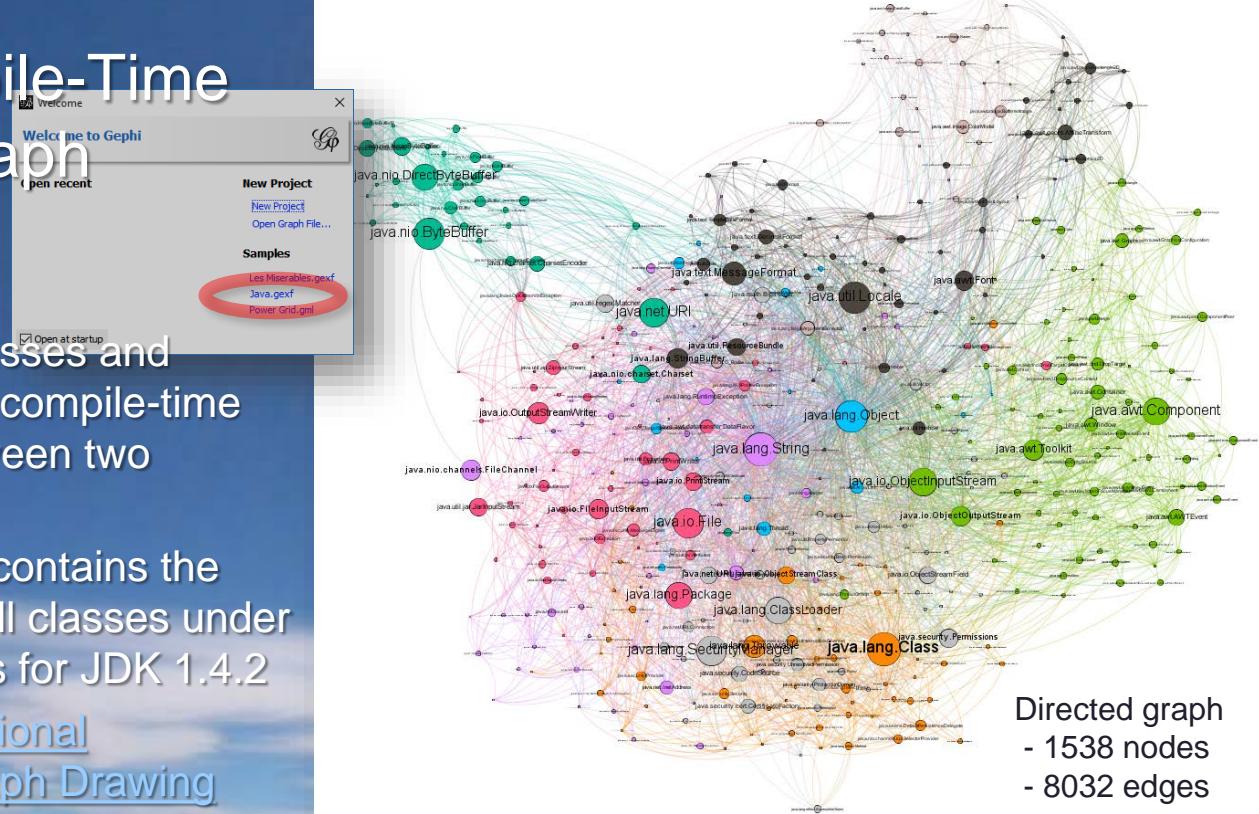
# More on Gephi Layouts

- Online tutorial
  - Gephi web site
- Book
  - Ken Cherven, *Mastering Gephi network visualization.* Packt Publishing, 2015.

Selecting the Layout				
Algorithm name	Type	Strengths	Weaknesses	When to use
DAG layout	Tree	Ordering hierarchical data	This is impractical for very large networks	This can be used in cases where you wish to see levels of data in a top to bottom order
Dual Circle layout	Circular	This has the ability to focus on a group of nodes within the larger network	This layout results in very large networks that might create viewing issues	This layout can be used in instances where a second circle is desirable to focus on a limited group of nodes
Force Atlas	Force-directed	This includes many options and has a high level of accuracy	This can be very slow and is not suited to large networks	This layout is useful for network analysis and discovery, and for measuring network behavior
Force Atlas 2	Force-directed	This is faster than original Force Atlas and handles very large networks	This suffers slightly on overall accuracy	This is used as a good tool for network analysis and discovery, and for detecting behavioral patterns

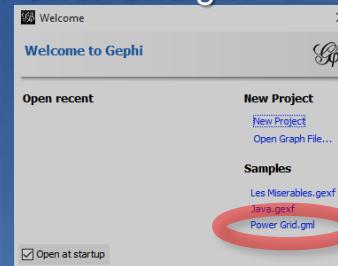
# More examples

- The Java Compile-Time Dependency graph
  - Gephi dataset
    - Java.gexf
  - Nodes are Java classes and directed edges are compile-time dependencies between two classes.
  - The data provided contains the dependencies for all classes under the `java.*` packages for JDK 1.4.2
  - From [2006 International Symposium on Graph Drawing](#)

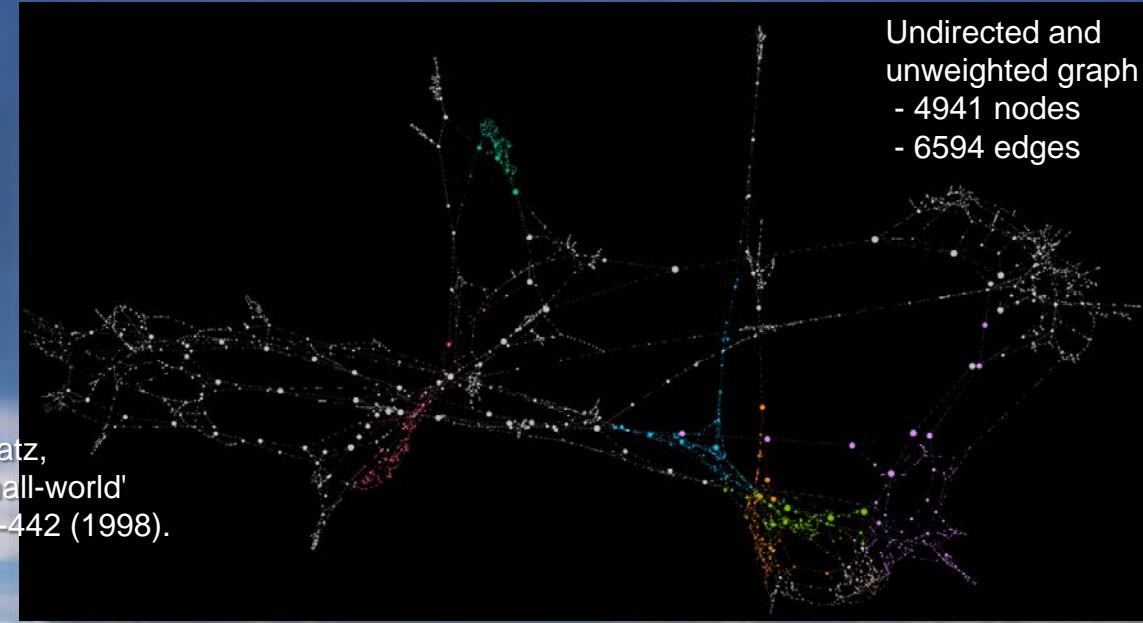


# More examples

- Topology of the Western States power grid in U.S.
  - Gephi dataset
    - Power Grid.gml



- Original paper citation
  - D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks", Nature 393, 440-442 (1998).



# More SPARQL examples

- Panama papers

- EP

- <http://data.ontotext.com/sparql>

- in RDF

- <ftp://ftp.ontotext.com/pub/leaks/rdf/rdf.zip>

The screenshot shows the homepage of the Linked Leaks service, specifically for the Panama Papers dataset. The top navigation bar includes links for 'Data', 'SPARQL', and 'Learn More'. A search bar is present, along with a button labeled 'SE' and 'leaks2'. The main header reads 'Linked Leaks' and 'Panama Papers and more as Linked Data'. Below the header, there are three main sections: 'The Panama Papers Database in Context', 'Investigative Reporting Workbench', and 'Linked Leaks for Download'. The 'Database in Context' section contains text about the unique perspective provided by linking the Panama Papers to other datasets like DBpedia and GeoNames. It also lists sample queries Q1 and Q2. The 'Workbench' section provides a SPARQL endpoint and encourages users to explore the data. The 'Download' section offers a zip file for download and a link to GraphDB Free.

Linked Leaks

Panama Papers and more as Linked Data

The Panama Papers Database in Context

Investigative Reporting Workbench

Linked Leaks for Download

See data model

Sample queries:

- Q1: Countries by number of entities related to them
- Q2: Country pairs by ownership statistics

Download Data

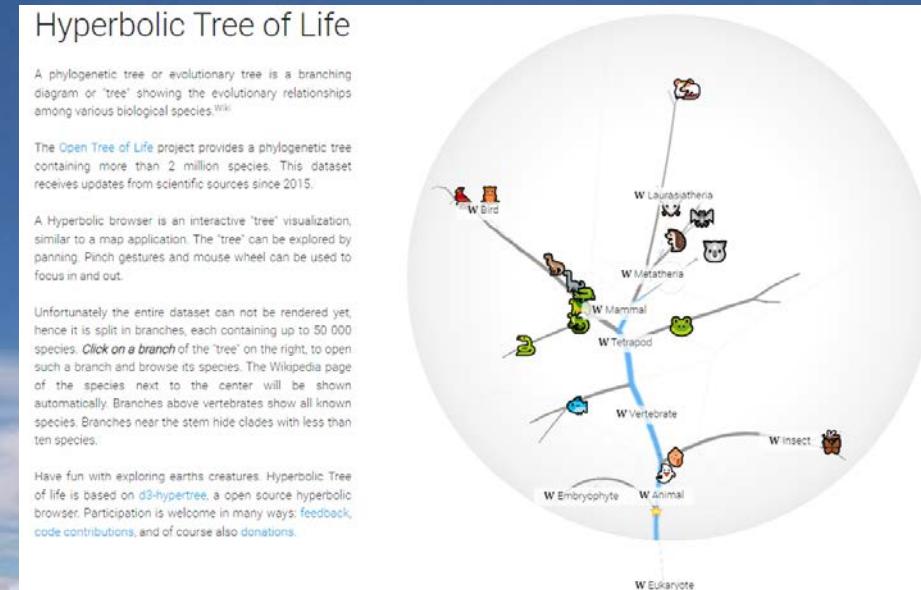
# More visualizations

- Visualization of trees
- [Treevis.net](#). 339 visualization methods as of April 2024

The screenshot shows the homepage of treevis.net. At the top, there are links for "How to cite this site?", "Check out other surveys!", and social media icons for Facebook, LinkedIn, and Twitter. The title is "treevis.net - A Visual Bibliography of Tree Visualization 2.0 by Hans-Jörg Schulz". Below the title, there is a date "v.04-SEP-2023". The main navigation bar includes filters for "Dimensionality" (All, 2D, 3D), "Representation" (All, hierarchical, radial, network), "Alignment" (All, circular, radial, hierarchical), "Fulltext Search" (with a search bar and clear button), and "Techniques Shown" (set to 339). Below the navigation is a grid of 339 small thumbnail images, each representing a different tree visualization method. Some thumbnails include labels such as "Acacia Tree (2020)". At the bottom of the page, there is a footer with a URL "https://treevis.net/#Burch2020\_2" and a "SEARCH" button.

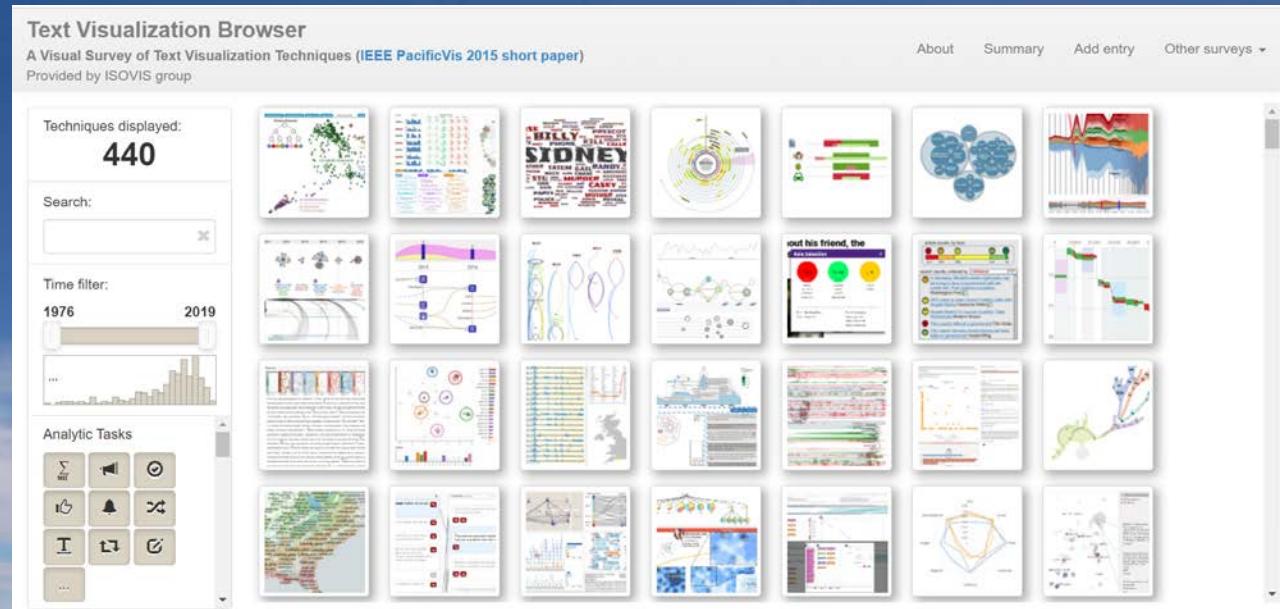
# More visualizations

- Visualization of trees
- Hyperbolic trees
  - [Michael Glatzhofer \(<https://glouwa.github.io/d3-hypertree-examples/hyperbolictree-slides>\)](https://glouwa.github.io/d3-hypertree-examples/hyperbolictree-slides)

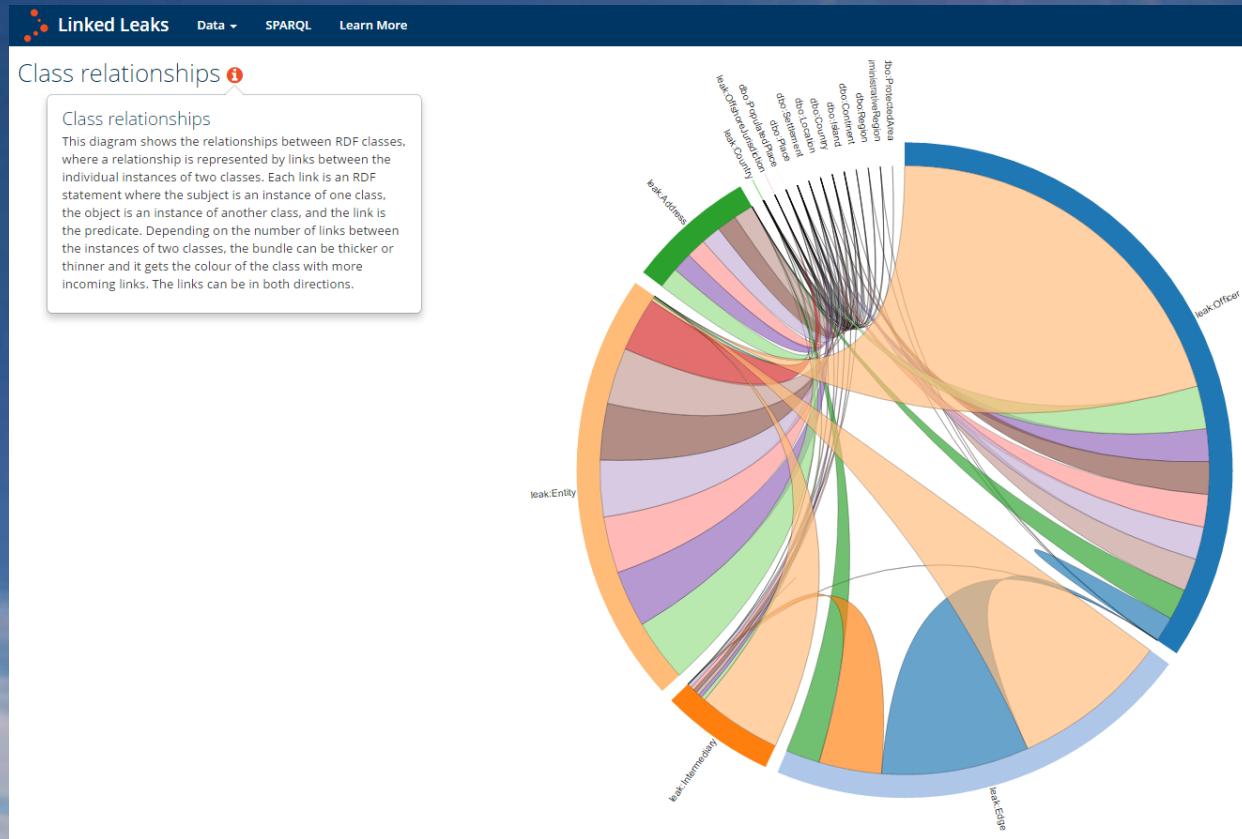


# More visualizations

- Visualization of texts
- [textvis](#). 440 visualization methods as of April 2024

