Visualizing large datasets in the Semantic Web

#VizLinkedData

Feb 2019

@marianorico

mariano.rico@upm.es

© @marianorico

Attendance check

© @marianorico

Get the material

- Slides and files can be downloaded from

http://goo.gl/qHlHG6

© @marianorico

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
- Displaying linked data with Gephi
- Other tools
  - Loupe. Explore datasets
  - RelFinder. A first graphical approach
  - Gruff. Graph SPARQL query results

Viewing **without** visualizing

LARGE-SCALE GRAPH ANALYTICS

© @marianorico

Problem

- Huge graphs
  - Facebook: 1 billion users → 1G nodes (aka vertex) graph
  - Google: the indexed web is ~50G pages graph
- Complex/heavy algorithms
  - Page rank
  - Shortest path in a graph

© @marianorico

State of the art

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

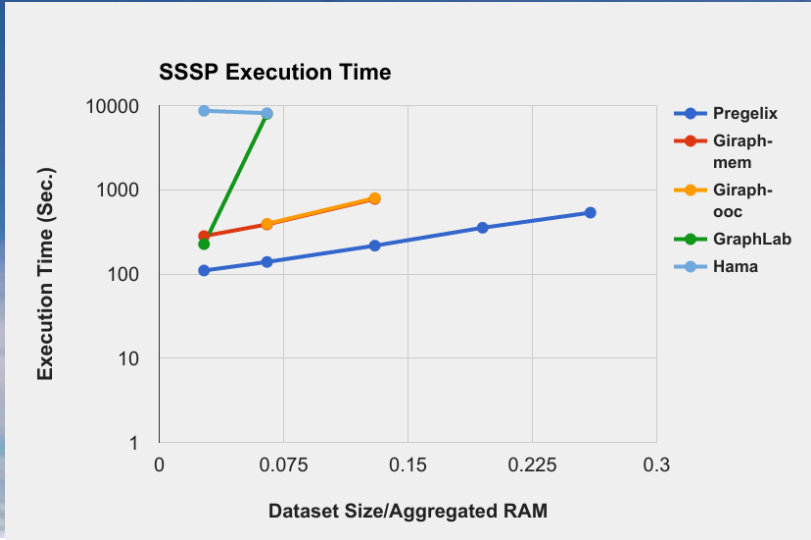
© @marianorico

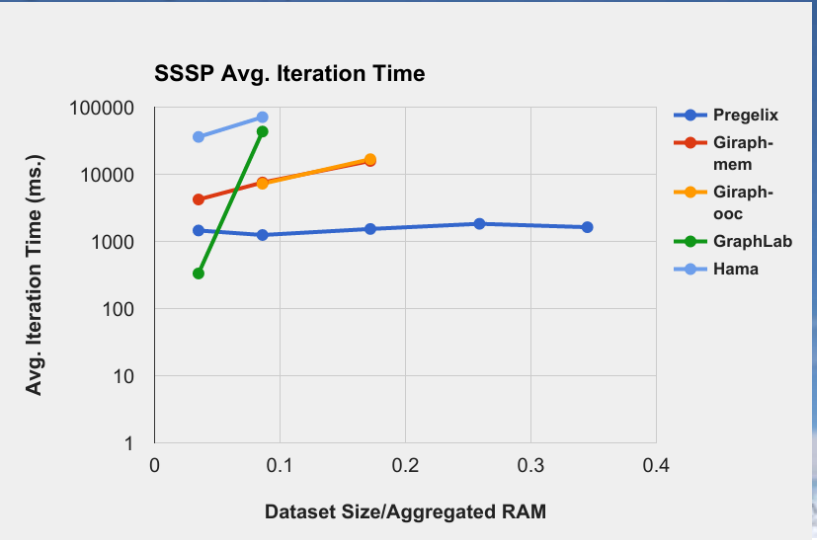
Pregelix

- Comparison with other Big Graph Analytics platforms (Giraph, Hama, ~~GraphLab~~, GraphX)
  - 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

© @marianorico

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)

© @marianorico

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

© @marianorico

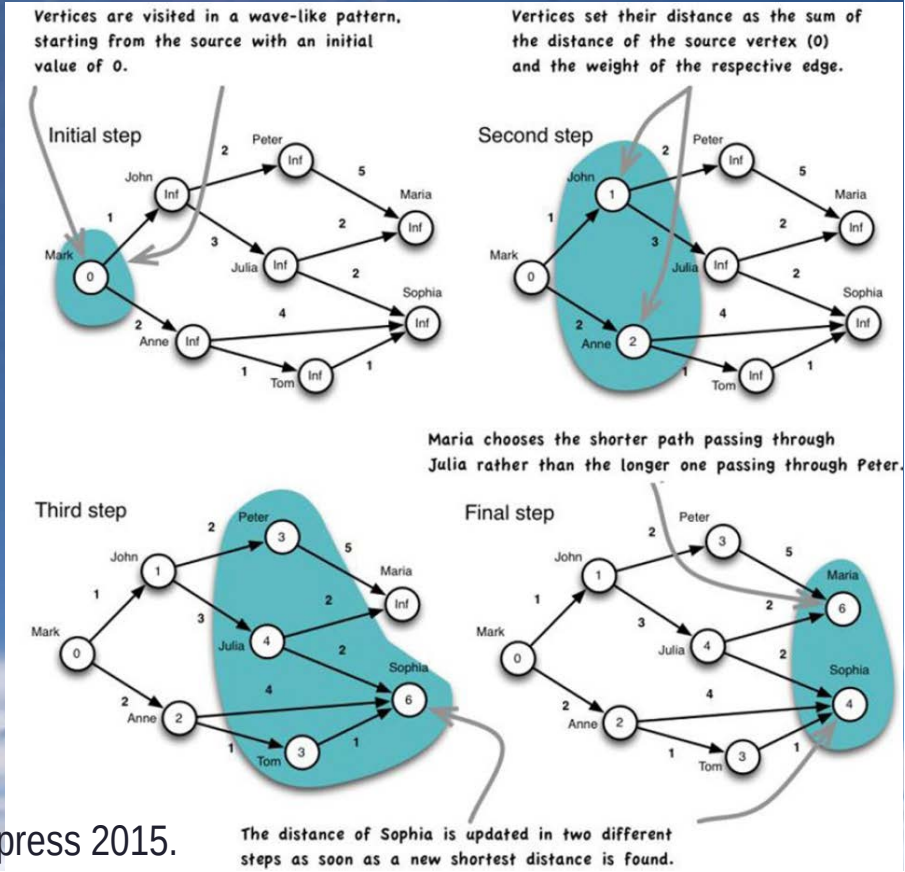
BSP method. An example

- 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

© @marianorico

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

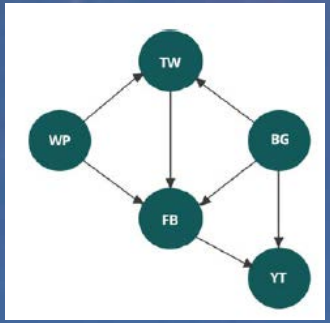


Source: Practial Graph Analytics with Apache Giraph. Apress 2015.