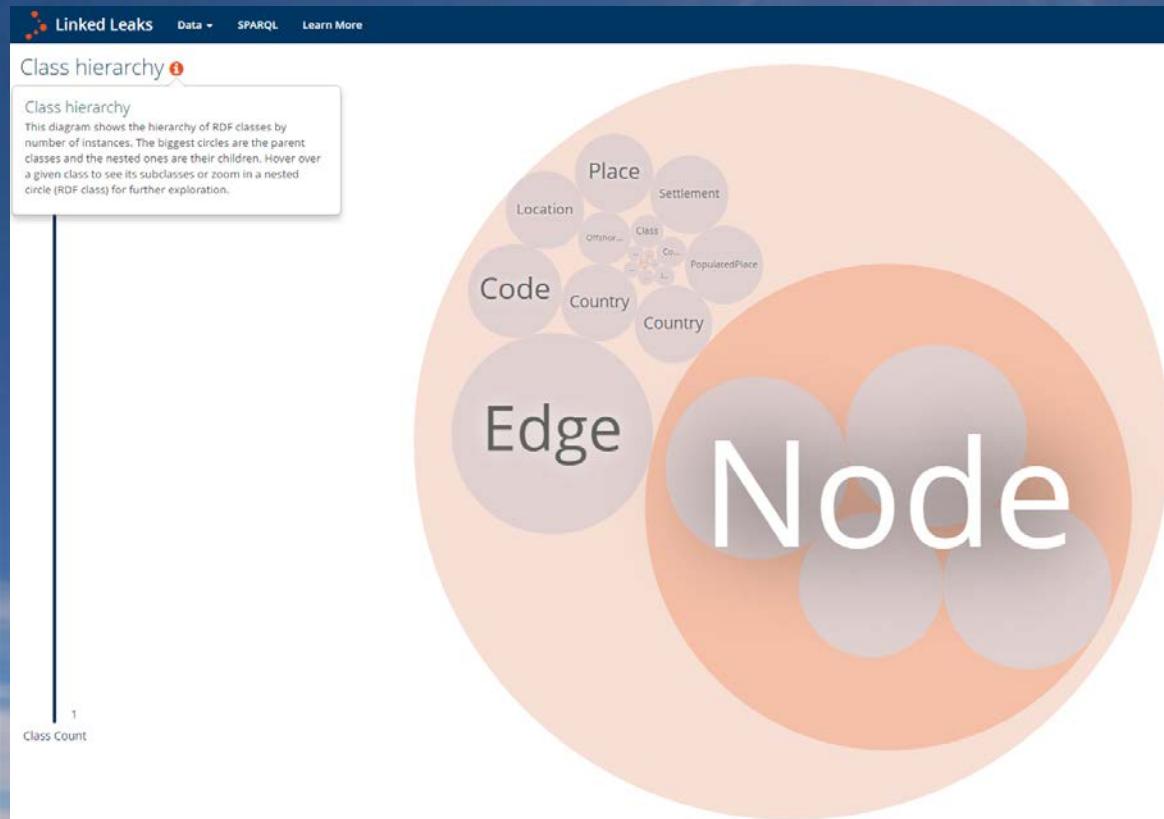


# More visualizations



See <http://data.ontotext.com/sparql>

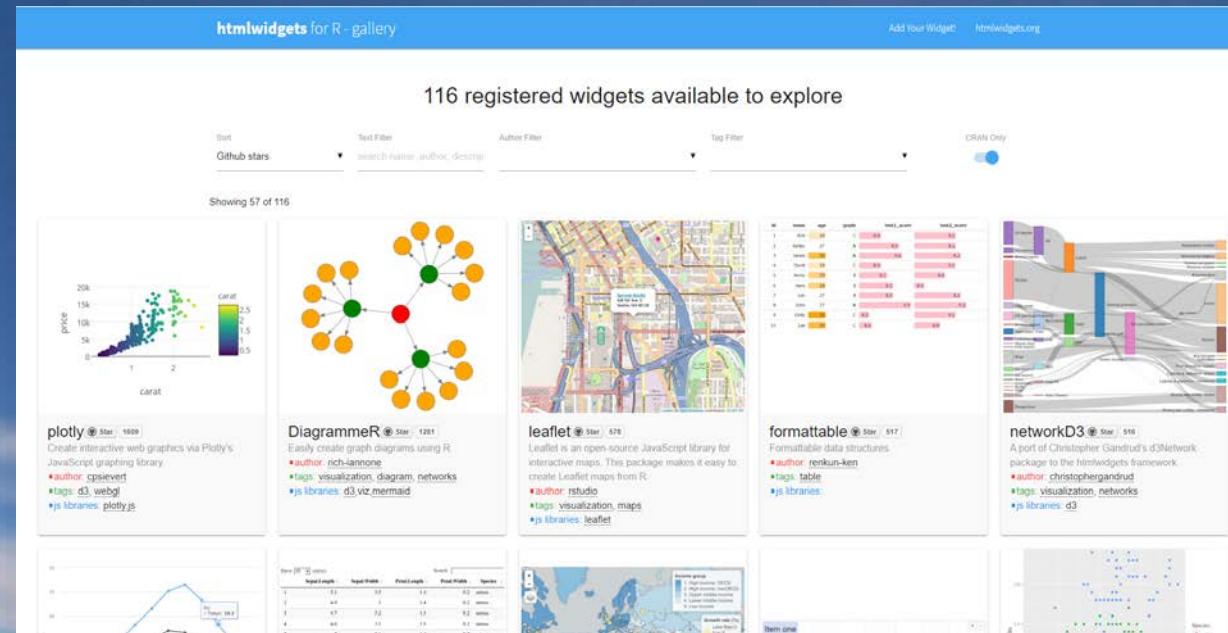
# More visualizations



See <http://data.ontotext.com/sparql>

# More visualizations

- The coolest for R: package `htmlwidgets`



# Displaying on a web page

- Sigma  
[sigmajs.org](http://sigmajs.org)
  - Javascript
  - Reads .gexf files ☺
  - R package
    - [github.com/jjallaire/sigma](https://github.com/jjallaire/sigma)



The screenshot shows the homepage of the sigma.js website. At the top, there's a navigation bar with links for 'GET STARTED', 'FEATURES', 'USE CASES', 'TUTORIAL', and 'REFERENCES'. Below the navigation is a large, complex network graph visualization composed of red nodes and orange edges. The word 'sigmajs' is written vertically in red text next to the graph. At the bottom of the main section, there are two buttons: a blue 'TUTORIAL' button with a gear icon and a red 'DOWNLOAD' button with a downward arrow icon. The version 'v1.1.0' is also visible. A descriptive paragraph at the bottom states: 'Sigma is a JavaScript library **dedicated to graph drawing**. It makes easy to publish networks on Web pages, and allows developers to integrate network exploration in rich Web applications.'

# Displaying on a web page

- Gephi exporter for Sigma
  - Generates all the html/js/json code in a given folder
    - Requires a html server (like Tomcat)
      - Does not work as a html file on the browser
    - Limitation: No labels ☹
    - Installation
      - Gephi 0.8.2 vía Marketplace
      - Gephi 0.9.1 vía Tools→Plugins→Available plugins

Sigmajs Exporter

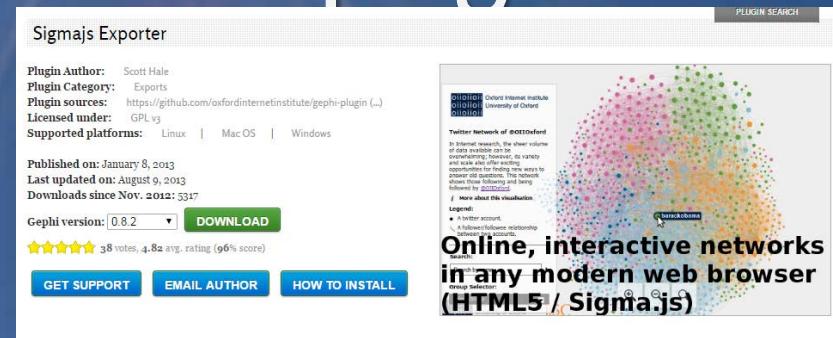
Plugin Author: Scott Hale  
Plugin Category: Exports  
Plugin sources: <https://github.com/oxfordinternetinstitute/gephi-plugin> (..)  
Licensed under: GPL v3  
Supported platforms: Linux | Mac OS | Windows

Published on: January 8, 2013  
Last updated on: August 9, 2013  
Downloads since Nov. 2012: 5347

Gephi version: 0.8.2 [DOWNLOAD](#)

★★★★★ 38 votes, 4.82 avg. rating (96% score)

[GET SUPPORT](#) [EMAIL AUTHOR](#) [HOW TO INSTALL](#)



Twitter Network of @OIIxford  
In Internet research, the sheer volume of data available can be overwhelming. This plugin allows us to easily generate network visualizations from large datasets, providing opportunities for finding new ways to analyse them. It also allows us to show those findings and bring them to life.

Legend:  
• Follower/Followee relationship  
• Retweet relationship

Online, interactive networks in any modern web browser (HTML5 / Sigma.js)



# Gephi datasets

- Gephi Wiki “Datasets” page (in Github)
  - <https://github.com/gephi/gephi/wiki/Datasets>
- Categories:
  - Web and Internet
  - Social networks
  - Biological networks
  - Infrastructure networks
  - Other networks

The screenshot shows a GitHub repository page for "gephi/gephi". The repository has 445 issues, 10 pull requests, and 10 discussions. The "Datasets" section contains links to various sample datasets in different formats (GEXF, GDF, GMML, NET, GraphML, DL, DOT). It also includes sections for "Web and Internet" and "Social networks", each listing specific datasets like "EuroSiS web mapping study" and "Les Miserables". A sidebar on the right provides links to "Software", "Manuals", "Community", and "Plugins".

Search or jump to... Pull requests Issues Marketplace Explore

gephi/gephi Public

Code Issues 445 Pull requests 10 Discussions Actions Projects Wiki Security Insights

Datasets

Marcus Bingerheimer edited this page on 23 Jul 2021 · 10 revisions

The Gephi sample datasets below are available in various formats (GEXF, GDF, GMML, NET, GraphML, DL, DOT). Feel free to add new datasets, but be sure to cite the original authors.

Supported graph formats are described [here](#).

Gephi can open zipped files directly.

Web and Internet

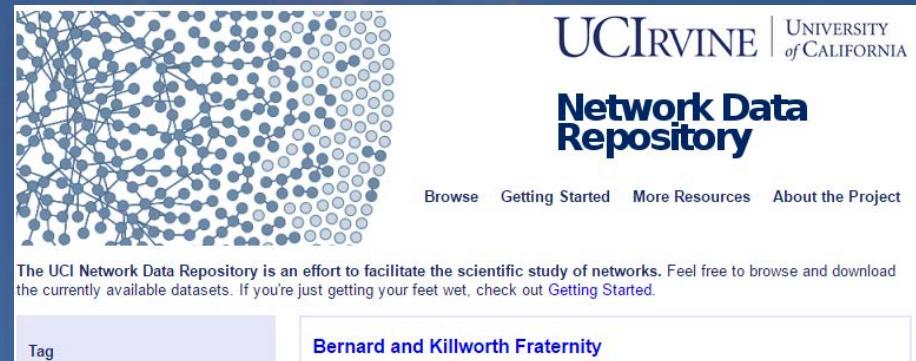
- [GEXF file. EuroSiS web mapping study](#): Mapping interactions between Science in Society actors on the Web of 12 European countries. Original report and data can be found [here](#).
- [GML file. Internet](#): a symmetrized snapshot of the structure of the Internet at the level of autonomous systems, reconstructed from BGP tables posted by the University of Oregon Route Views Project. This snapshot was created by Mark Newman on July 22, 2006 and was not previously published.

Social networks

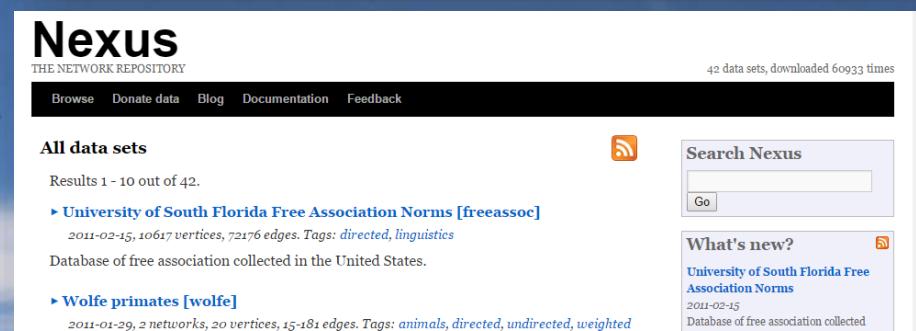
- [GML file. Les Misérables](#): coappearance weighted network of characters in the novel Les Misérables. D. E. Knuth. The Stanford GraphBase: A Platform for Combinatorial Computing. Addison-Wesley, Reading, MA (1995).
- [GEXF file. Hypertext 2009 dynamic contact network](#): contact network during the Hypertext 2009 conference. Source: [Sociopatterns.org](#).
- [GEXF file. CLASS of 1880/81](#): friendship network of a German boys' school class from 1880/1881. It's based on the probably first ever primarily collected social network dataset, assembled by the primary school teacher Johannes Dötsch. The data was reanalyzed and compiled for the article: Heidler, R., Gamper, M., Herz, A., Eßer, F. (2014): Relationship patterns in the 19th century: The friendship network in a German boys' school class from 1880 to 1881 revisited. *Social Networks* 13: 1–13.
- [GML file. Zachary's karate club](#): social network of friendships between 34 members of a karate club at a US university in the 1970s. W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* 33, 452–473 (1977).
- [GML file. Coauthorships in network science](#): coauthorship network of scientists working on network theory and experiment, as compiled by M. Newman in May 2006. A figure depicting the largest component of this network can be found [here](#). M. E. J. Newman, Phys. Rev. E 74, 036104 (2006).
- [GEXF file. CPAN authors](#): CPAN Explorer is a visualization project aiming at analyzing the relationships between the developers and the packagers of the Perl modules, known as the CPAN community. This network uses

# More datasets

- UCI Network data repository
  - <https://networkdata.ics.uci.edu/>
- Nexus (igraph.org)
  - [http://nexus.igraph.org/api/dataset\\_info](http://nexus.igraph.org/api/dataset_info)



The screenshot shows the homepage of the UCI Network Data Repository. At the top right is the UCI IRVINE logo with "UNIVERSITY OF CALIFORNIA". Below it is the "Network Data Repository" logo. A navigation bar at the bottom has links for "Browse", "Getting Started", "More Resources", and "About the Project". The main content area features a large, dense network graph visualization composed of blue and grey nodes connected by lines. Below the graph is a text block: "The UCI Network Data Repository is an effort to facilitate the scientific study of networks. Feel free to browse and download the currently available datasets. If you're just getting your feet wet, check out Getting Started." There are two search/filter boxes: one labeled "Tag" and another labeled "Bernard and Killworth Fraternity".