© @marianorico

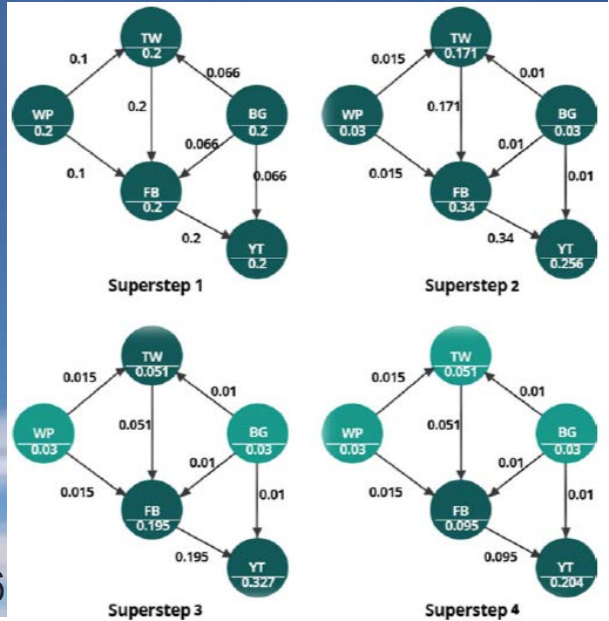
BSP method. PageRank example

- PageRank algorithm for BSP.
  - Initialization
    - Assign 0.2 (1/#vertices) to each vertex.
    - Message sent: val / #outEdges
  - In each iteration
    - Each vertex sums the incoming messages (im) * 0.85 and sums 0.15/#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();
    }
  }
}
```

© @marianorico

Algorithms complexity

- Comparison by tool

Graph Theoretic Approaches for Analyzing Large-Scale Social Networks

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

© @marianorico

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
Kamanda Kawai	No	Yes	Yes	No
Fruchterman Reingold	No	Yes	Yes	No
Force Atlas Layout	No	No	Yes	No

Graph Theoretic Approaches for Analyzing Large-Scale Social Networks

Finally!: viewing visualizing

VISUALIZING BIG GRAPHS

© @marianorico

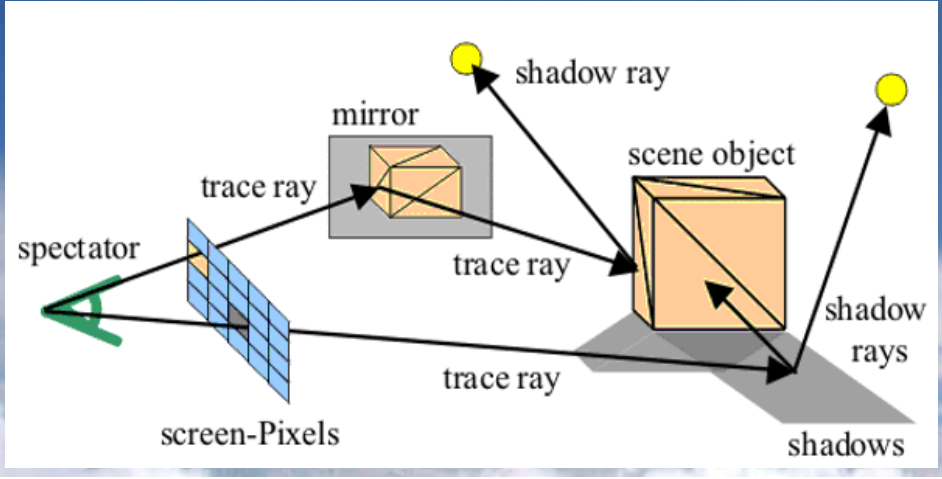
Display limits

- 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

© @marianorico

Display limits

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

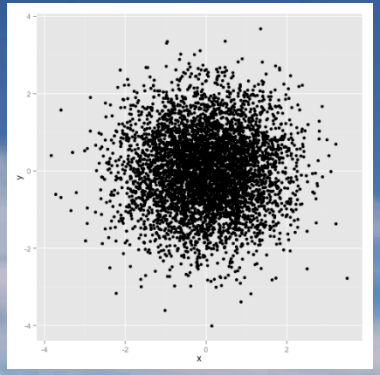


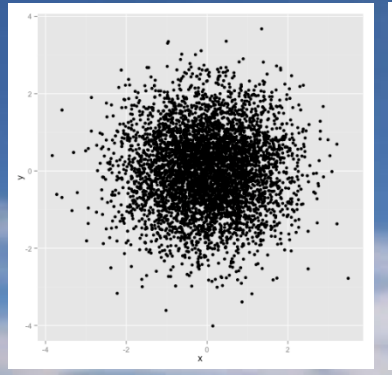
Source

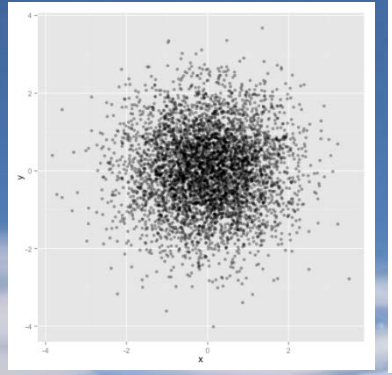
© @marianorico

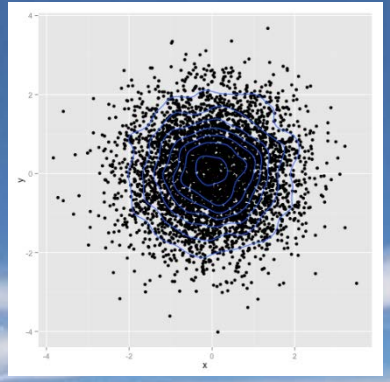
How to display

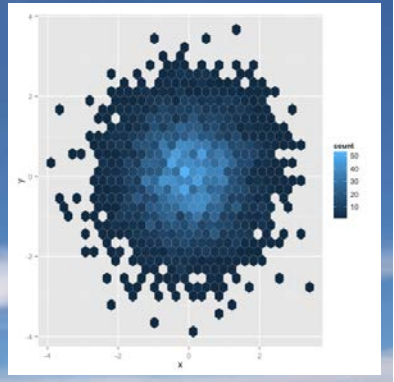
- Sometimes the problem is how to display the data
- Example: The “all black” case.
  - Use: jitter, alpha, density contour… hexbins!!









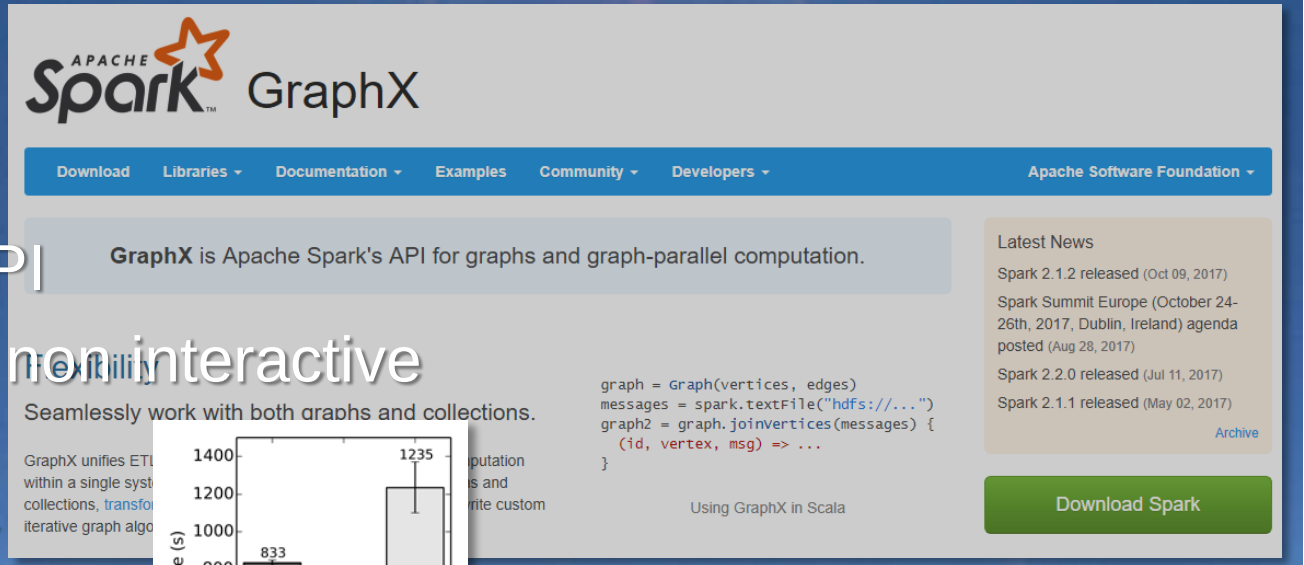


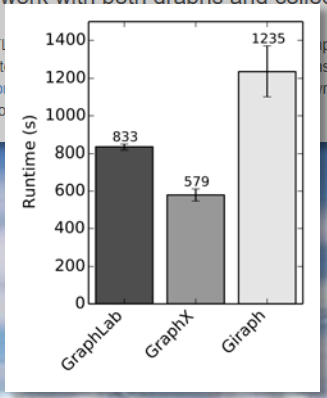
R Source (using ggplot2)

© @marianorico

How to display

- For Spark
  - GraphX
    - Pregel API
    - Fast, but non interactive
    - Doc
      - API for
        - » Java
        - » Scala
        - » Python
        - » R





© @marianorico

How to display

- For R