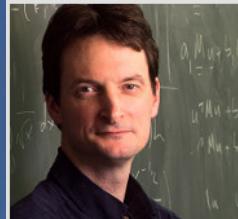
The screenshot shows the homepage of the Nexus (igraph.org) website. At the top left is the "Nexus" logo with "THE NETWORK REPOSITORY" underneath. To the right, a message says "42 data sets, downloaded 60933 times". A black navigation bar below the header includes links for "Browse", "Donate data", "Blog", "Documentation", and "Feedback". On the left, a sidebar lists "All data sets" with a count of "Results 1 - 10 out of 42". The main content area displays two dataset entries:

- University of South Florida Free Association Norms [freeassoc]  
2011-02-15, 10617 vertices, 72176 edges. Tags: directed, linguistics  
Database of free association collected in the United States.
- Wolfe primates [wolfe]  
2011-01-29, 2 networks, 20 vertices, 15-181 edges. Tags: animals, directed, undirected, weighted  
Database of free association collected

On the right side, there is a "Search Nexus" input field with a "Go" button and a "What's new?" section listing the same two datasets with their respective dates.

# Even more datasets

- Mark Newman datasets
  - <http://www-personal.umich.edu/~mejn/netdata/>



**Mark Newman**

Anatol Rapoport Distinguished University Professor of Physics  
Department of Physics and Center for the Study of Complex Systems  
University of Michigan

External Faculty  
Santa Fe Institute

## Network data

This page contains links to some network data sets I've compiled over the years. All of these are free already made the data freely available, or that I have consulted the authors and received permission to please cite the original sources.

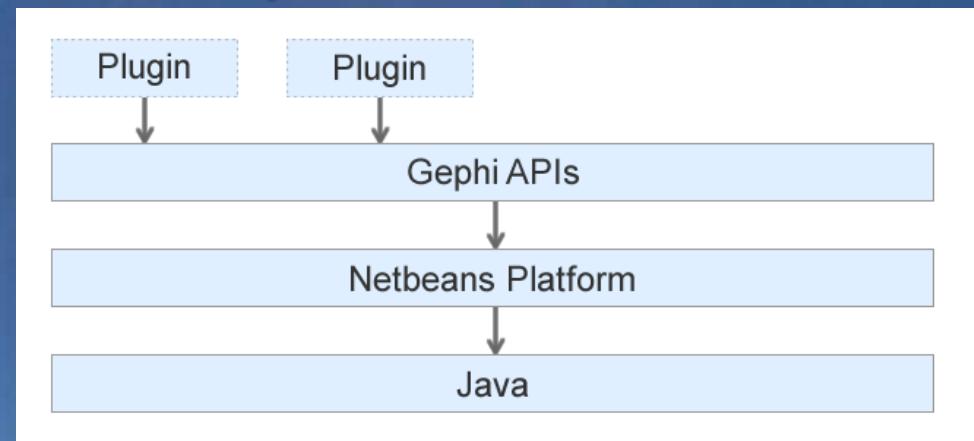
The data sets are in GML format. For a description of GML see [here](#). GML can be read by many net that will read the files into a data structure. It's available [here](#). There are many features of GML not a Python parser for GML available as part of the NetworkX package [here](#) and another in the [igraph pa](#) software (Java, C++, Perl, R, Matlab, etc.) that reads GML, let me know.

### Data sets

- [Zachary's karate club](#): social network of friendships between 34 members of a karate club at a conflict and fission in small groups, *Journal of Anthropological Research* **33**, 452-473 (1977).
- [Les Misérables](#): coappearance network of characters in the novel *Les Misérables*. Please cite L Addison-Wesley, Reading, MA (1993).
- [Word adjacencies](#): adjacency network of common adjectives and nouns in the novel *David Co* (2006).
- [American College football](#): network of American football games between Division IA college *Natl. Acad. Sci. USA* **99**, 7821-7826 (2002).
- [Dolphin social network](#): an undirected social network of frequent associations between 62 dol K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, *Behavioral Ecology an* [these data on this web site](#)

# Gephi API

- Built on top of NetBeans platform
  - Thread-safe
  - Multi-view,  
host sub graphs



- More at [Gephi.org](http://Gephi.org)  
(<https://github.com/gephi/gephi/wiki/How-to-build-Gephi>)



# Gephi & R

- From R:
  - Create a graph with the [igraph](#) package
    - This package creates igraph objects
  - Write the igraph object to an graphML file
  - Private (non CRAN) package [ggraph](#) (grammar's graph, like ggplot2 vs plot).
    - Benefits: igraph+bigrams+ggforce+dendrogram+Hierarchical Edge Bundles + Treemaps + animations...
    - Can convert an igraph to a ggraph
- From Gephi:
  - Read the graphML file
- An example [here](#)

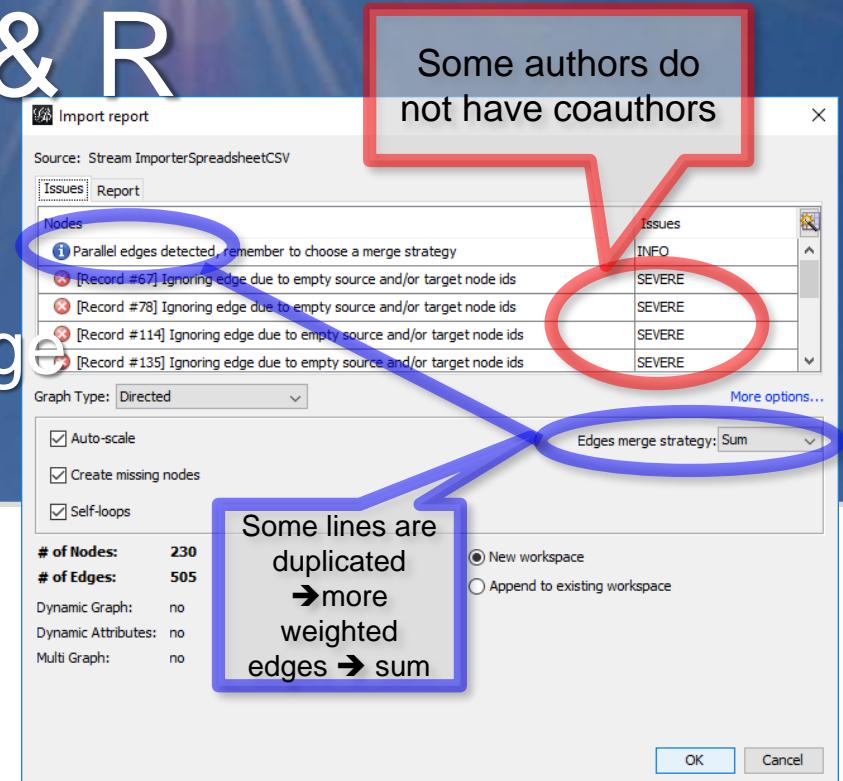
# Gephi & R

- From R:
  - With the scholar package
    - Export the edges list

```
library(scholar)

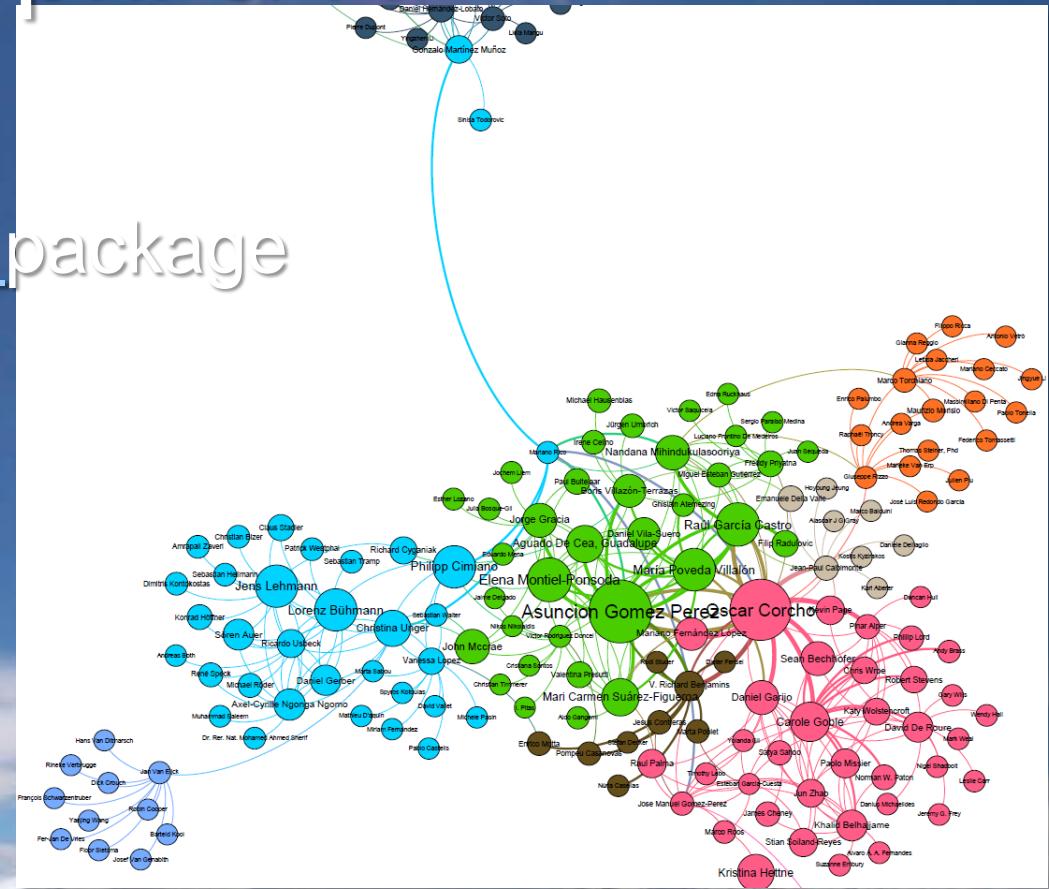
#Network of coauthorship of Mariano Rico
co <- get_coauthors("h15WPQEAAAAJ&hl", 10, 2)

write.table(co, file="coauthorsnet.ok.csv",
           quote = F, row.names = F,
           col.names = c("Source", "Target"), #To be readable by Gephi as 'edges table'
           fileEncoding="UTF-8", sep = ";" #Authors can have commas
)
```



# Gephi & R

- From R:
  - With the scholar package
    - Result



# For Python fans

- **Igraph**
  - Also available for Python fans
- [graph-tool](#)
  - ToDo: make a comparison to Igraph

The screenshot shows the official website for graph-tool. At the top, there's a navigation bar with links for "Download", "Documentation", "Mailing List", "Git", and "Issues". Below the header, a large heading says "What is graph-tool?". A brief description follows: "Graph-tool is an efficient Python module for manipulation and statistical analysis of graphs (a.k.a. networks). Contrary to most other python modules with similar functionality, the core data structures and algorithms are implemented in C++, making extensive use of template metaprogramming, based heavily on the Boost Graph Library. This confers it a level of performance that is comparable (both in memory usage and computation time) to that of a pure C/C++ library." There are three main callout boxes at the bottom: "It is Fast!" (with a speedometer icon), "Extensive Features" (with a gear icon), and "Powerful Visualization" (with an eye icon). Each box has a short explanatory sentence.

graph-tool | Efficient network analysis

Download Documentation Mailing List Git Issues

What is graph-tool?

Graph-tool is an efficient Python module for manipulation and statistical analysis of graphs (a.k.a. networks). Contrary to most other python modules with similar functionality, the core data structures and algorithms are implemented in C++, making extensive use of template metaprogramming, based heavily on the Boost Graph Library. This confers it a level of performance that is comparable (both in memory usage and computation time) to that of a pure C/C++ library.

Download version 2.26 ↗

See Instructions | See Changelog

It is Fast!

Extensive Features

Powerful Visualization

Despite its nice, soft outer appearance of a regular python module, the core

An extensive array of features is included, such as support for arbitrary vertex, edge or graph

Conveniently draw your graphs, using a variety of algorithms and output formats (including to



Graphs from R

**IGRAPH**

# igraph

- Table of Contents
  - Basics
    - Graph structure
    - Measures
      - Diameter
      - Degree
      - Edge connectivity
      - Edge betweenness
  - Selecting
  - Automatic graphs
  - Layouts para grafos grandes
    - [DRL](#) (Distributed Recursive Layout)
    - [LGL](#) (Large Graph Layout)
    - [Reingold-Tilford tree layout](#) (useful for (almost) tree-like graphs)

# igraph

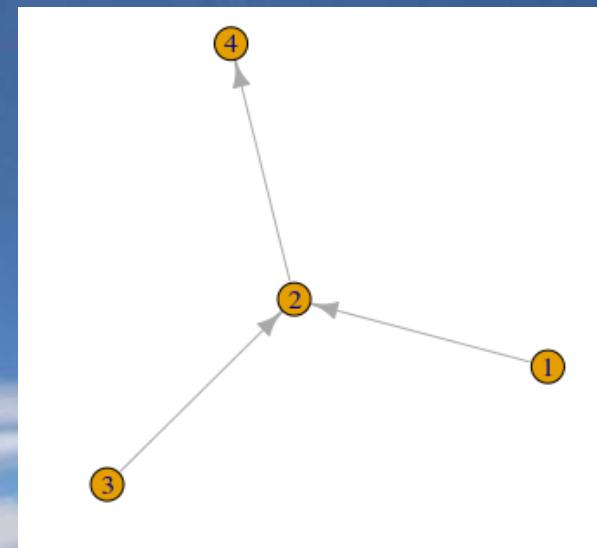
- Well known software
  - Versions for Python, C and R.
- No good manual ☹
  - But it has a [Spanish online tutorial for R](#)

# Igraph basics

- A graph is
  - Set of nodes (vertices, singular vertex)
  - Set of arcs (edges).

```
library(igraph)

#Graphs with the given list of edges
edges <- c(1,2, 3,2, 2,4)
g<-graph(edges, n=max(edges), directed=TRUE)
plot(g)
```



# Igraph basics

- Graph structure

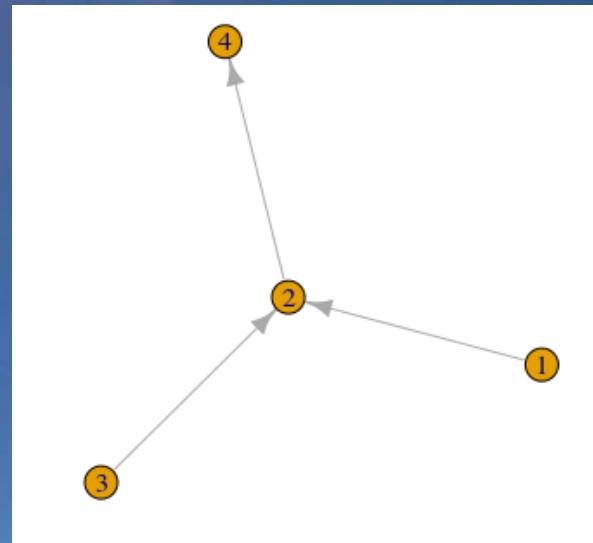
```
library(igraph)

#Graphs with the given list of edges
edges <- c(1,2, 3,2, 2,4)
g<-graph(edges, n=max(edges), directed=TRUE)

#Number of vertices (nodes)
vcount(g) #Returns 4

#Number of edges
ecount(g) #Returns 3

#The vertices
V(g) #Returns an object igraph.vs ("vertex sequence")
> [1] 1 2 3 4
#The edges
E(g) #Returns an object igraph.es ("edge sequence")
>[1] 1->2 3->2 2->4
```



# Igraph basics

- Graph structure

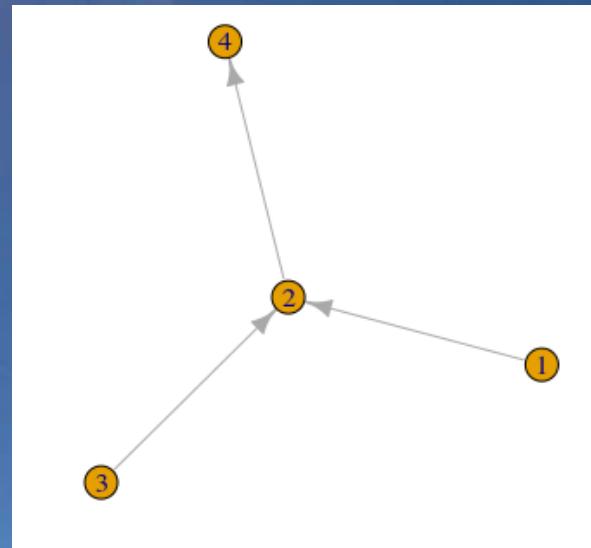
```
library(igraph)

#Graphs with the given list of edges
edges <- c(1,2, 3,2, 2,4)
g<-graph(edges, n=max(edges), directed=TRUE)

#Neighbors
neighbors(g,
           V(g)[1],      # 1 2 3 4
           mode = "all") # "in" and "out"
>[1] 2

neighbors(g, V(g)[2], mode = "all") #All nodes from/to node 2
>[1] 1 3 4

#Connection
are.connected(g, V(g)[1], V(g)[3])
> [1] FALSE
```