  - bigvis package (in RStudio, not yet in CRAN despite it's 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)

© @marianorico

How to display

- For R
  - bigvis package (in RStudio, not yet in CRAN despite it's 2013)

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

Hadley Wickham

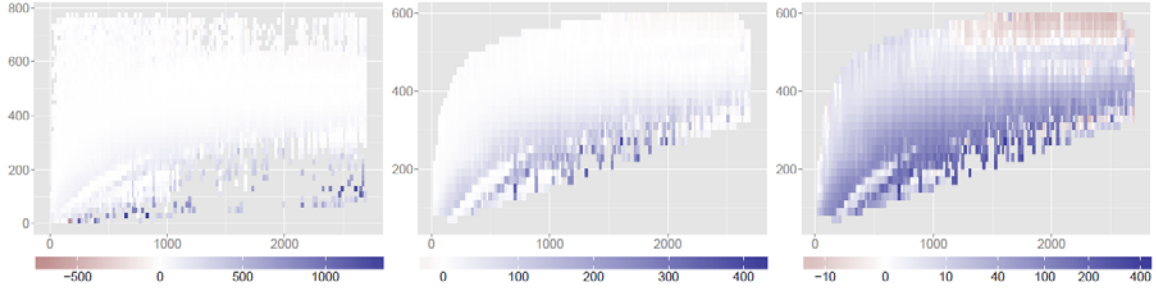


Fig. 1. Average delay (colour, in minutes) as a function of distance (x axis, in miles) and speed (y axis, in mph) for 76 million flights. The initial view (left) needs refinement to be useful: first we focus on the middle 99.5% of the data (centre) then transform average delay to shrink the impact of unusually high delays and focus on typical values (right). Flights with higher than average speeds (top-right) have shorter delays (red); more interestingly, a subset of shorter, slower flights (bottom-left) have average delays very close to 0 (white).

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)

© @marianorico

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.

© @marianorico

How to display networks

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

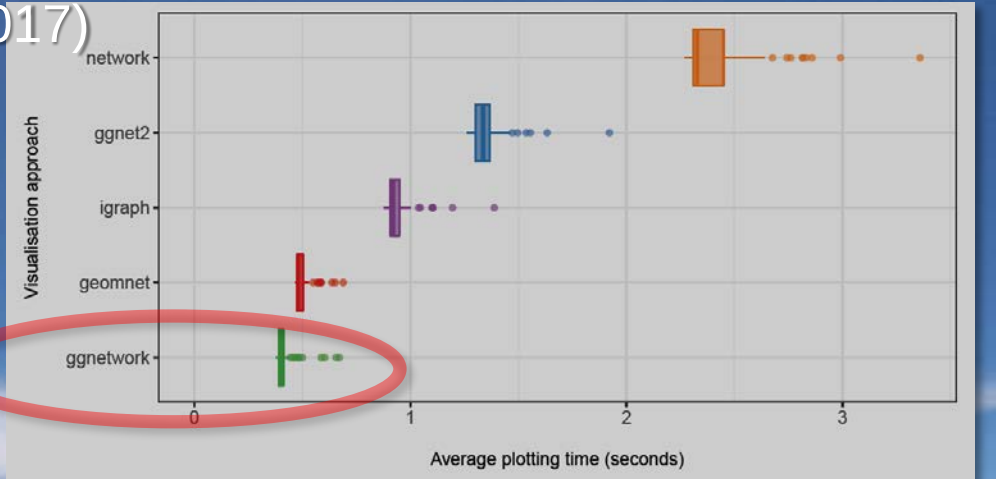
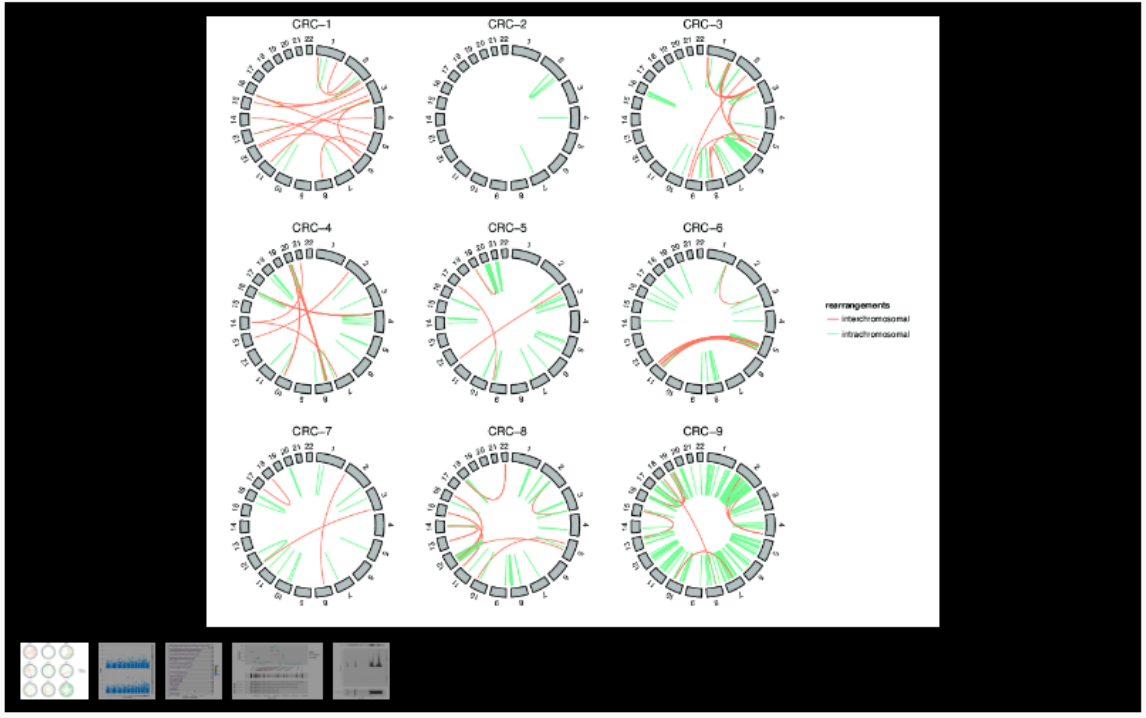


Figure 10: Comparison of the times needed for calculating and rendering the previously discussed protein interaction network in the three **ggplot2** approaches and the standard plotting routines of the **network** and **igraph** packages based on 100 evaluations each.

© @marianorico

More ggplot2 style

- ggbio

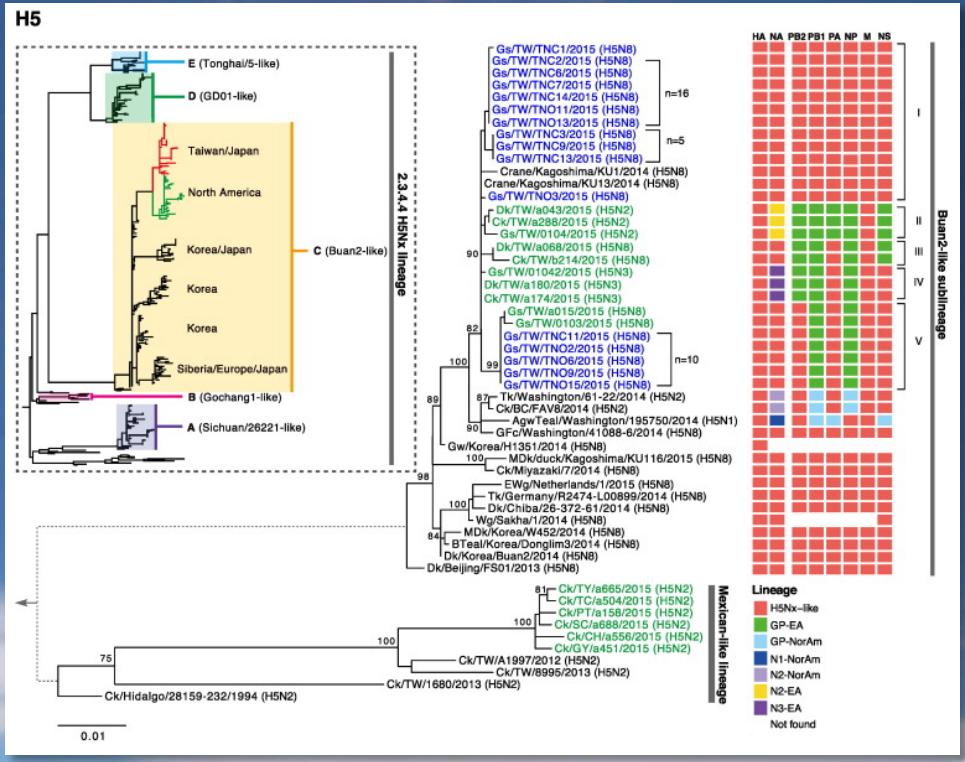


ggbio: An R implementation for extending the Grammar of Graphics for Genomic Data

© @marianorico

More ggplot2 style

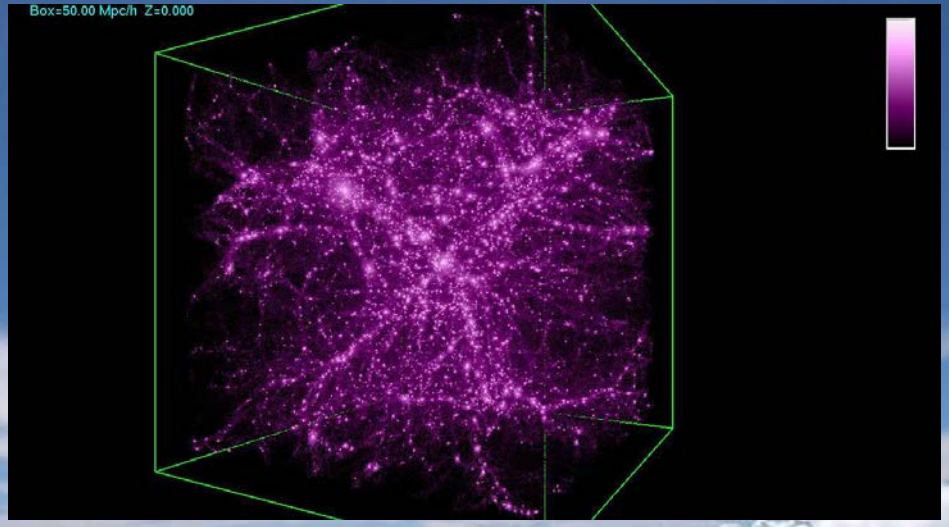
- ggtree



© @marianorico

Hardware limits

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



© @marianorico