# Igraph basics

- Graph attributes
  - For vertices and edges

```
g <- graph.full(5) #Fully connected
```

```
set.seed(666) #To get always the same graph  
E(g)$weight <- runif(ecount(g)) #rand uniform between 0 and 1
```

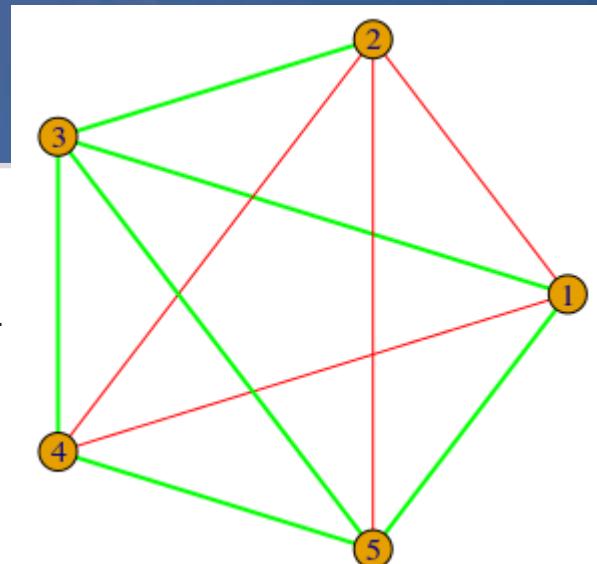
```
E(g)$width <- 1
```

```
E(g)$color <- "red"
```

```
E(g)[ weight < 0.5 ]$width <- 2
```

```
E(g)[ weight < 0.5 ]$color <- "green"
```

```
plot(g,  
      layout=layout.circle,  
      edge.width=E(g)$width,  
      edge.color= E(g)$color)
```



# Igraph basics

## • Plot params

NODES	
vertex.color	Node color
vertex.frame.color	Node border color
vertex.shape	One of "none", "circle", "square", "csquare", "rectangle", "crectangle", "vrectangle", "pie", "raster", or "sphere"
vertex.size	Size of the node (default is 15)
vertex.size2	The second size of the node (e.g. for a rectangle)
vertex.label	Character vector used to label the nodes
vertex.label.family	Font family of the label (e.g."Times", "Helvetica")
vertex.label.font	Font: 1 plain, 2 bold, 3, italic, 4 bold italic, 5 symbol
vertex.label.cex	Font size (multiplication factor, device-dependent)
vertex.label.dist	Distance between the label and the vertex
vertex.label.degree	The position of the label in relation to the vertex, where 0 right, "pi" is left, "pi/2" is below, and "-pi/2" is above
EDGES	
edge.color	Edge color
edge.width	Edge width, defaults to 1
edge.arrow.size	Arrow size, defaults to 1
edge.arrow.width	Arrow width, defaults to 1
edge.lty	Line type, could be 0 or "blank", 1 or "solid", 2 or "dashed", 3 or "dotted", 4 or "dotdash", 5 or "longdash", 6 or "twodash"
edge.label	Character vector used to label edges
edge.label.family	Font family of the label (e.g."Times", "Helvetica")
edge.label.font	Font: 1 plain, 2 bold, 3, italic, 4 bold italic, 5 symbol
edge.label.cex	Font size for edge labels
edge.curved	Edge curvature, range 0-1 (FALSE sets it to 0, TRUE to 0.5)
arrow.mode	Vector specifying whether edges should have arrows, possible values: 0 no arrow, 1 back, 2 forward, 3 both
OTHER	
margin	Empty space margins around the plot, vector with length 4
frame	if TRUE, the plot will be framed
main	If set, adds a title to the plot
sub	If set, adds a subtitle to the plot

# Igraph basics

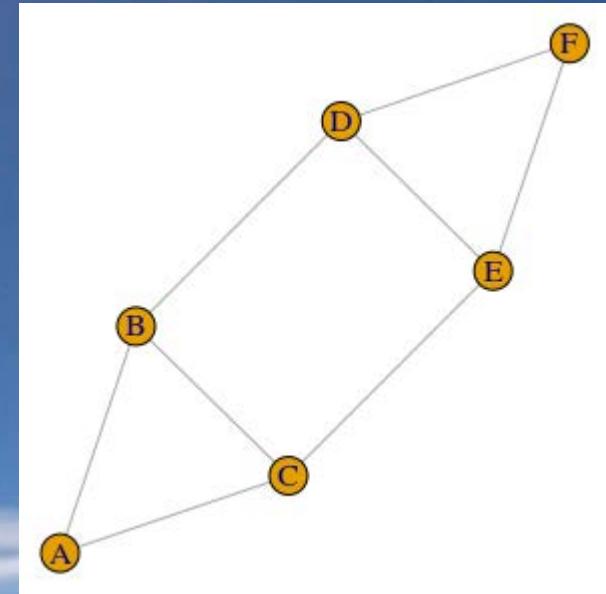
- Measures

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)
plot(g)

diameter(g) #length of the longest geodesic
> [1] 3

get_diameter(g) #path with the actual diameter
+ 4/6 vertices, named:
[1] A B D F

farthest_vertices(g) #a list with
$vertices                  # the 2 farthest vertices
+ 2/6 vertices, named:
[1] A F
$distance                  # the distance
[1] 3
```



# Igraph basics

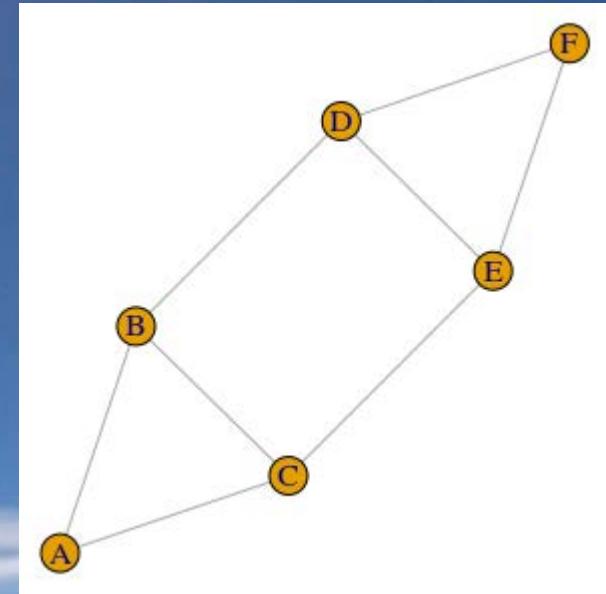
- Measures: diameter

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)
plot(g)

diameter(g) #length of the longest geodesic
> [1] 3

get_diameter(g) #path with the actual diameter
+ 4/6 vertices, named:
[1] A B D F

farthest_vertices(g) #a list with
$vertices                  # the 2 farthest vertices
+ 2/6 vertices, named:
[1] A F
$distance                  # the distance
[1] 3
```



# Igraph basics

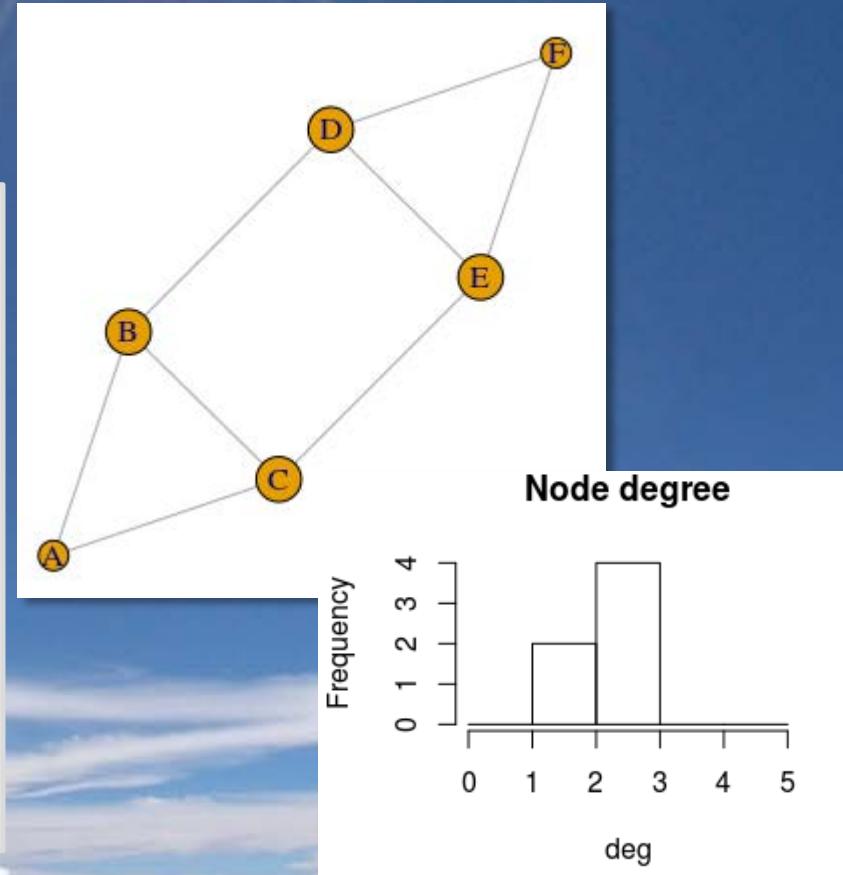
- Measures: degree

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)
plot(g)

deg <- degree(g, mode="all")
#deg
#A B D F E C
#2 3 3 2 3 3    # 2 nodes with degree 2
                  # 4 nodes with degree 3

plot(g, vertex.size=deg*6)

#The histogram
hist(deg,
      breaks=1:vcount(g)-1,
      main="Node degree")
```



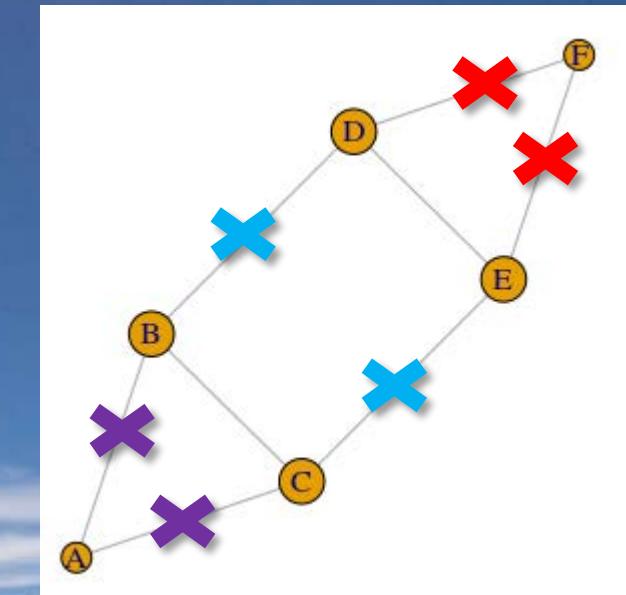
# Igraph basics

- Measures: edge connectivity

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)
plot(g)

#Minimum number of edges needed to remove
#to obtain a graph which is not strongly connected.
edge_connectivity(g)
#[1] 2

#You can get the edges with min_cut.
#WARNING! If several min cuts are...
# ...available returns the first one ONLY.
min_cut(g, value.only = F)$cut
#+ 2/8 edges (vertex names):
#[1] D--F F--E
```



# Igraph basics

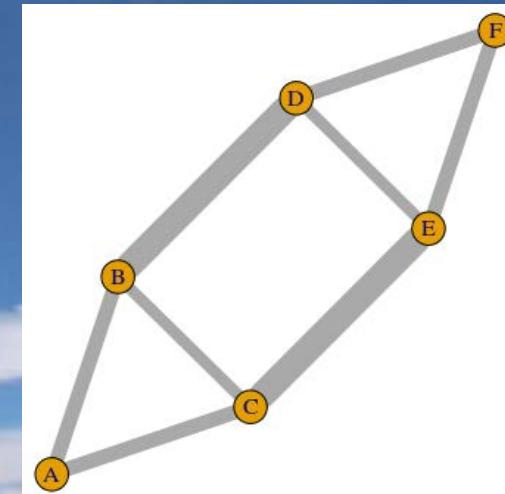
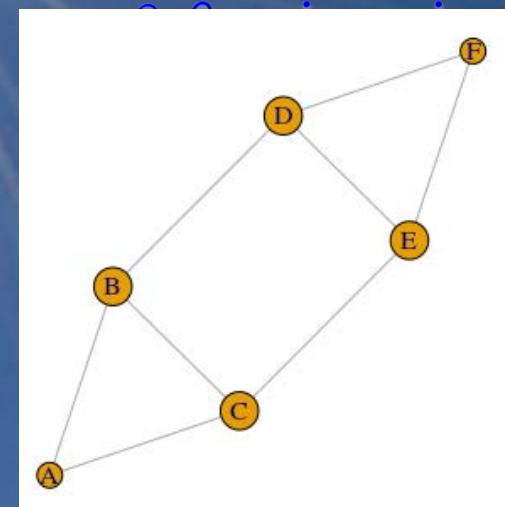
- Measures: edge betweenness

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)

#is (roughly) defined by the number of geodesics
#(shortest paths) going through an edge.
#Similar to vertex betweenness.
betw <- edge_betweenness(g,
                         directed = FALSE)

betw
#[1] 2.5 2.5 4.5 2.0 2.5 2.0 2.5 4.5
E(g)
#[1] A--B A--C B--D B--C D--F D--E F--E E--C

plot(g,
      edge.width=betw*4)
```



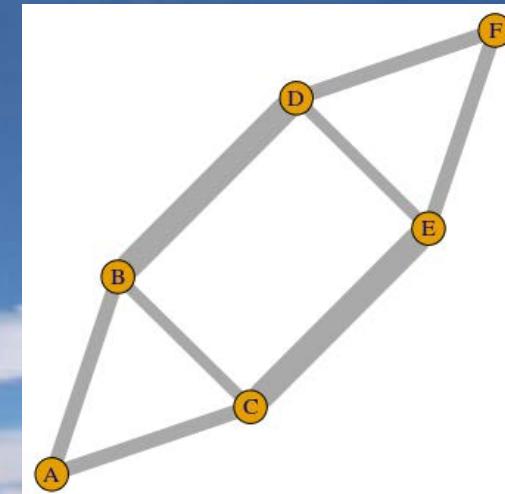
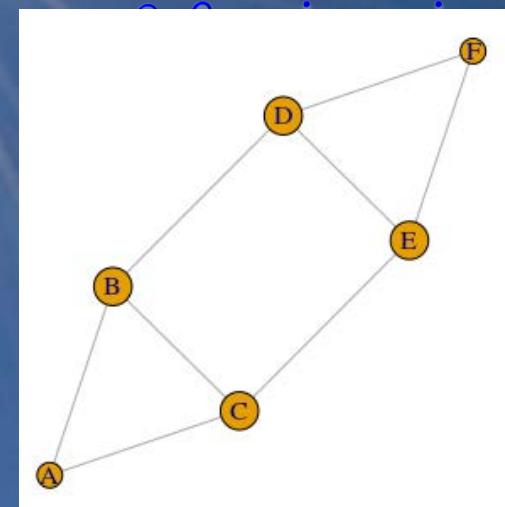
# Igraph basics

- Attributes

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)

#is (roughly) defined by the number of geodesics
#(shortest paths) going through an edge.
#Similar to vertex betweenness.
betw <- edge_betweenness(g,
                           directed = FALSE)

betw
#[1] 2.5 2.5 4.5 2.0 2.5 2.0 2.5 4.5
E(g)
#[1] A--B A--C B--D B--C D--F D--E F--E E--C
E(g)$betw <- betw
plot(g)
```



# Selecting

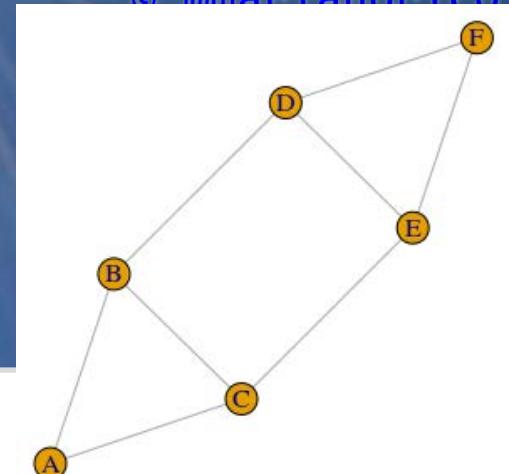
- `incident()`, `from()` and `to()`

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)
```

#`incident` takes a vertex sequence, and selects all edges that have at least one incident vertex in the vertex sequence.