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

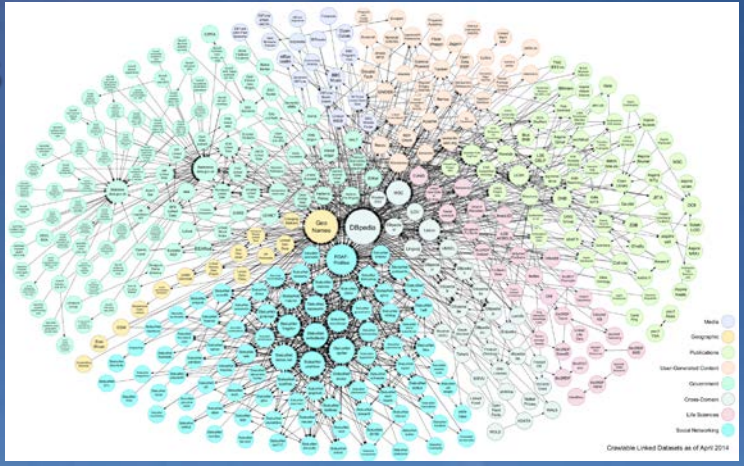
BIG DATA USE CASE: LOD CLOUD

In the LOD galaxy…

© @marianorico

LOD galaxy

- Numbers
  - Number of datasets: thousands
  - Number of triples: billions
- How do we explore it?
  - Aggregators:
    - LDcatalog (the “classical picture”)
      - 1014 datasets as of Aug. 2014 (last version)
    - LOD2
      - 9960 datasets as of April 2017



Big VISIBLE & INTERACTIVE graphs with

GEPHI

© @marianorico

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

© @marianorico

Gephi

Version 0.9.2 (Sep 2017)
goes fine on Windows 10 & iOS
Version 0.9.1 (Feb 2016)
has platform issues

- 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)

Java SE Development Kit 7 Downloads

End of Public Updates for Oracle JDK 7

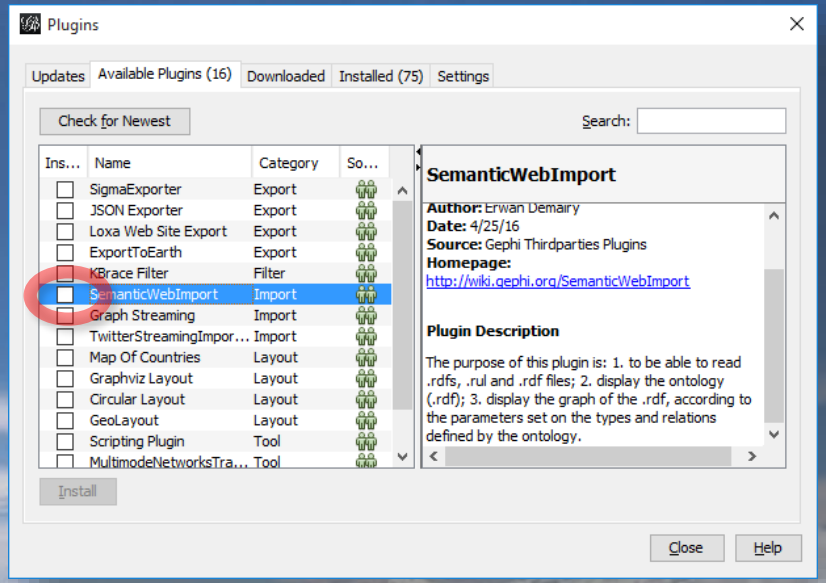
This release will be the last Oracle JDK 7 publicly available update. For more information, and details on how to receive longer term support for Oracle JDK 7, please see the Oracle Java SE Support Roadmap.

      - 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

© @marianorico

Gephi

- 0.92 version Installation on Windows
  - Can run without **un**installing previous versions
  - The WebImport plugin is listed ☺



© @marianorico

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 ☹

© @marianorico

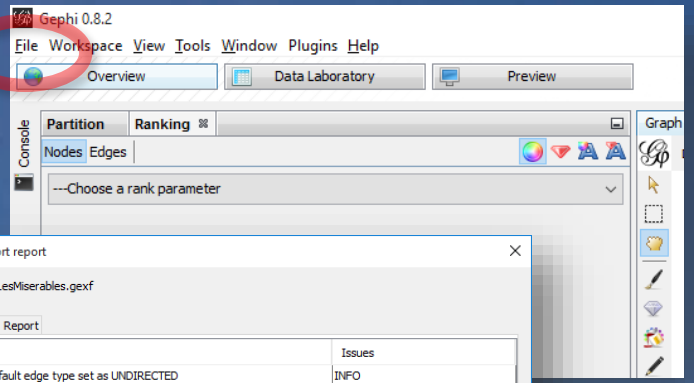
Hands on

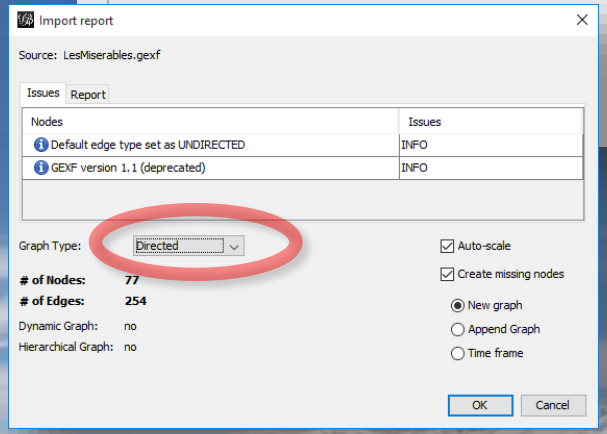
- Type data manually in Gephi's table
- See the resulting graph

© @marianorico

A not so quick tutorial

- Download a “gephi graph file”
  - LesMiserables.gexf (17 KB)
- Open Gephi
- Load LesMiserables.gexf
  - File → Open…
  - Change the graph type to “Directed”

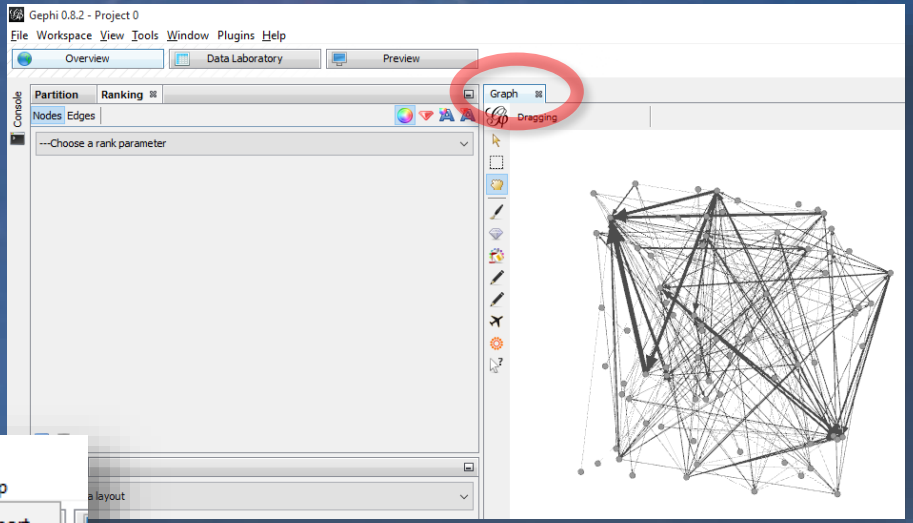




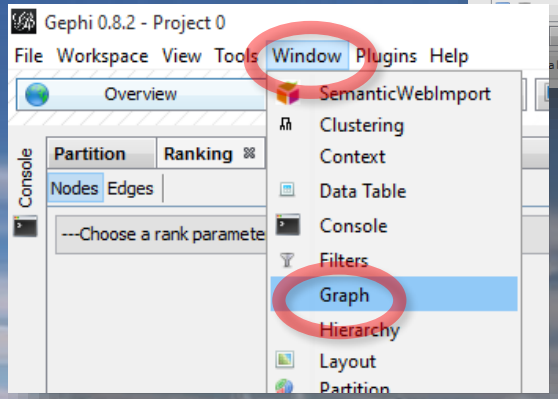
© @marianorico

Quick tutorial

- If you do not see this



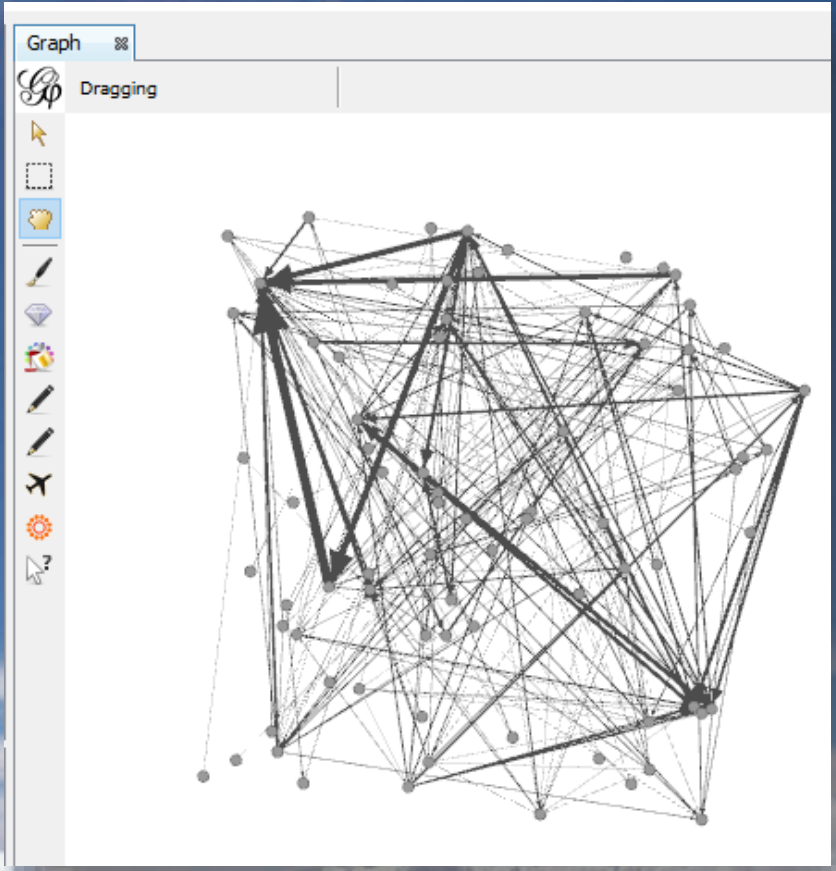
- Try this



© @marianorico

Quick tutorial

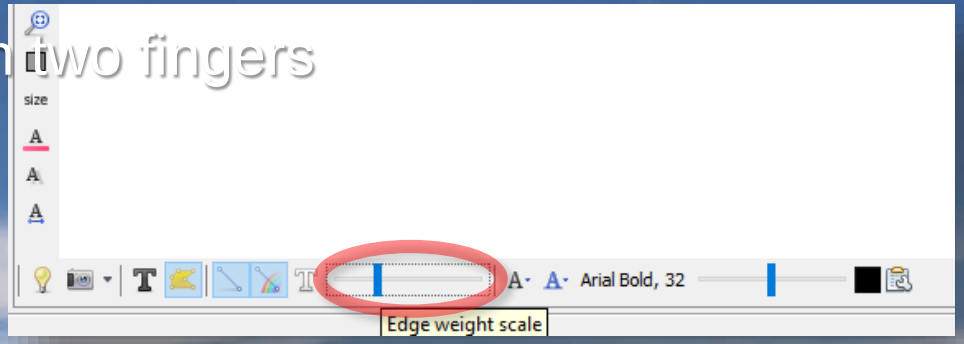
- Node positions are random. You will see a different graph
- Each node is a character of Victor Hugo's novel
- This graph is a "Co-appearance weighted network", that is,
  - Two vertices are adjacent if the corresponding characters encounter each other, in selected chapters of the novel



© @marianorico

Quick tutorial

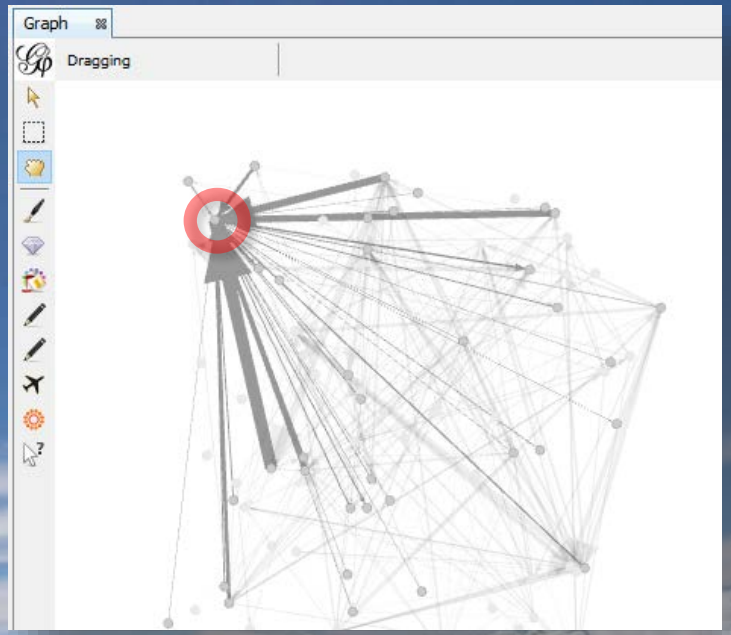
- Basics
  - Move the whole graph
    - Drag the mouse while **right** button is pressed
  - Zoom in/out
    - Use mouse wheel
    - or drag touchpad with two fingers
  - Width of edges (arcs)
    - Slider (no “text size” slider)



© @marianorico

Quick tutorial

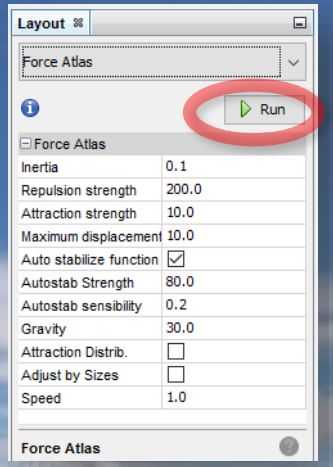
- Basics (cont.)
  - Hover (locate mouse pointer) over a node shows the links to/from that node
    - “Fading out” the rest of the graph

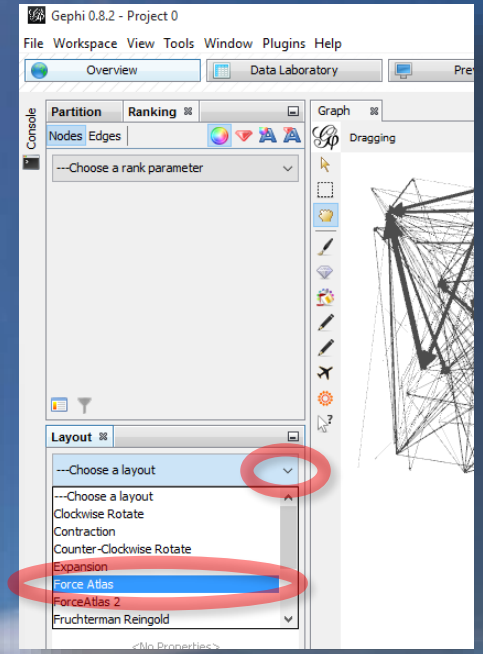


© @marianorico

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

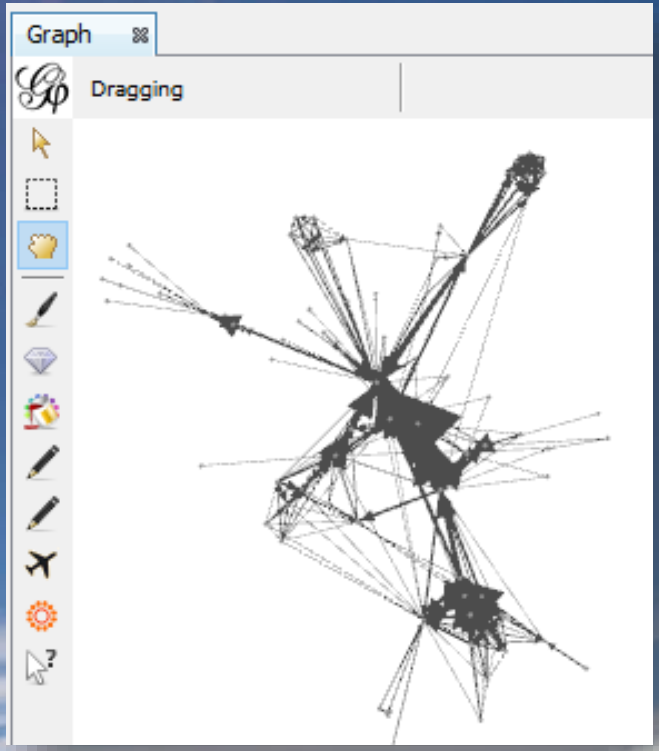




© @marianorico

Quick tutorial

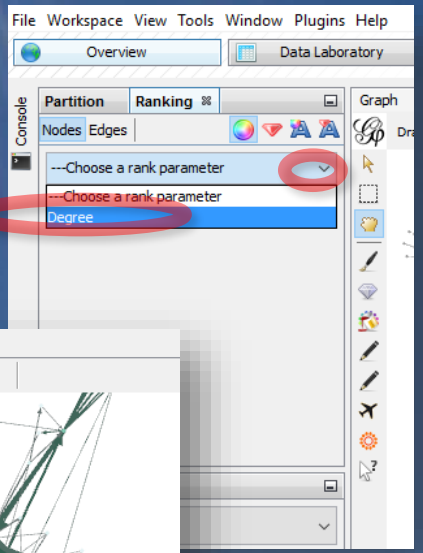
- Layout (cont.)
  - Press “Stop” button
  - Zoom in
  - Drag the graph
- You should get (something like) this

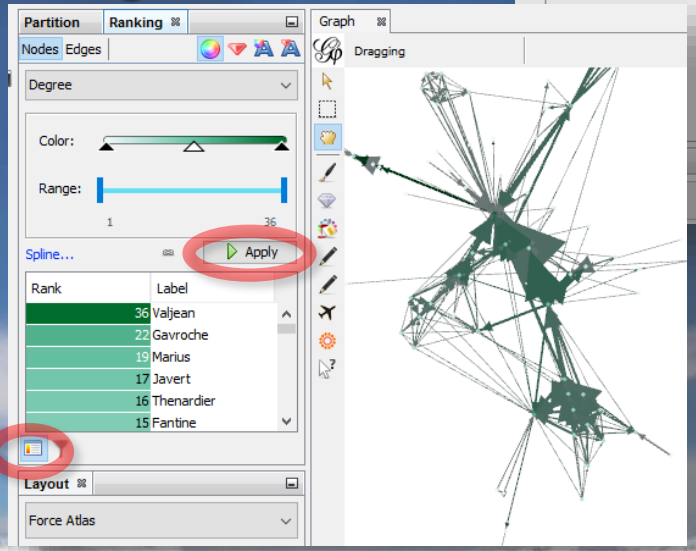


© @marianorico

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

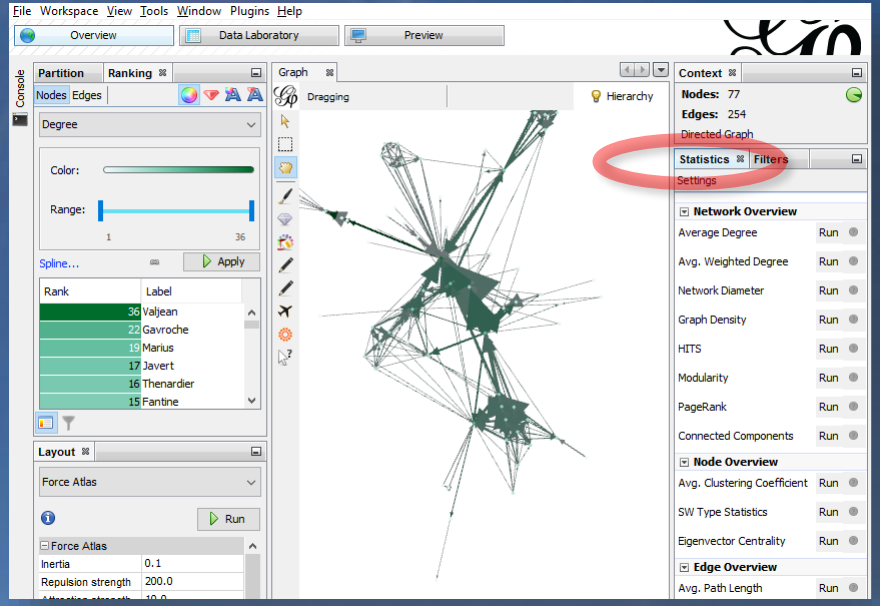




© @marianorico

Quick tutorial

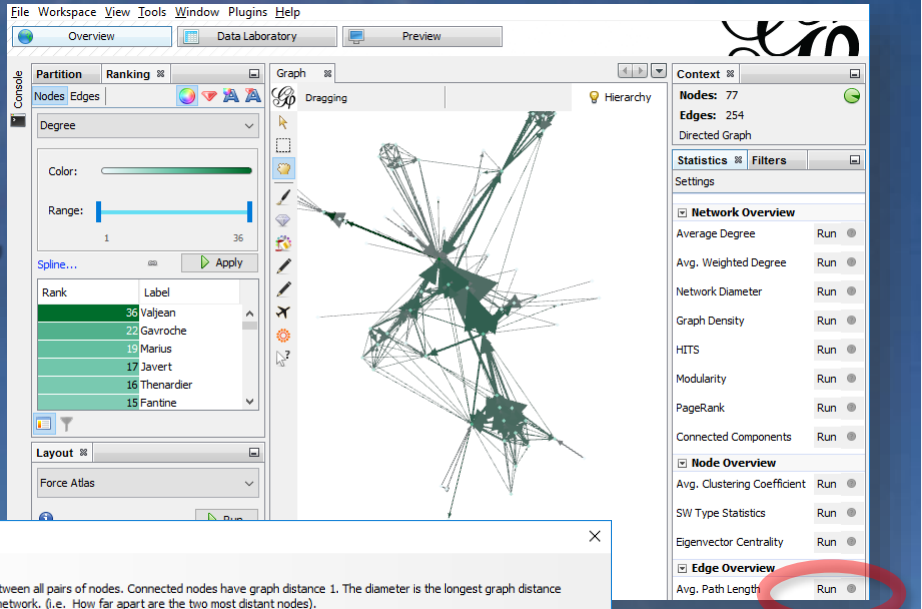
- Graph metrics
  - Many metrics (Statistics tab on the right side)
    - In 3 groups
      - Network overview
        - » Average Degree
        - » …
        - » Page Rank
      - Node overview
      - Edge overview

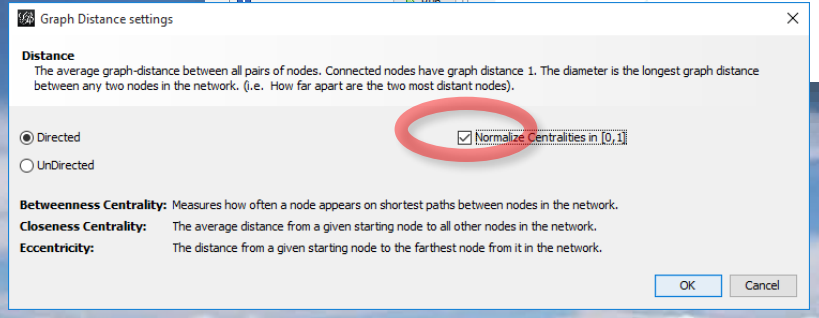


© @marianorico

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