```
#mode = c("all", "out", "in", "total")
```

```
incident(g, V(g)[name == "C"], mode="all")
# + 3/8 edges (vertex names):
# [1] A--C B--C E--C
```

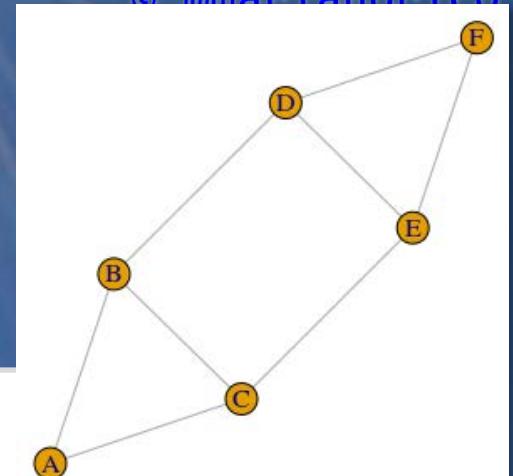


# Selecting

- `incident()`, `from()` and `to()`

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)

#g[[from=seq]] returns vertices
#g[[from=seq, edges=TRUE]] returns edges
g[[from = 1:2]] #equivalent to g[[from = c("A", "B")]]
  #$A
  #+ 2/6 vertices, named:
  #[1] B C
  #$B
  #+ 3/6 vertices, named:
  #[1] A D C
g[[from = 1:2, edges = T]] #returns the edges from nodes A and B
```



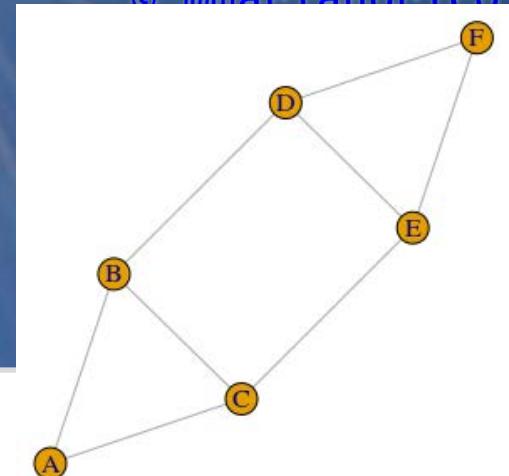
# Selecting

- `incident()`, `from()` and `to()`

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)
```

```
#g[[to=seq]] returns vertices
#g[[to=seq, edges=TRUE]] returns edges
g[[to = c("E")]] # vertices to node E
#$E
#+ 3/6 vertices, named:
#[1] D F C

g[[to = c("E"), edges=T]] #edges to node E
$E
+ 3/8 edges (vertex names):
[1] D--E F--E E--C
```

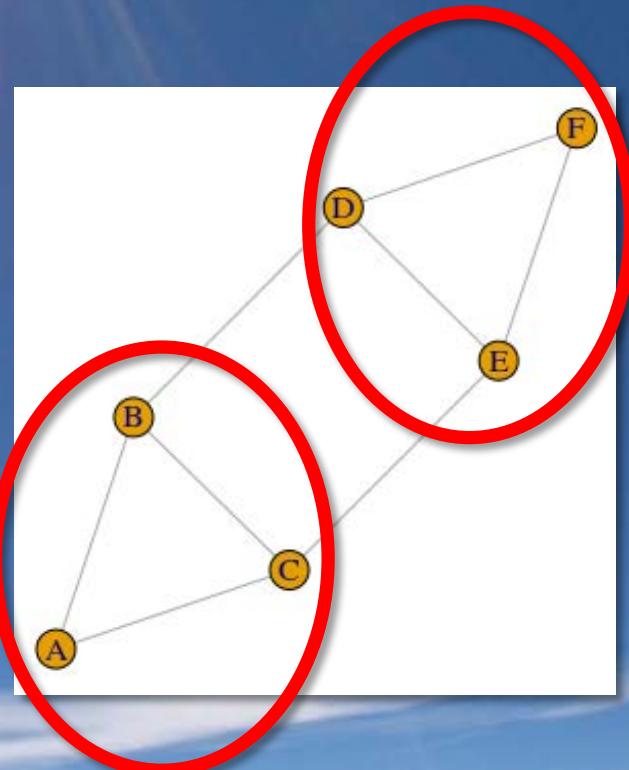


# Selecting

- Operators `%--%`, `%->%`, `%<-%`

```
set.seed(222)
g <- graph_from_literal(A-B-D-F-E-C-A, B-C, D-E)

# Select all edges between two sets of vertices.
# vertices between set (A-B-C) and set(D-E-F)
E(g)[V(g)[name <= "C"] %--% V(g)[name >= "D"]]
  #+ 2/8 edges (vertex names):
  #[1] B--D E-C
```



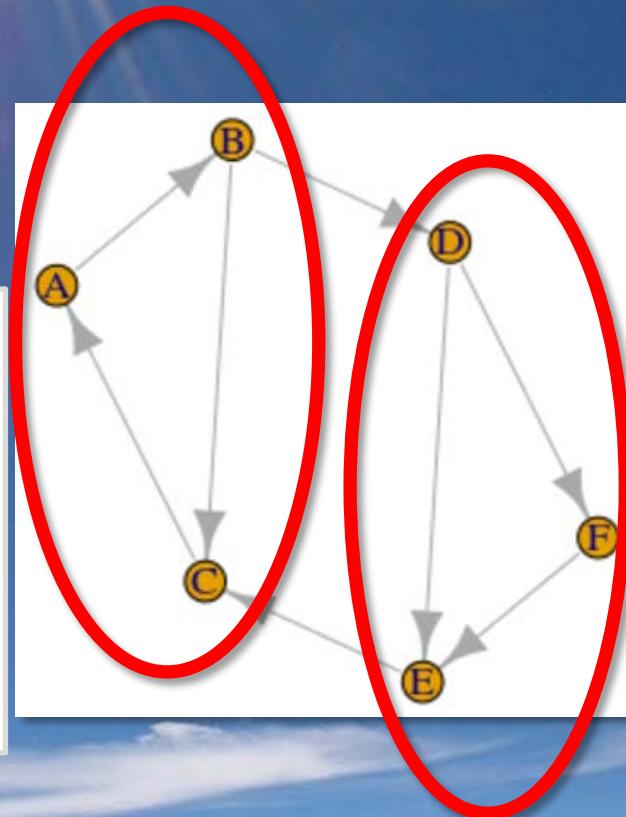
# Selecting

- Operators `%--%`, `%->%`, `%<-%`

```
set.seed(666)
g <- graph_from_literal(A--B-->D-->F-->E-->C-->A, B-->C, D-->E)

# Edges from set (A-B-C) to set(D-E-F).
E(g)[V(g)[name <= "C"] %->% V(g)[name >= "D"]]
  # + 1/8 edge (vertex names):
  # [1] B->D

E(g)[V(g)[name <= "C"] %<-% V(g)[name >= "D"]]
  # + 1/8 edge (vertex names):
  # [1] E->C
```

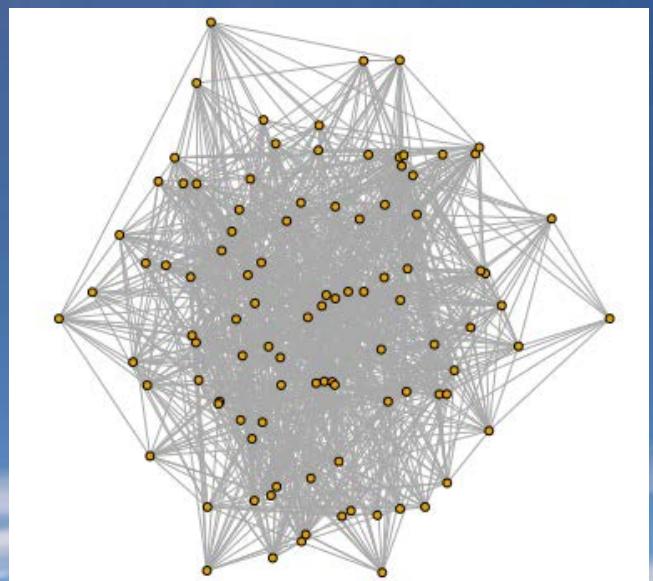


# Igraph automatic graphs

- Erdos-Renyi model
  - `sample_gnm()`.  $G(n, m)$  model.
    - $n$  vertices,  $m$  edges (selected randomly from all possible edges)

```
library(igraph)
set.seed(123)
g <- as.undirected(sample_gnm(n=100, m=1000))
l <- layout_with_drl(g, dim=3) #3D positions
plot(g, layout=l, vertex.size=3, vertex.label=NA)
```

`edge_density(g) = 0.202`

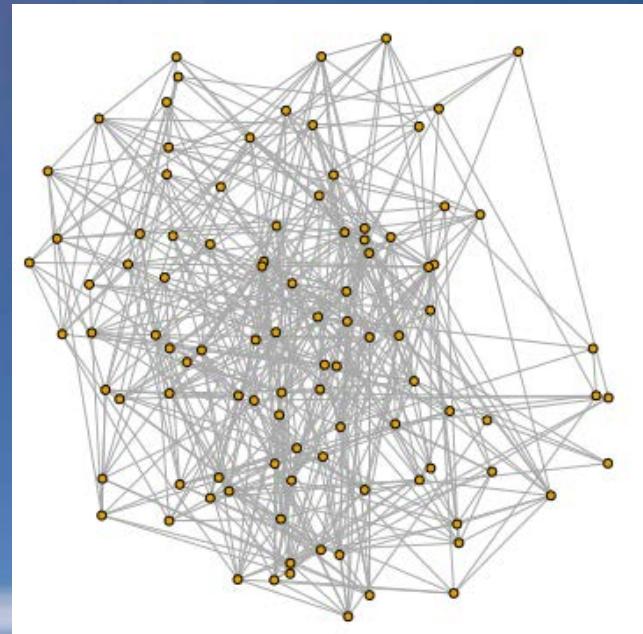


# Igraph automatic graphs

- Erdos-Renyi model
  - `sample_gnp()`. G( $n, p$ ) model.
    - $n$  vertices,  $p$  probability of creating an edge between 2 random vertices

```
library(igraph)
set.seed(123)
g <- as.undirected(sample_gnp(n=100, p=0.1))
l <- layout_with_drl(g, dim=3) #3D positions
plot(g, layout=l, vertex.size=3, vertex.label=NA)
```

`edge_density(g) = 0.096 (~0.1)`

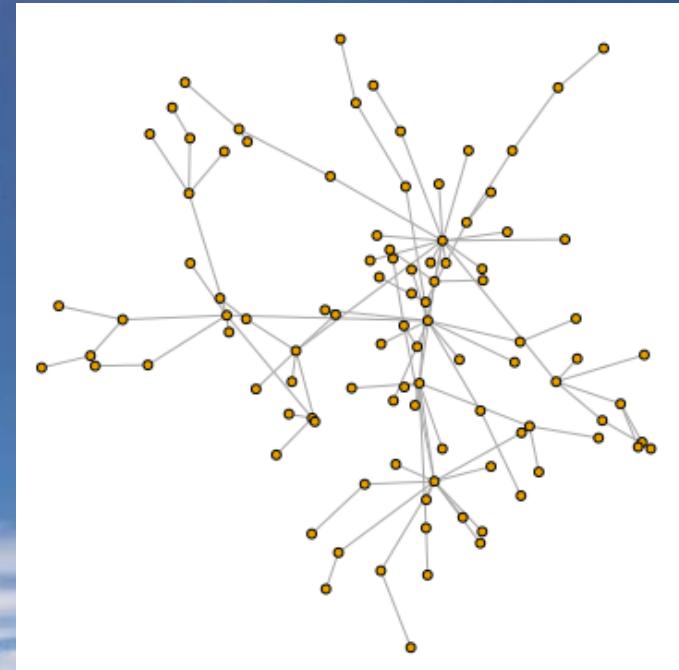


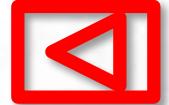
# Igraph automatic graphs

- Barabasi-Albert model
  - `sample_pa()`.
    - n vertices, many more params (with default values)