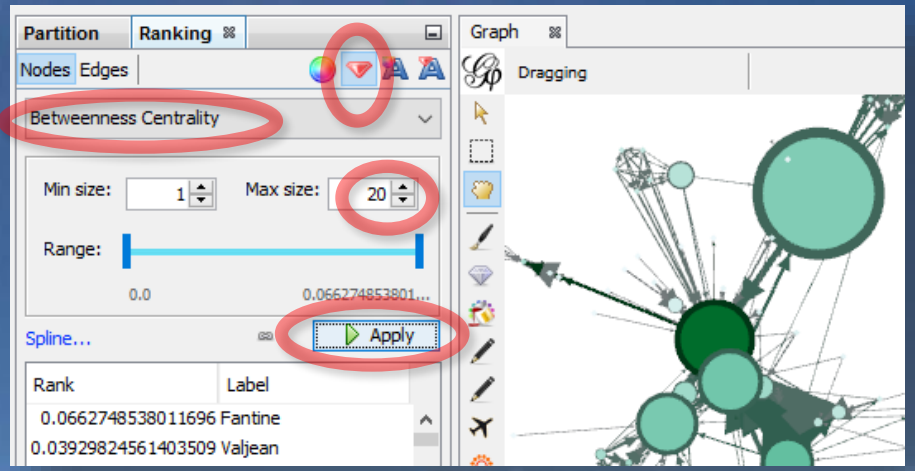




© @marianorico

Quick tutorial

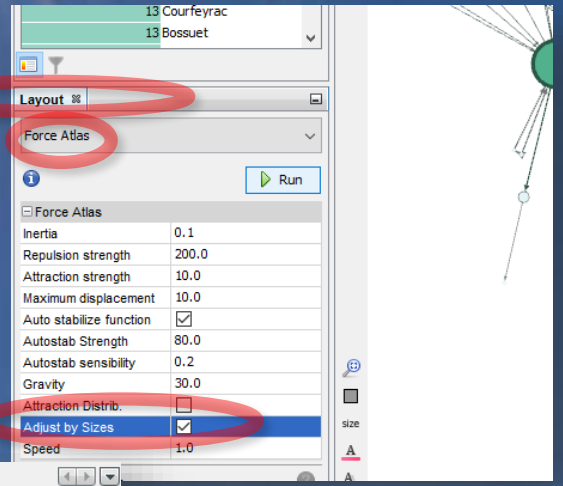
- Graph metrics (cont.)
  - Now you have more metrics in the "rank parameter"
  - Click node size icon
  - Select "Betweenness Centrality"
    - Higher values mean more influential (nodes); i.e. "bridges" between relevant nodes.
  - Apply (change "Max size") to apply this metric to nodes' size. Notice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

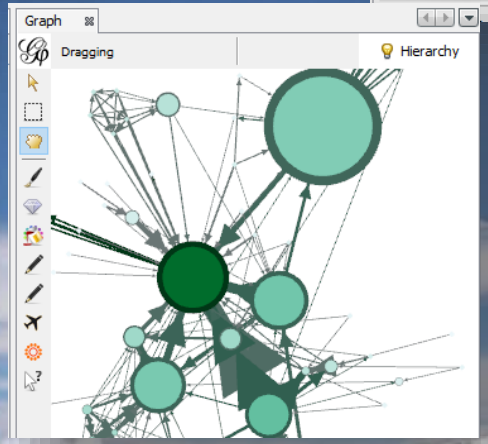


© @marianorico

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

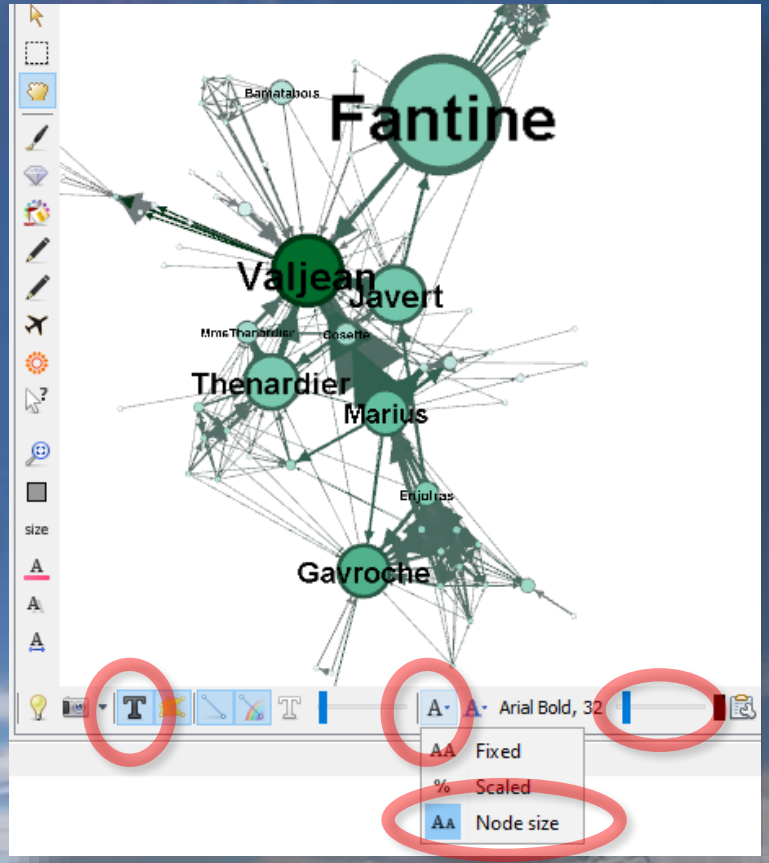




© @marianorico

Quick tutorial

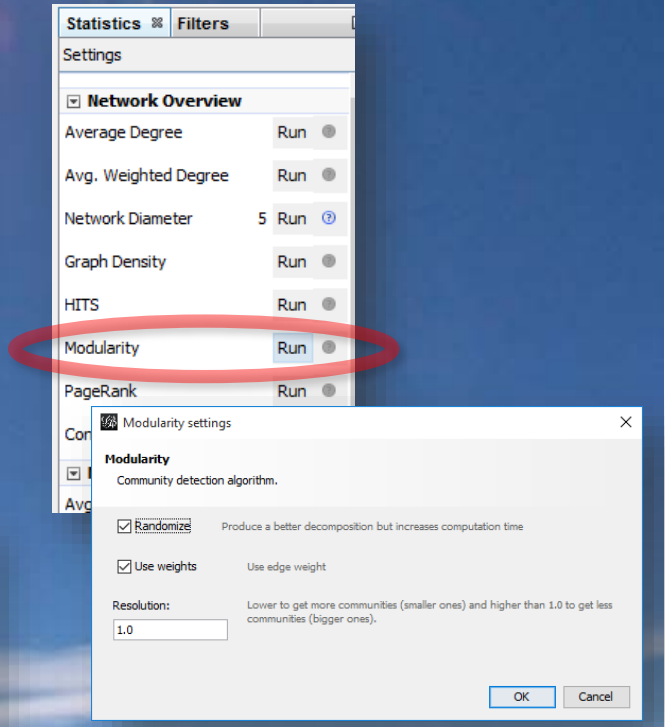
- Show Nodes' labels
  - Click on the icon
  - Select "Node size" in the "size mode" selector
  - Move the "text size" slider
- You should get (something like) this

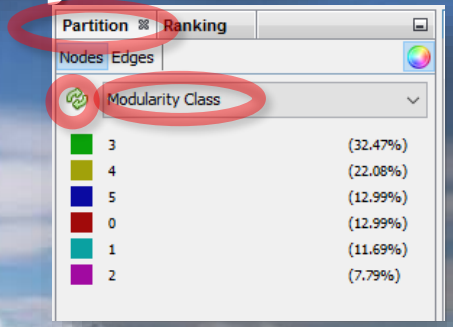


© @marianorico

Quick tutorial

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

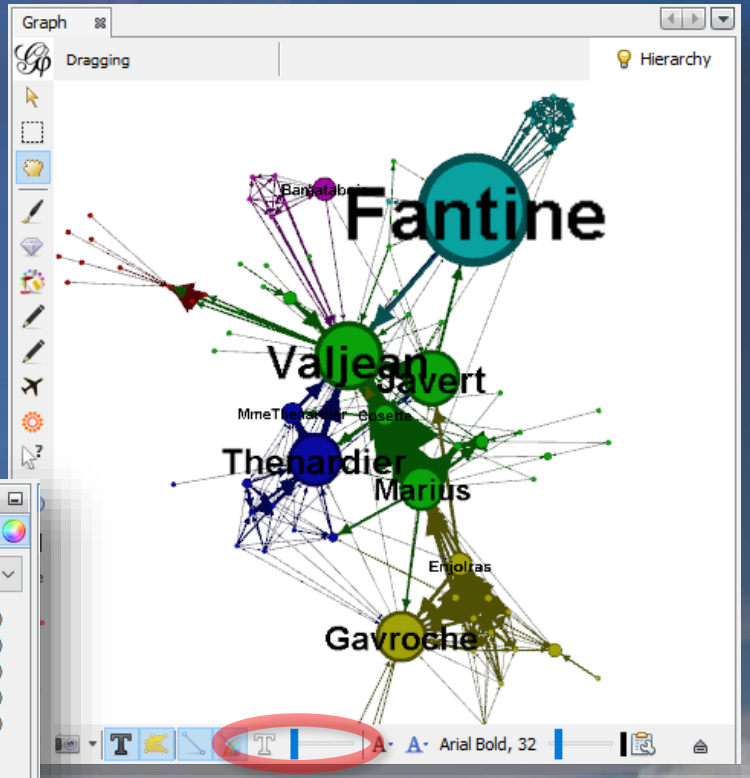


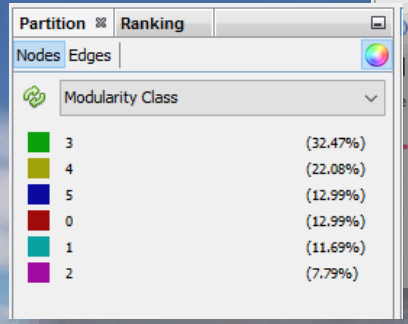


© @marianorico

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

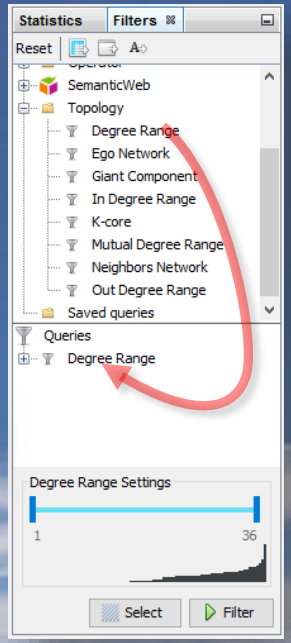


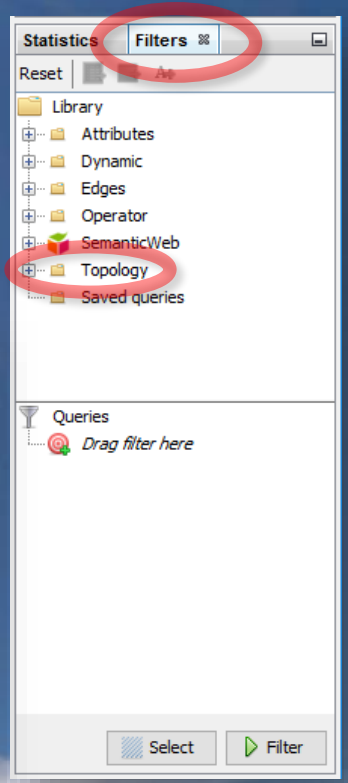


© @marianorico

Quick tutorial

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

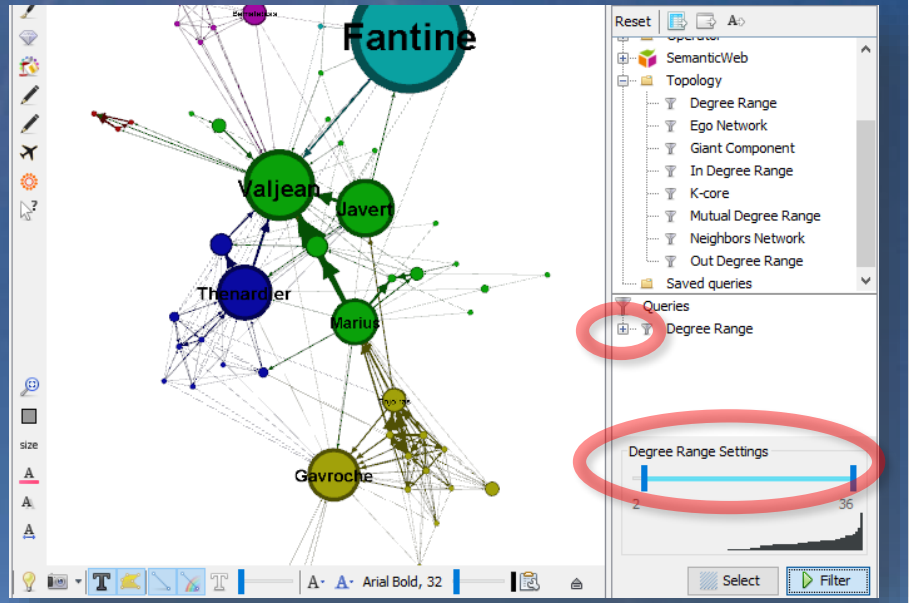


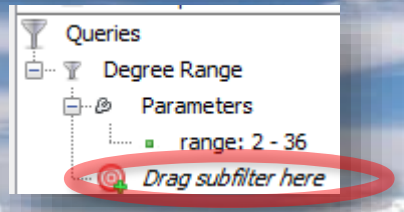


© @marianorico

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

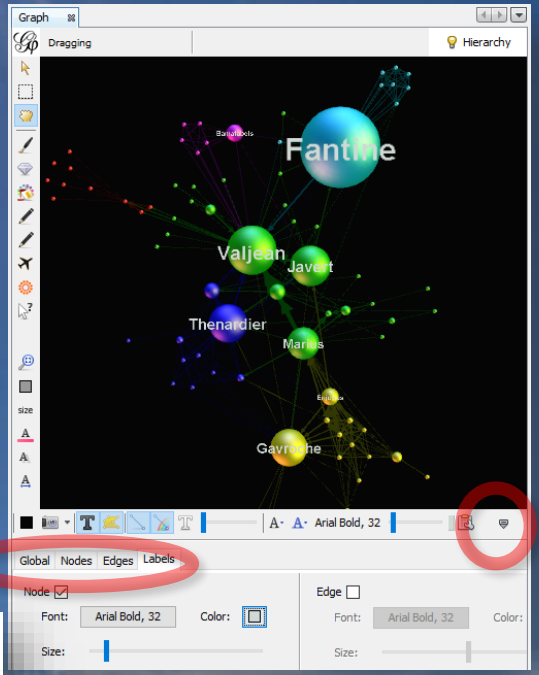


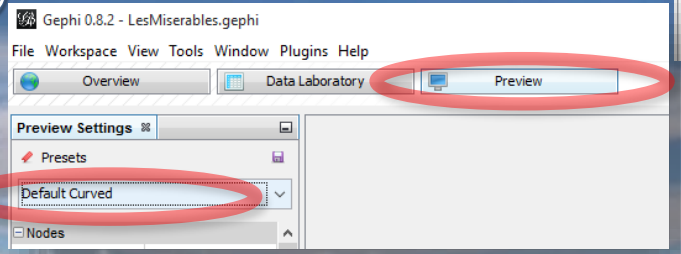


© @marianorico

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

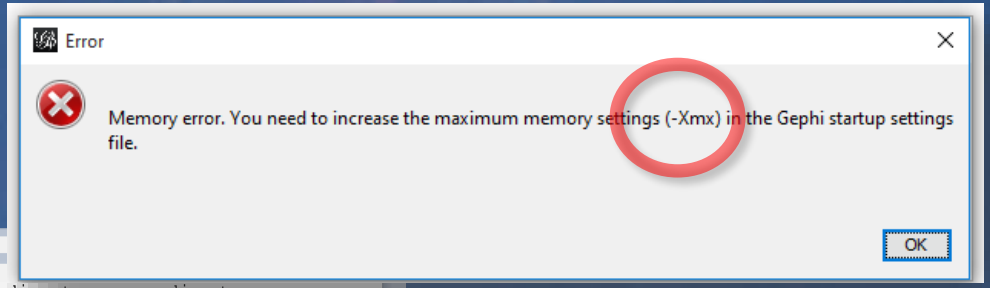




© @marianorico

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.noddraw=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=
```

Big Linked Data graphs with

GEPHI LAYOUTS

© @marianorico

Gephi layouts

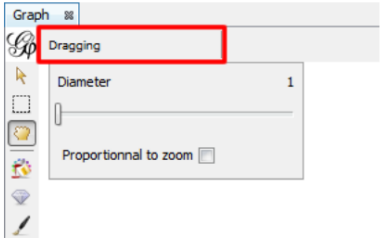
- See documentation at https://gephi.org/tutorials/gephi-tutorial-layouts.pdf
- Here I will show you a summary

© @marianorico

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.
- Ajust 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.

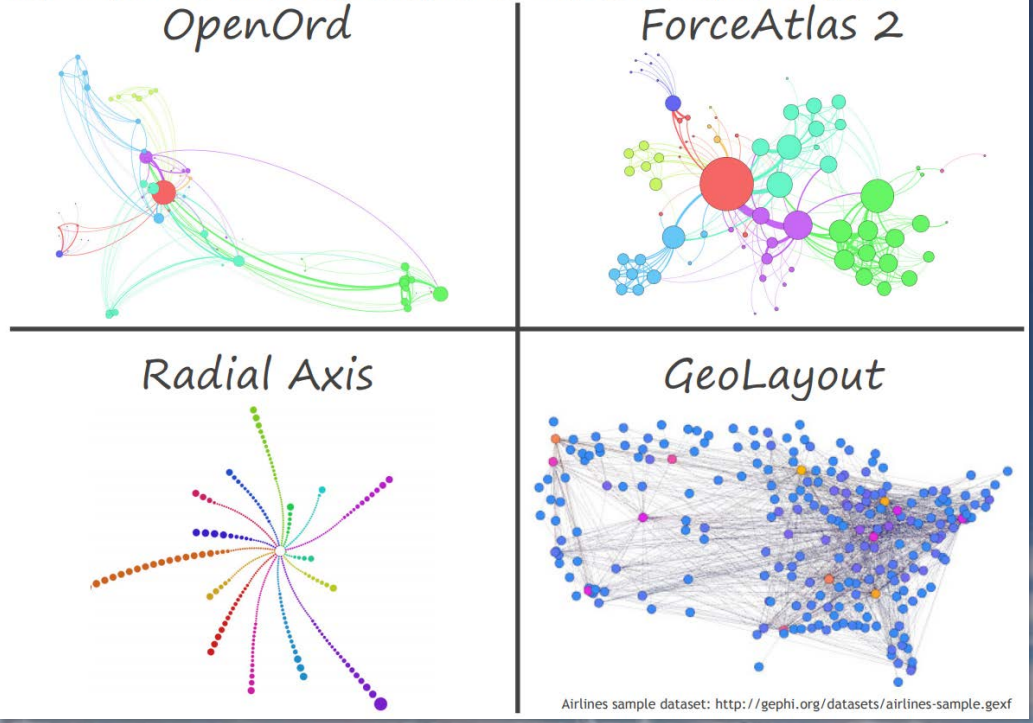
Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

© @marianorico

Gephi layouts

Various layouts exist

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



Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

© @marianorico

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*

Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

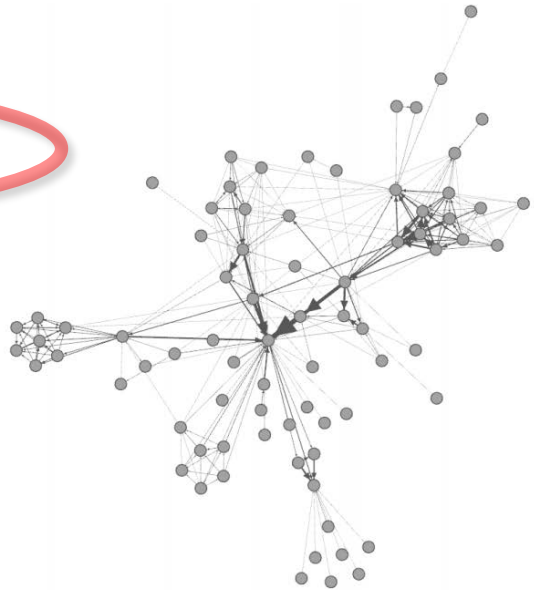
© @marianorico

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



Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

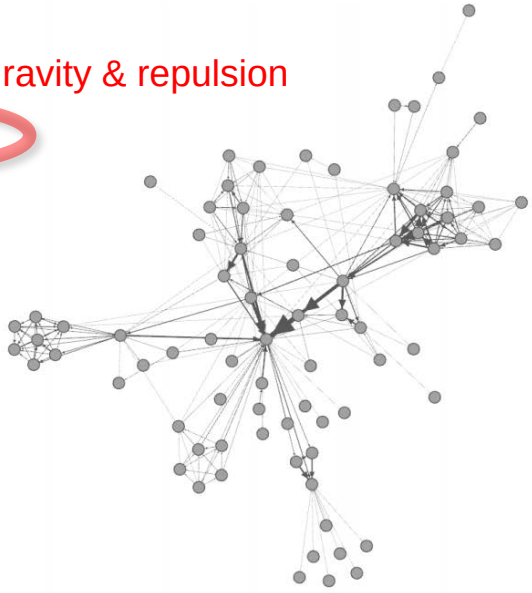
© @marianorico

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.

Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

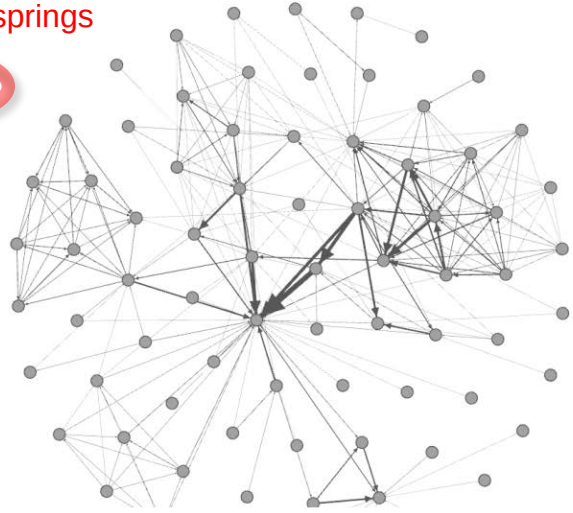
© @marianorico

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:	Thomas Fruchterman & Edward Reingold[1]
Date:	1991
Kind:	Force-directed: springs
Complexity:	$O(N^2)$
Graph size:	1 to 1 000 nodes
Use edge weight:	No



Run Fruchterman-Reingold

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

• Area = 100 • Area = 100 000	Graph size area.
• Gravity = 1 000 • Gravity = 100	Attract all nodes to the center to avoid dispersion of disconnected components.

And now Stop the algorithm.

Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

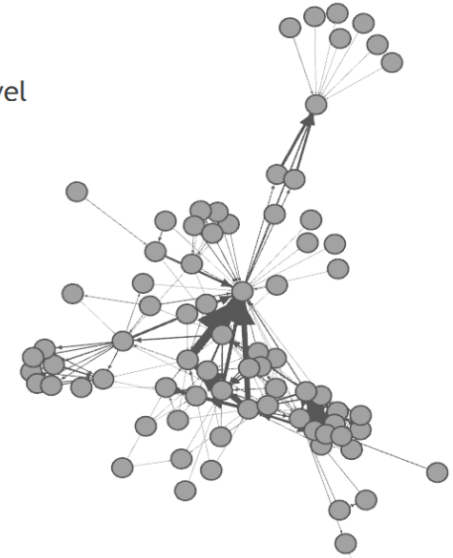
© @marianorico

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[1]
Date:	2005
Kind:	Force-directed + multilevel
Complexity:	O(N*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.

Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

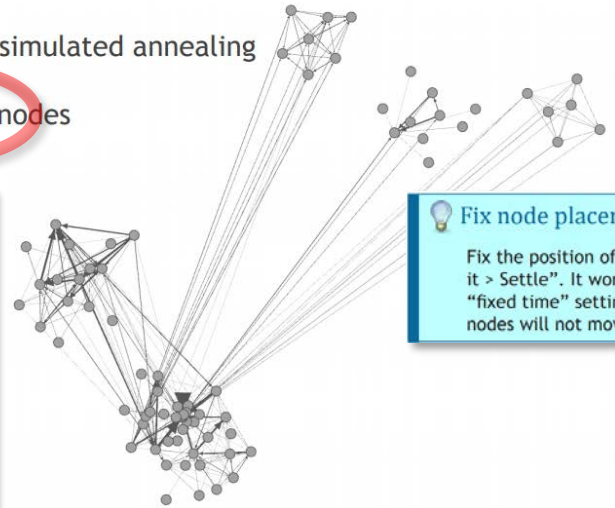
© @marianorico

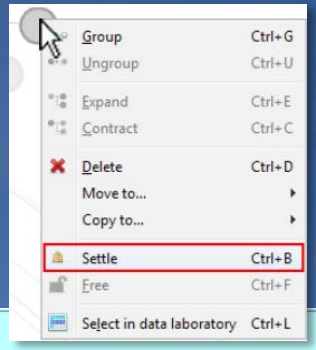
Gephi layouts

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 Frutcherman-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[1]
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





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.

Run OpenOrd

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

- Edge cut = 0.95 — Run — From 0 (standard Frutcherman-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 — Run
- Num iterations = 850 — Run — Contract the clusters. Expand the clusters.
- Random seed = -6308261588084905834 — Run — Use this value to produce exactly the same shape as shown before.

Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

© @marianorico

Gephi layouts

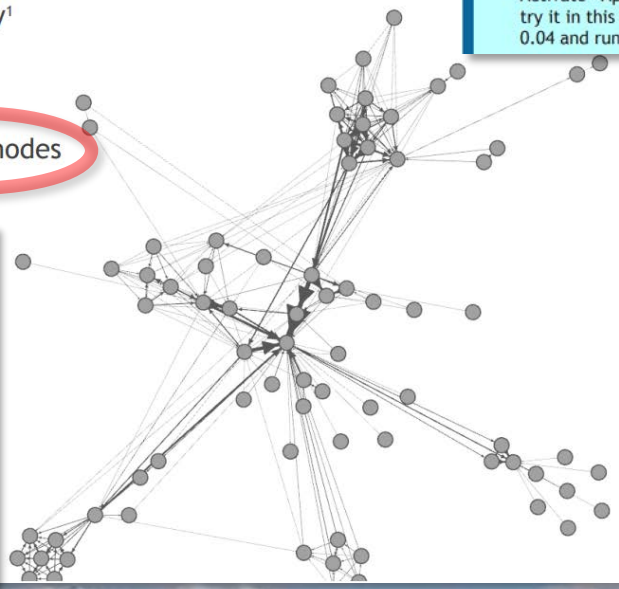
ForceAtlas 2 layout

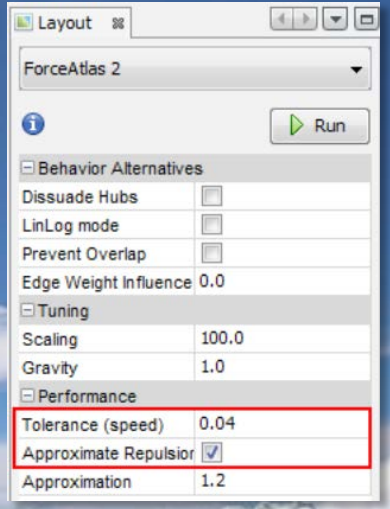
Improved version of the Force Atlas to handle large networks while keeping a very good quality. Nodes repulsion is approximated with a Barnes-Hut calculation, which therefore reduces the algorithm complexity. Replace the “attraction” and “repulsion” forces by a “scaling” parameter.