```
library(igraph)
set.seed(123)
g <- as.undirected(sample_pa(n=100))
l <- layout_with_drl(g, dim=3) #3D positions
plot(g, layout=l, vertex.size=3, vertex.label=NA)
```

`edge_density(g) = 0.02`



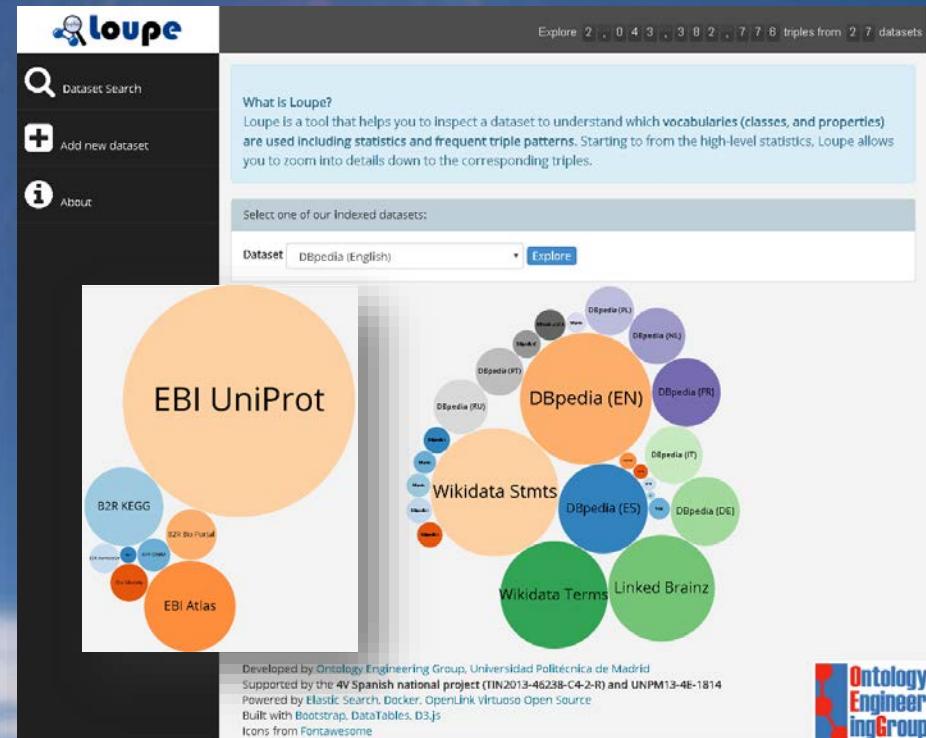
A photograph showing a group of hikers with backpacks and ski poles walking along a snowy mountain slope. The sun is low, casting long shadows on the snow.

Exploring a given dataset

# LOUPE

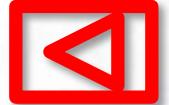
# Loupe

- Explore datasets
  - Two flavors
    - Regular Loupe
      - Popular LOD datasets
        - » 27 datasets
        - » 2.0 billion triples
    - BioLoupe
      - Popular Bio datasets
        - » 8 datasets
        - » 4.8 billion triples



# Loupe

- Better an online demo ☺

A photograph of a group of hikers walking up a steep, snow-covered mountain slope. They are wearing various colored jackets and backpacks, and some are using ski poles. The sun is shining brightly, creating strong shadows on the snow.

# A first glimpse of graphs with **RELFINDER**

# RelFinder

- A first graphical exploration
- Configurable for any SPARQL endpoint
  - By default: DBpedia
- Go!



Main Developers

- Philipp Heim
- Steffen Lohmann
- Timo Stegemann

Visual Data Web

Visually Experiencing the Data Web

Home Tools Publications People

Home > Tools > RelFinder

**RelFinder** → Interactive Relationship Discovery in RDF Data

Are you interested in how things are related with each other? The RelFinder helps to get an overview: It extracts and visualizes relationships between given objects in RDF data and makes these relationships interactively explorable. Highlighting and filtering features support visual analysis both on a global and detailed level. The RelFinder is based on the open source framework [Adobe Flex](#), easy-to-use and works with any RDF dataset that provides standardized SPARQL access.

Experience RelFinder

Watch Video

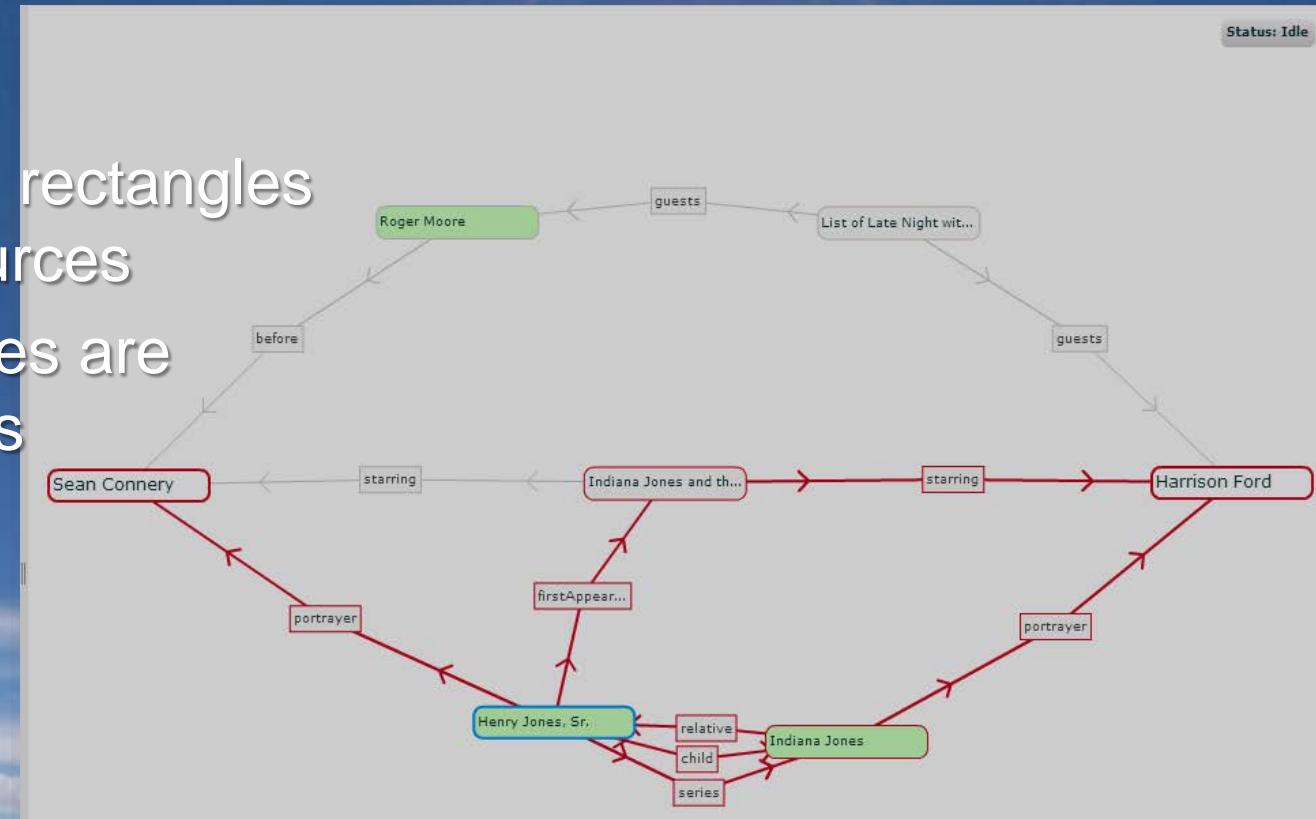
# RelFinder

- Look for the relationship between Sean Connery and Harrison Ford



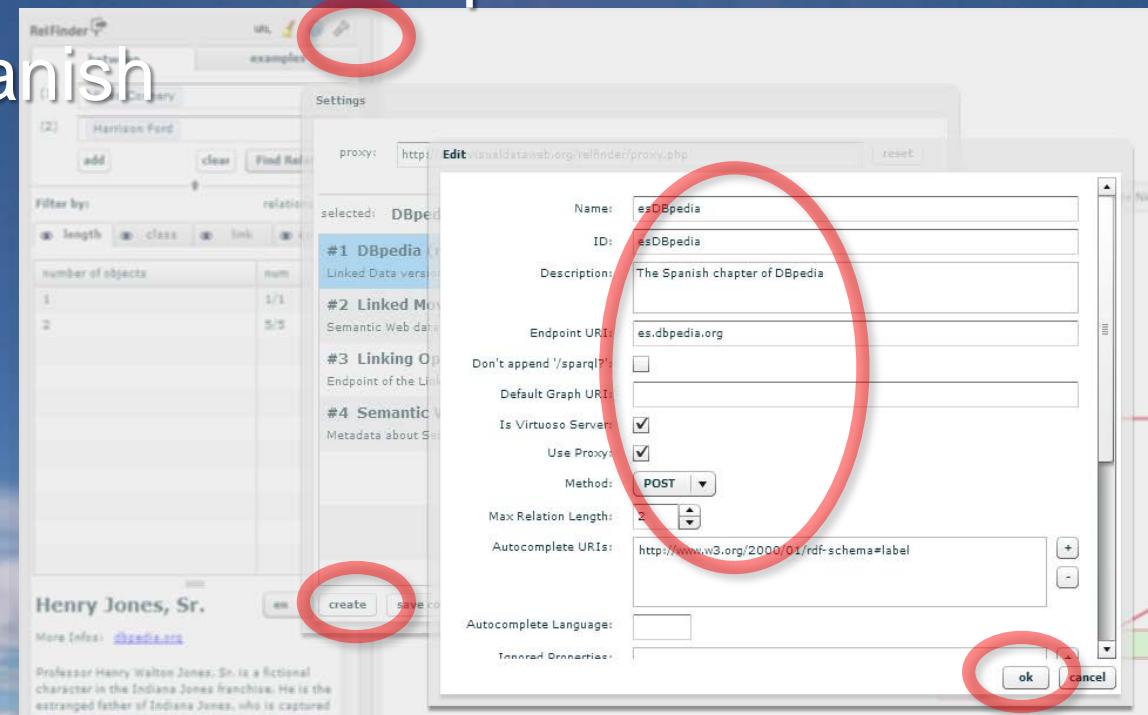
# RelFinder

- Result
  - Rounded rectangles are resources
  - Rectangles are properties



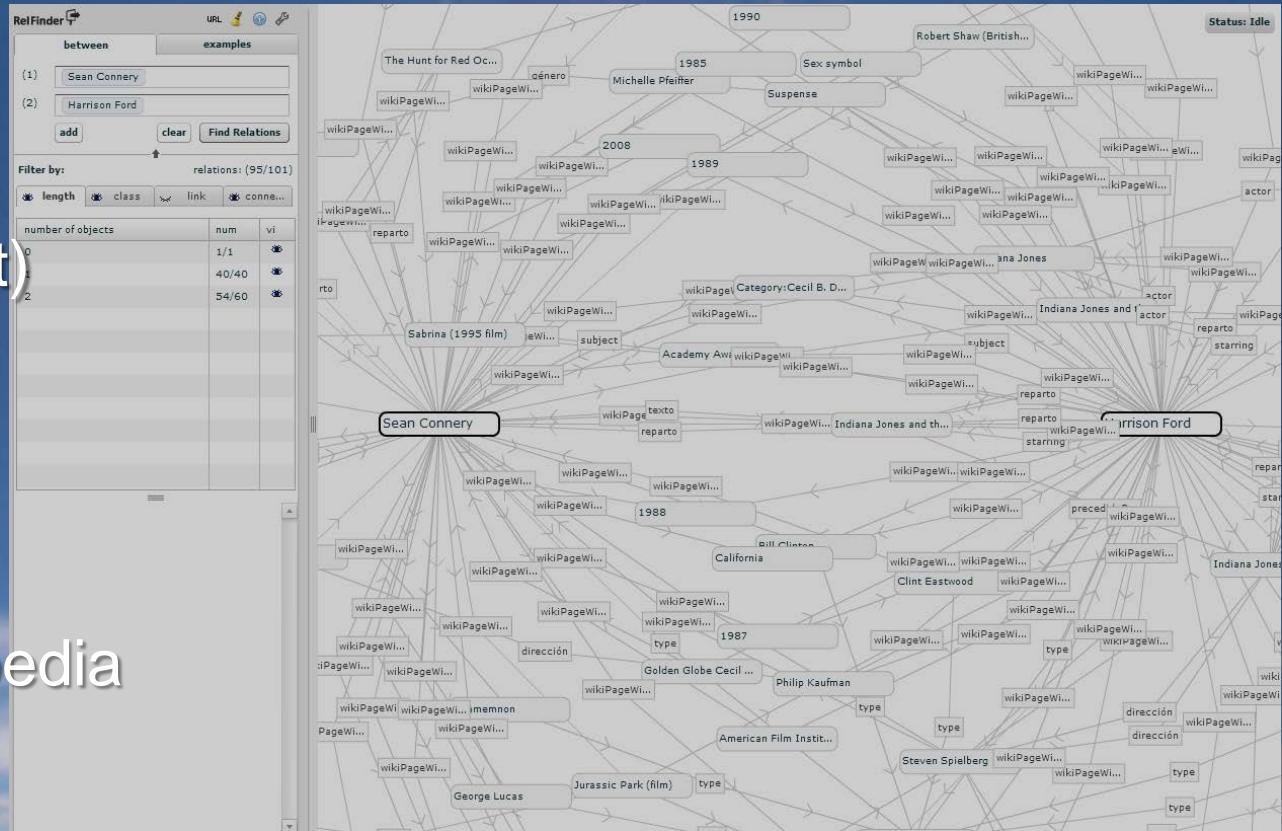
# RelFinder

- Setup for other SPARQL endpoints
  - E.g. The Spanish DBpedia



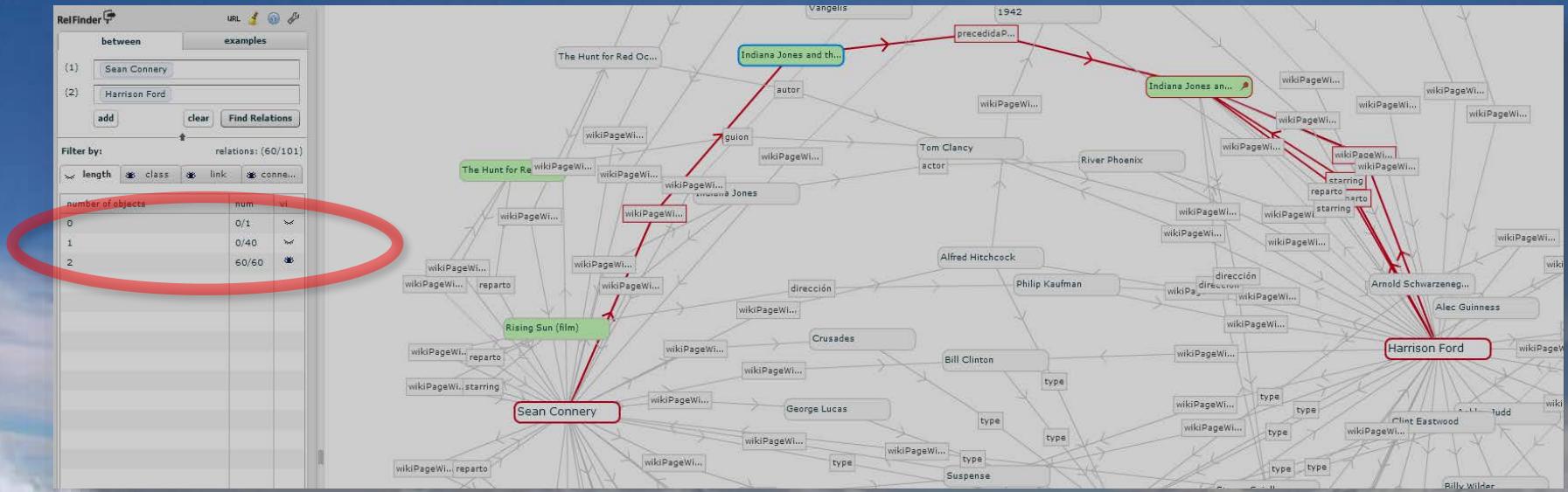
# RelFinder

- Result
  - Much more than (default) DBpedia
  - Hypothesis:  
They use a old private copy of DBpedia



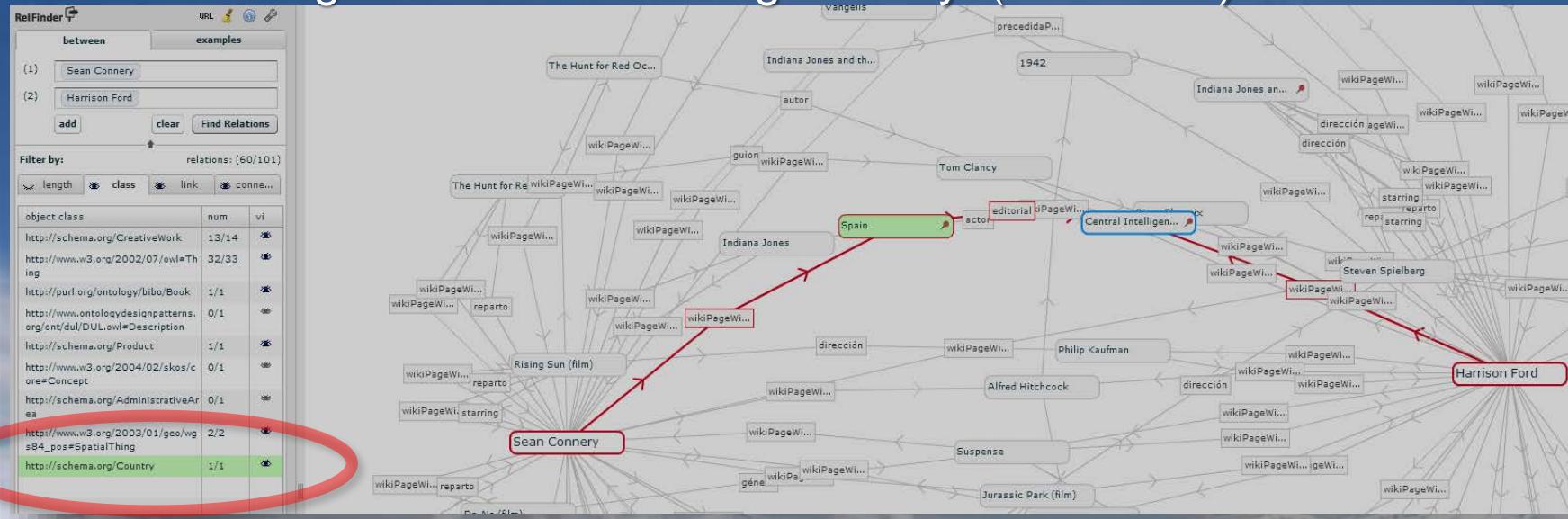
# RelFinder

- Filter by
  - Length (number of resources in between)
    - E.g. length = 2 → 2 resources in between



# RelFinder

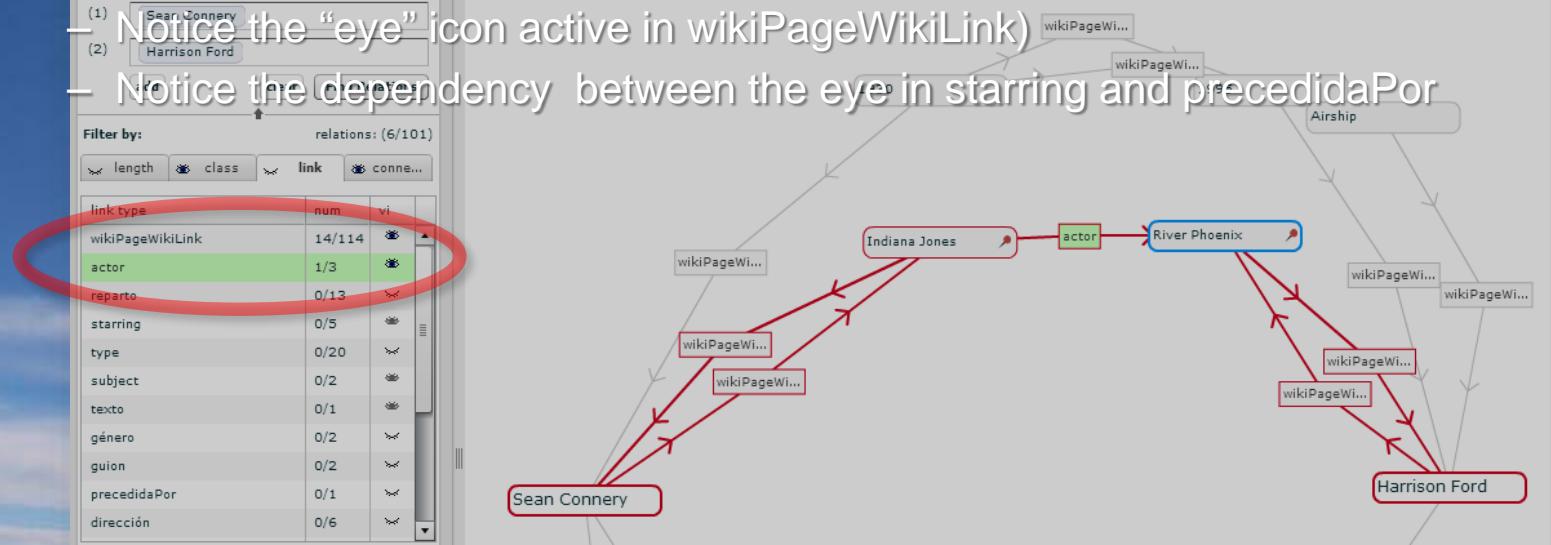
- Filter by
  - Class (list of resource's class)
    - E.g. class = schema.org/Country (num = 1/1)



# RelFinder

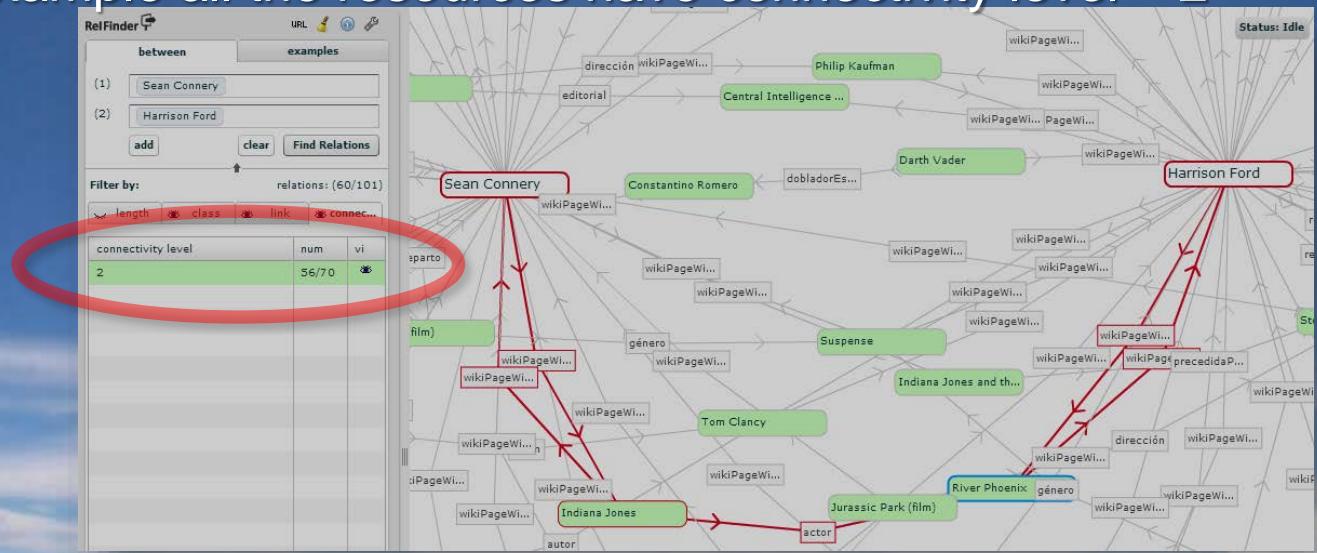
- Filter by
  - Link (list of resource's properties)

- E.g. link = actor



# RelFinder

- Filter by
  - Connectivity level (number of resources in between)
    - In this example all the resources have connectivity level = 2

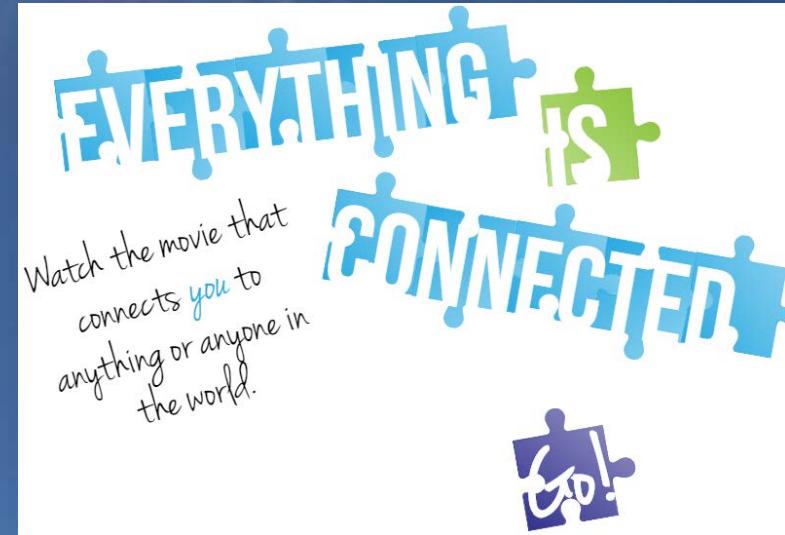


# RelFinder

- Limitations
  - Oriented to resources
    - 2 resources (or more)
  - Very noisy
    - properties as graph nodes

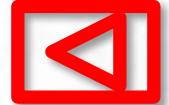
# After RelFinder

- EveryThingIsConnected
  - Up to 15 hops
    - 2 resources (or more)
  - Creates a movie with the explanation
  - Has a REST API
    - E.g.: Rajoy ↔ Arabidopsis



My record: 11 hops

AS YOU CAN SEE, MARIANO RAJOY  
IS CONNECTED TO EVERYTHING IN THIS WORLD,  
INCLUDING ARABIDOPSIS THALIANA!

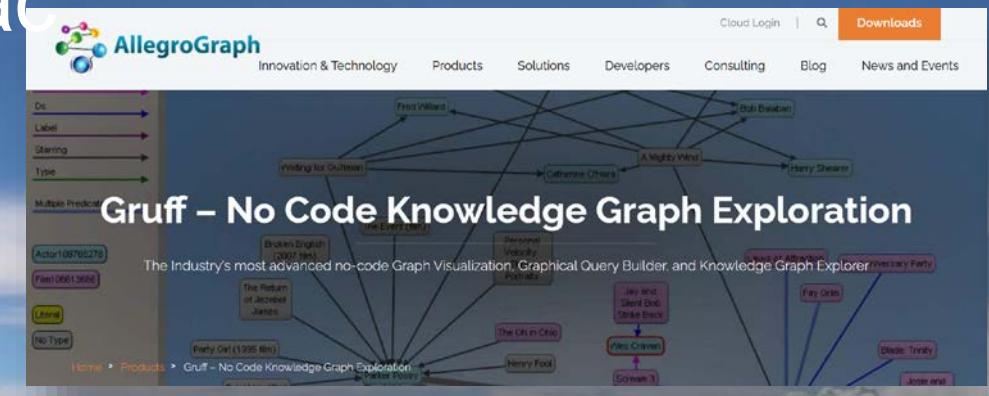


Ordered small graphs

**GRUFF**

# Gruff

- Private product (evaluation license)
- Last version includes access to EPs
  - Local installation or in cloud
- Windows/Linux/Mac



# Gruff

- Windows/Linux/Mac

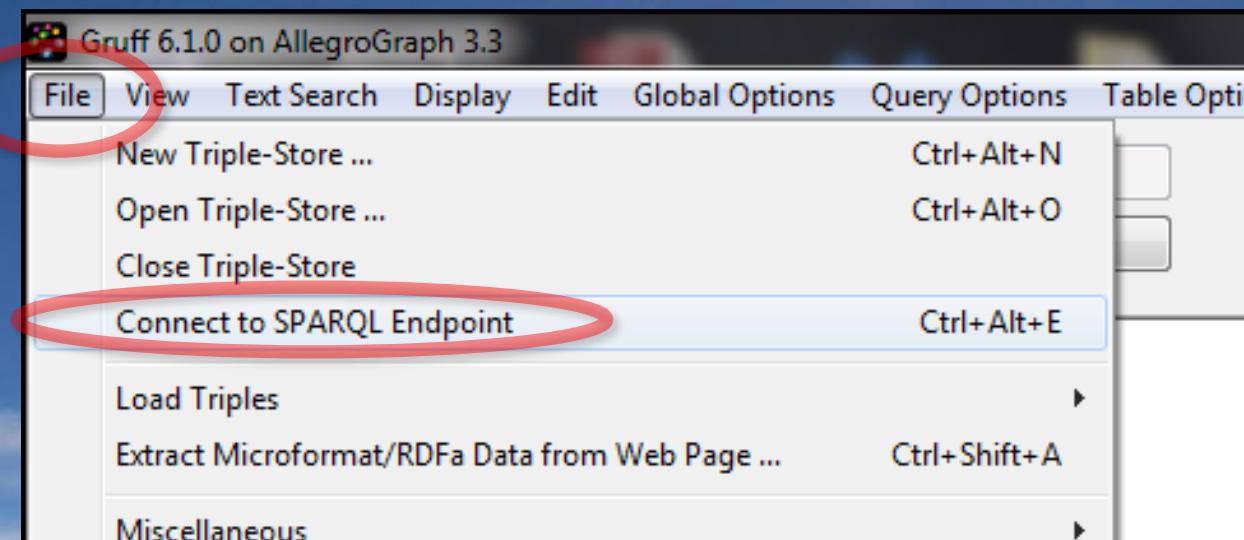
## Gruff for AllegroGraph 3.3

A Standalone Gruff with built-in AllegroGraph 3.3, no Additional Download Required:

Platform	Download	System Requirements
64-bit Linux	<a href="#">Gruff v6.1.0-AG3.3</a>	glibc 2.4 or newer -- <a href="#">Installation instructions</a>
32-bit Linux	<a href="#">Gruff v6.1.0-AG3.3</a>	glibc 2.3 or newer -- <a href="#">Installation instructions</a>
64-bit Mac OS X	<a href="#">Gruff v6.1.0-AG3.3</a>	Mac OS X 10.6 or newer -- <a href="#">Installation instructions</a>
64-bit Windows	<a href="#">Gruff v6.1.0-AG3.3</a>	Windows Vista or newer -- <a href="#">Installation instructions</a>
32-bit Windows	<a href="#">Gruff v6.1.0-AG3.3</a>	Windows Vista or newer -- <a href="#">Installation instructions</a>

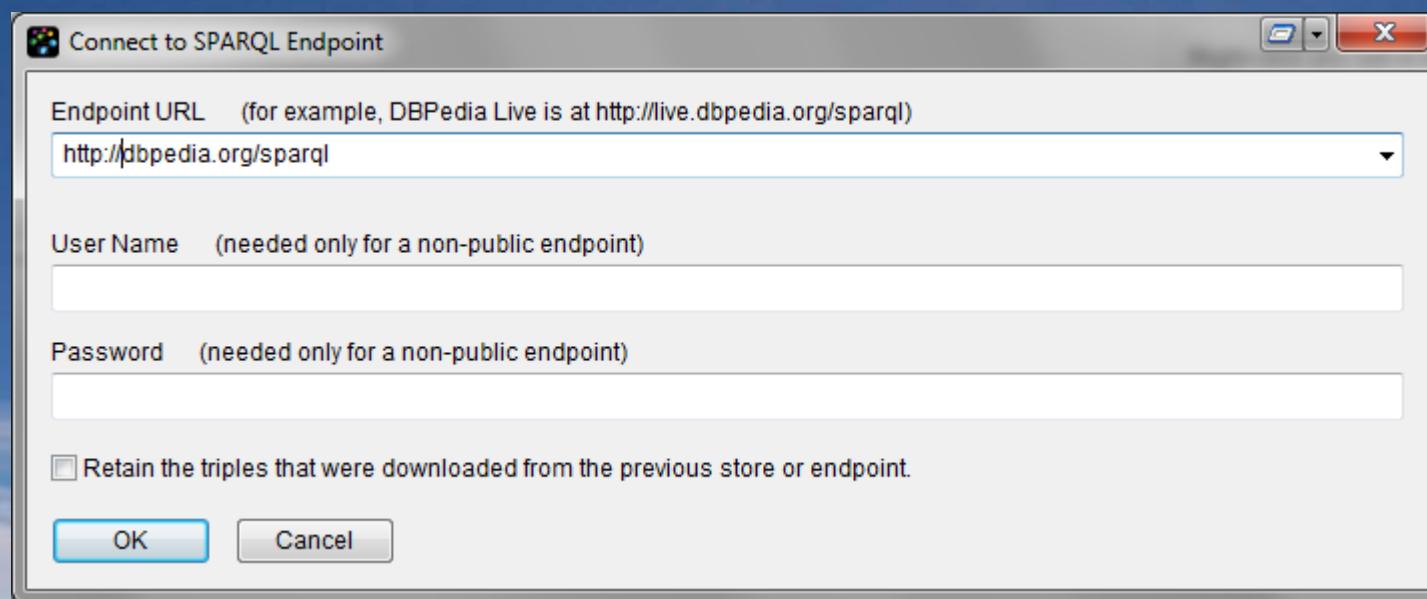
# Gruff

- Specify the EP



# Gruff

- Specify the EP



# Gruff

- View → Query View

The screenshot shows the Gruff 6.1.0 application window. At the top, there's a menu bar with File, View, Text Search, Display, Edit, Global Options, Query Options, Table Options, and Help. Below the menu is a toolbar with buttons for SPARQL (selected), Use Planner, Reindent, Name Query, Revisit, Run Query, Select All, Graph View, Table View, and Graphical Query View.

The main area is divided into several sections:

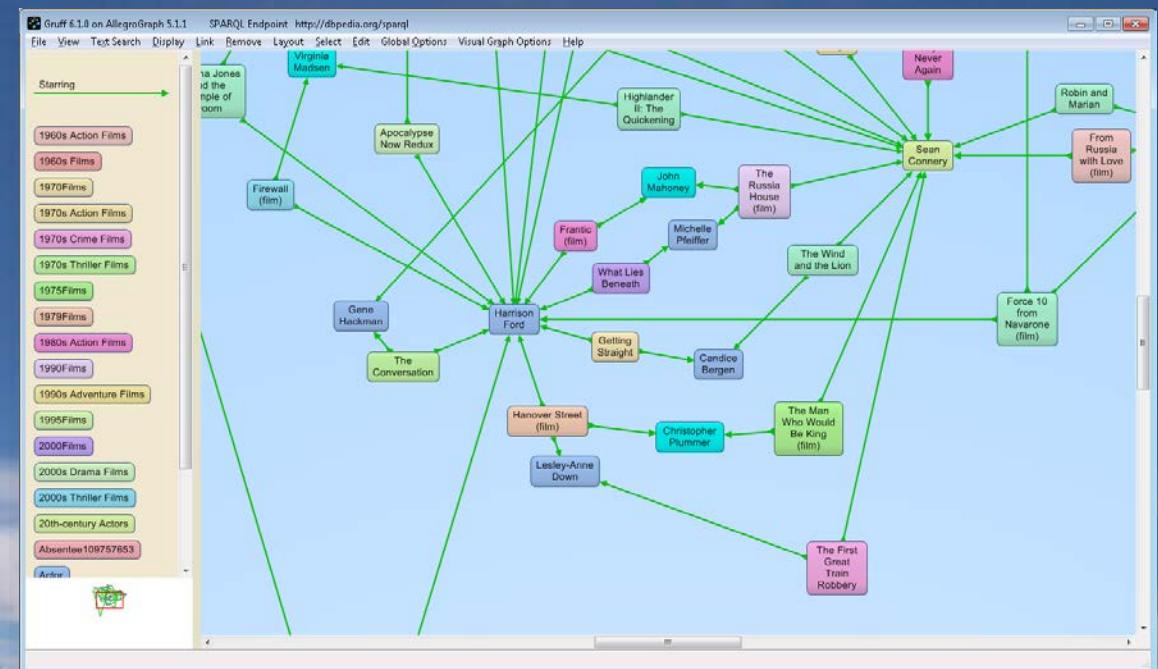
- Query:** A text input field containing a SPARQL query:

```
prefix dbpp: <http://dbpedia.org/property/>
prefix dbpr: <http://dbpedia.org/resource/>

# Find actors who were in different films with Harrison Ford and Sean Connery.
select ?actor ?second_film ?first_film where
{ ?first_film dbpp:starring dbpr:Harrison_Ford ,
  ?actor .
  ?second_film dbpp:starring ?actor ,
  dbpr:Sean_Connery .
  filter ( ?actor != dbpr:Harrison_Ford )
  filter ( ?actor != dbpr:Sean_Connery )
  filter ( ?second_film != ?first_film )
} limit 100
```
- Results:** A large empty table where results would be displayed.
- Buttons:** Create Visual Graph, Add to Visual Graph, Write Text Report, Save as CSV.
- Help/Instructions:** A panel on the right with instructions for using the application.
- Tables:** Two smaller tables at the bottom labeled "Explicit Nodes from Query" and "Explicit Predicates from Query".

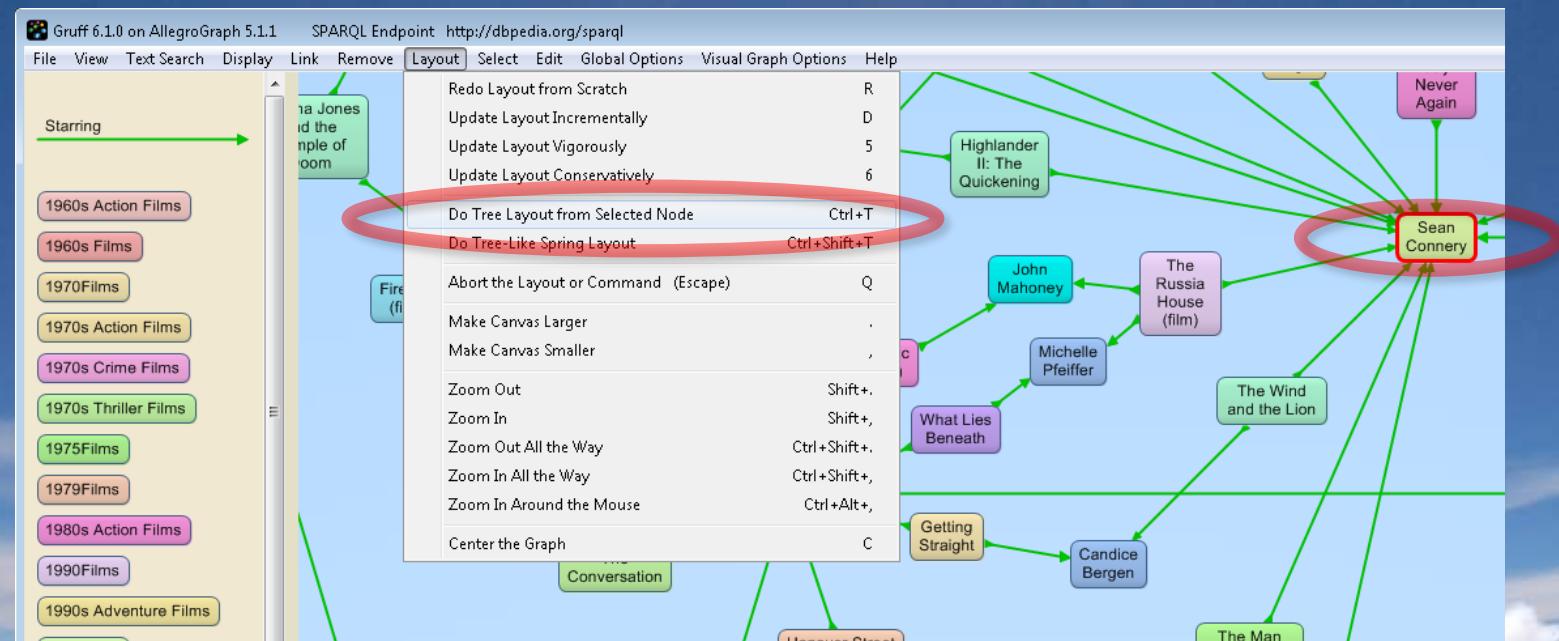
# Gruff

- SPARQL query results as a graph



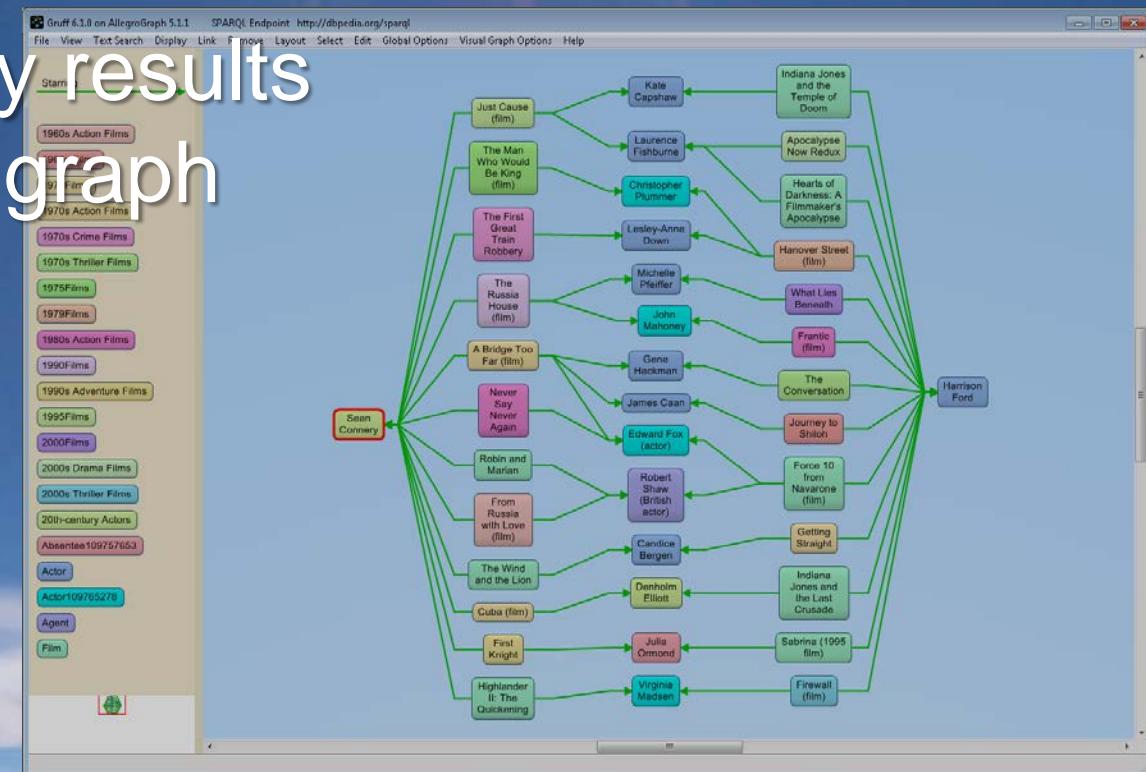
# Gruff

- SPARQL query results as a graph



# Gruff

- SPARQL query results as an ordered graph



# Gruff

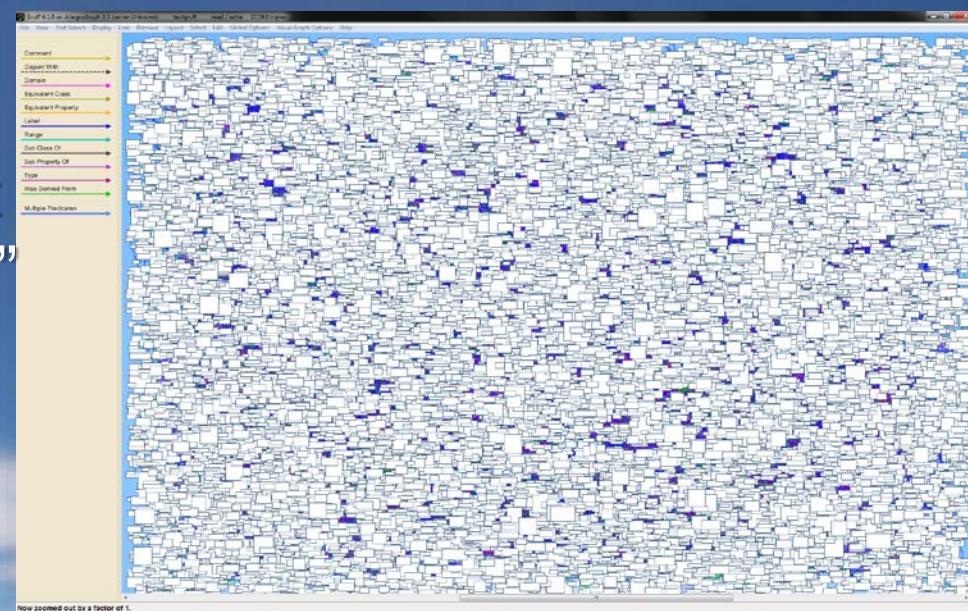
- Benefits
  - Intuitive graph (properties are hidden)
  - Nodes = Resources
  - Easy selection of nodes for a given type
- Drawbacks
  - Limited set of layouts
  - Does not work fine for large datasets
    - That is why, by default, limits the triples to display to 100
      - You can change this limit: Visual Graph Options → Inclusion Options → Maximum Samples Triple to Display (e.g. dbpedia\_2014.owl has 27,063 triples)

# Gruff

- Trying to visualize the DBpedia ontology
  - Create a new triplestore
    - Select an empty folder in your file system
  - Load RDF/XML (select dbpedia\_2014.owl)
  - Show graph (by default 100 triples)
  - Remove literals (right-click yellow box on the left → remove)
  - Increase the triple limit to 30,000
  - And....

# Gruff

- Trying to visualize the DBpedia ontology
  - And...
  - Wait almost 1 hour 😞 later you get an almost “impossible to manage” graph 😞



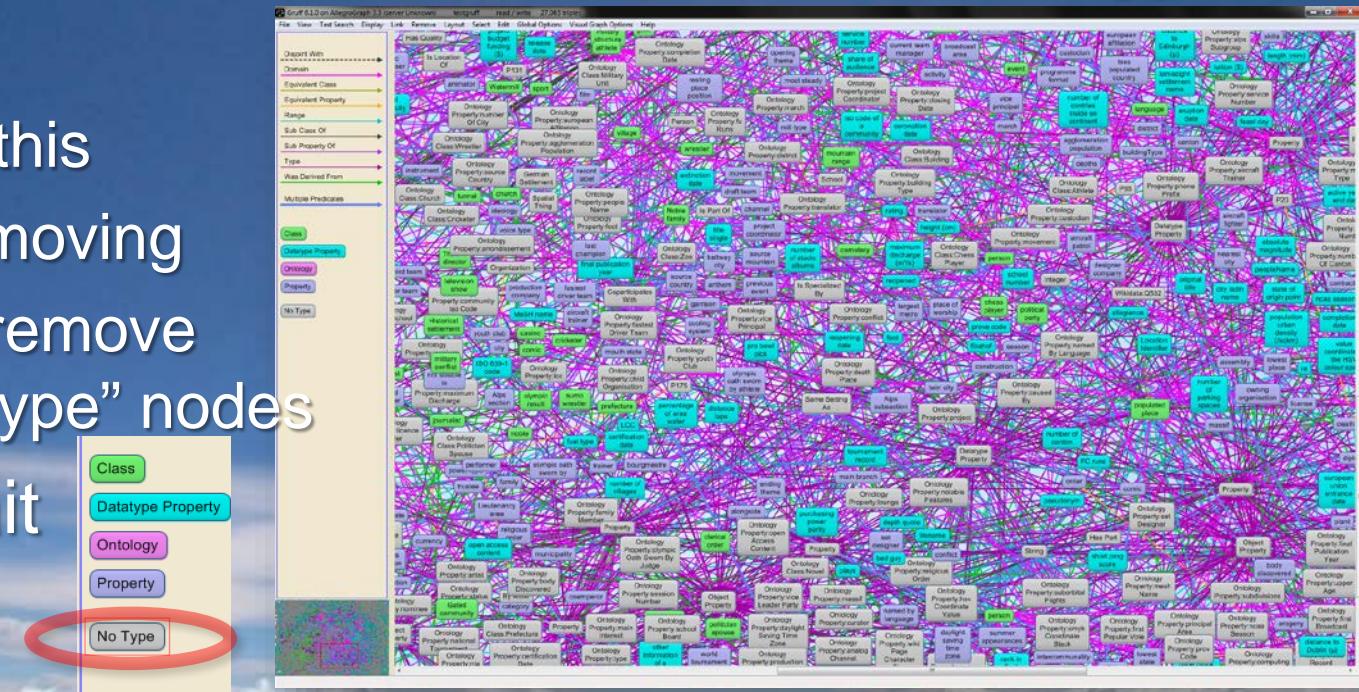
# Gruff

- Trying to visualize the DBpedia ontology
  - Zoom in
  - Remove again
  - Literals
    - Wait for hours 😞 again
  - And...



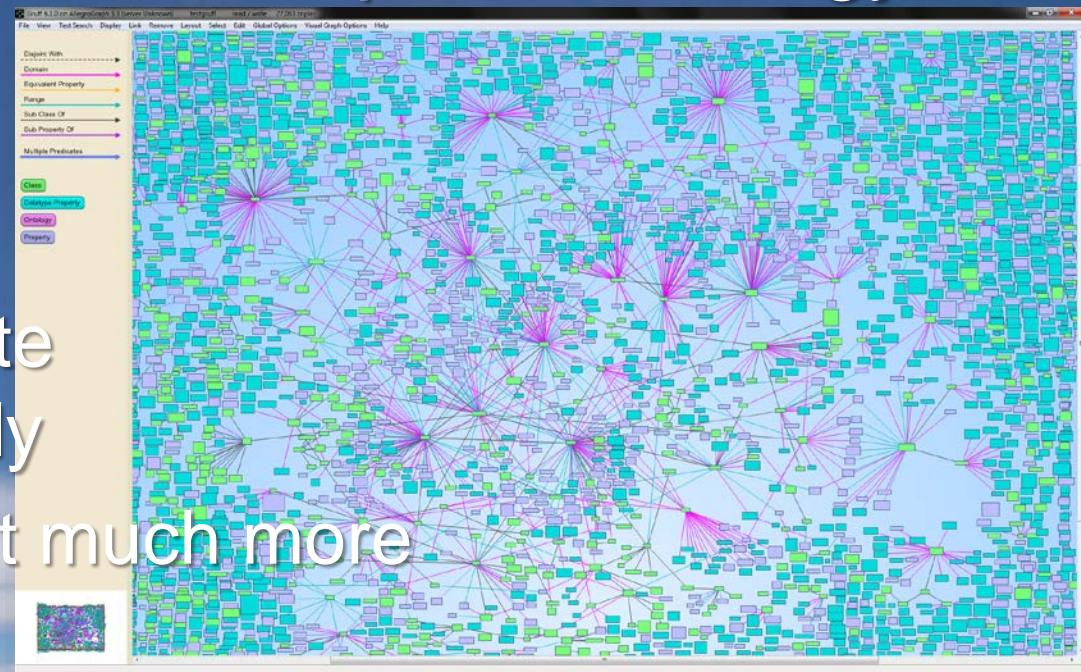
# Gruff

- Trying to visualize the DBpedia ontology
  - And...
  - You get this
  - Keep removing
    - Now remove “No Type” nodes
  - And wait



# Gruff

- Trying to visualize the DBpedia ontology
  - And...
  - After 20 min...
  - Now, apply  
Layout → Update  
layout vigorously
  - This is a dataset much more  
usable



# Homework

- Your own proposals
- My proposals
  - Social graph (e.g. Twitter (“TwitterStreamingImporter”))
    - Nodes: persons
    - Arcs: friend of (unlabeled)
  - (DBpedia) ontology graph
    - Nodes: classes. Size ~ instances
    - Arcs (labeled): properties with domain & range
      - How could you manage a “arc size”? (in this case, arc size is proportional to the number of triples using that property)
  - Covid-19 Dataset

# Homework

- A written work (pdf) made by **up to 4** people. Describe clearly the work done by each author.
- Deadline: April 14th 2025
- Plagiarism is severely prosecuted. Be clever, do not cheat!
- Procedure
  - Upload it to Moodle **and**
  - Send an email to [mariano.rico@upm.es](mailto:mariano.rico@upm.es) with
    - Subject: DSS Big Data Visualization
    - A pdf file, up to 4 pages, including at least:
      - The SPARQL query (if you use the SemanticWeb Importer) and the rationale behind the SPARQL query
      - The final graph (otherwise agreed, made with Gephi)
      - A description of the operations to achieve the final graph (reproducible results)
      - An analysis of the resulting graph
      - Do not forget to include author(s) name(s)!!.

#VizLinkedData





*Thanks for your attention*

*Mariano.Rico@upm.es*