Author:	Mathieu Jacomy[1]
Date:	2011
Kind:	Force-directed
Complexity:	O(N*log(N))
Graph size:	1 to 1 000 000 nodes
Use edge weight:	Yes

Performance

Activate “Approximate Repulsion” on large graphs only, but let’s try it in this tutorial. Check it, set the “Tolerance” option to 0.04 and run the algorithm to see how nodes are swinging!





Run ForceAtlas 2

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

• LinLog mode = checked • LinLog mode = unchecked	Linear attraction & logarithmic repulsion (lin-lin by default), makes clusters tighter.
• Scaling = 100	Increase to make the graph sparser.
• Edge weight influence = 0	From 0 (no influence) to 1 (normal). Set 0 to calculate forces without edge weight.

And now Stop the algorithm.

Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

© @marianorico

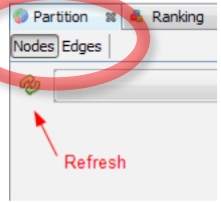
Gephi layouts

Detect communities

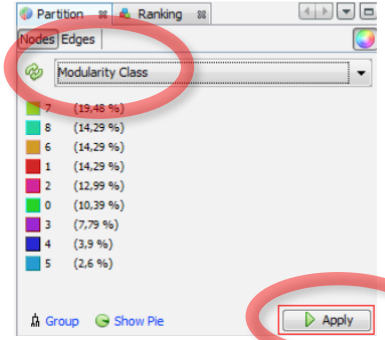
We now want to study the community structure in this network: does it divide naturally into groups of nodes with dense connections within groups and sparser connections between groups?

In the Σ Statistics panel, click on Run near the "Modularity"[1] 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 Apply to colorize nodes.

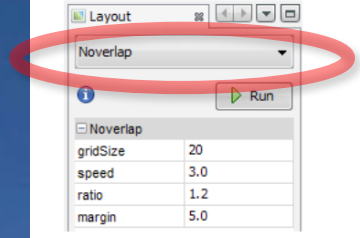
Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

© @marianorico

Gephi layouts

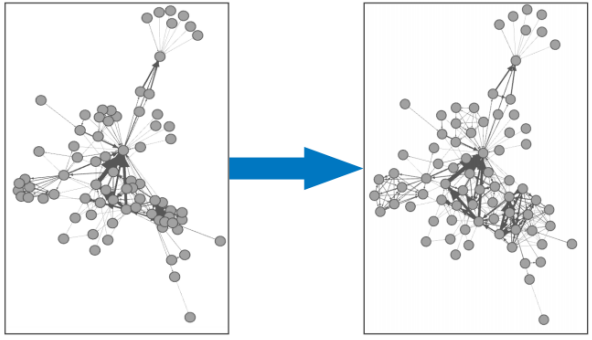
Noverlap layout

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 “Noverlap” 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.



Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

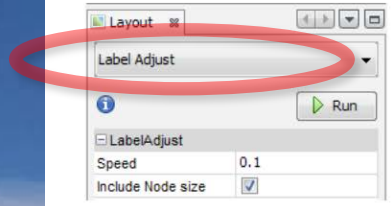
© @marianorico

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 T 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.

Source: https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

Big Linked Data graphs with

GEPHI + DBPEDIA

© @marianorico

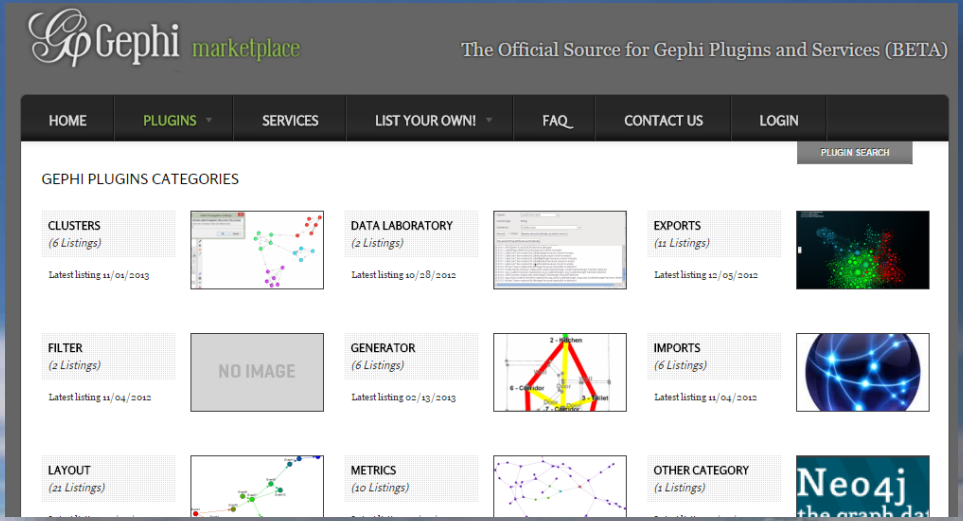
Gephi

- Plugins mechanish
  - SemanticWebImport (for Gephi 0.8.2)
  - ~~Virtuoso Importer (for Gephi 0.8.2)~~
  - Gephi Graph Streaming
  - Neo4J Graph Database Support

© @marianorico

Gephi

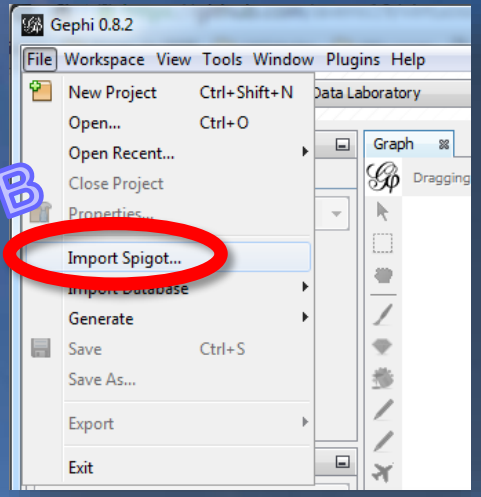
- Plugins
  - Use the Gephi marketplace



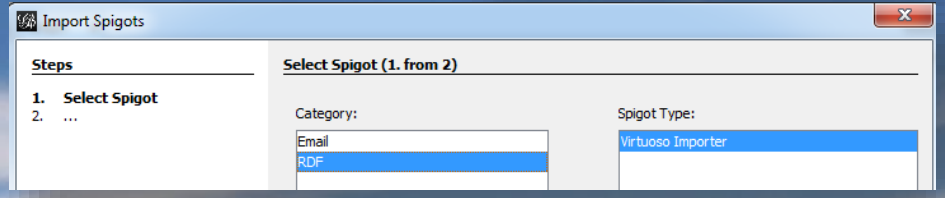
© @marianorico

Gephi

- Virtuoso Importer plugin
  - In the marketplace/imports
  - Download it as a nbm file
  - Install it in Gephi
    - Tools → Plugins → Downloaded → Add Plugins
  - Use it from File → Import Spigot…

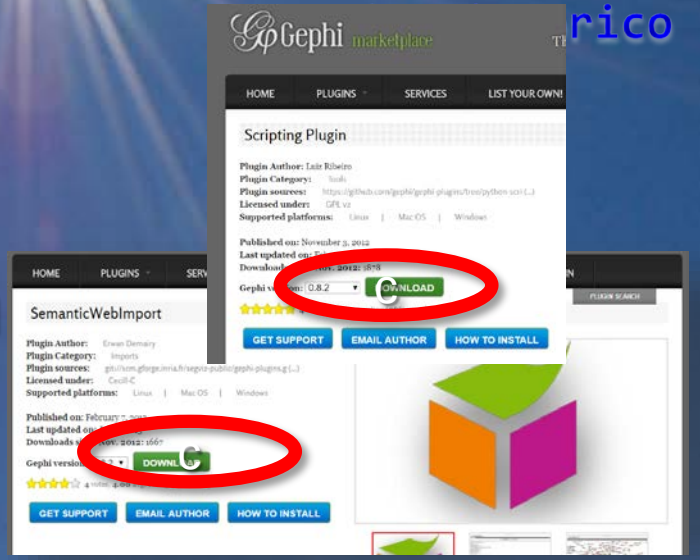


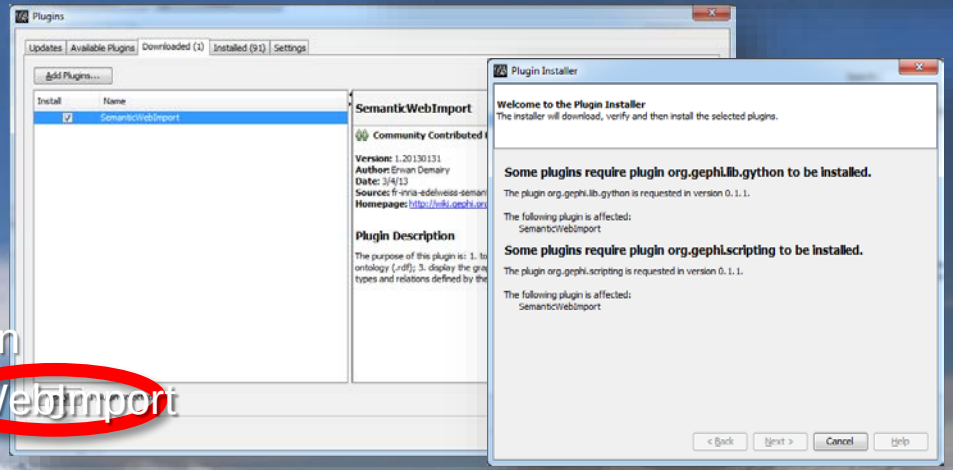
Only for Virtuoso as a traditional DB
No support since 2013



Gephi

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

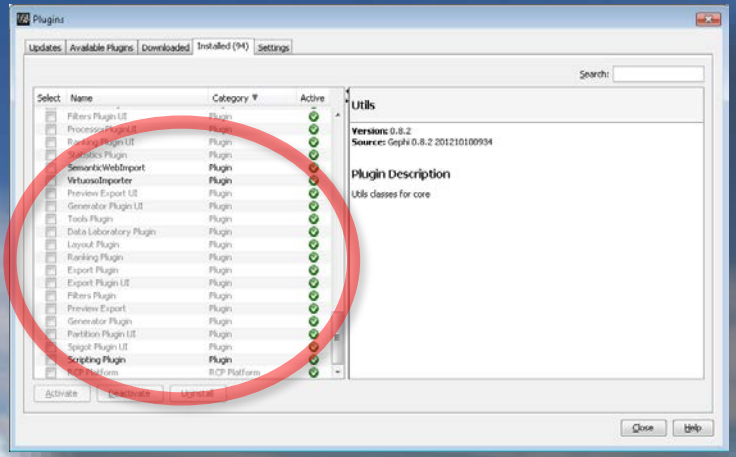


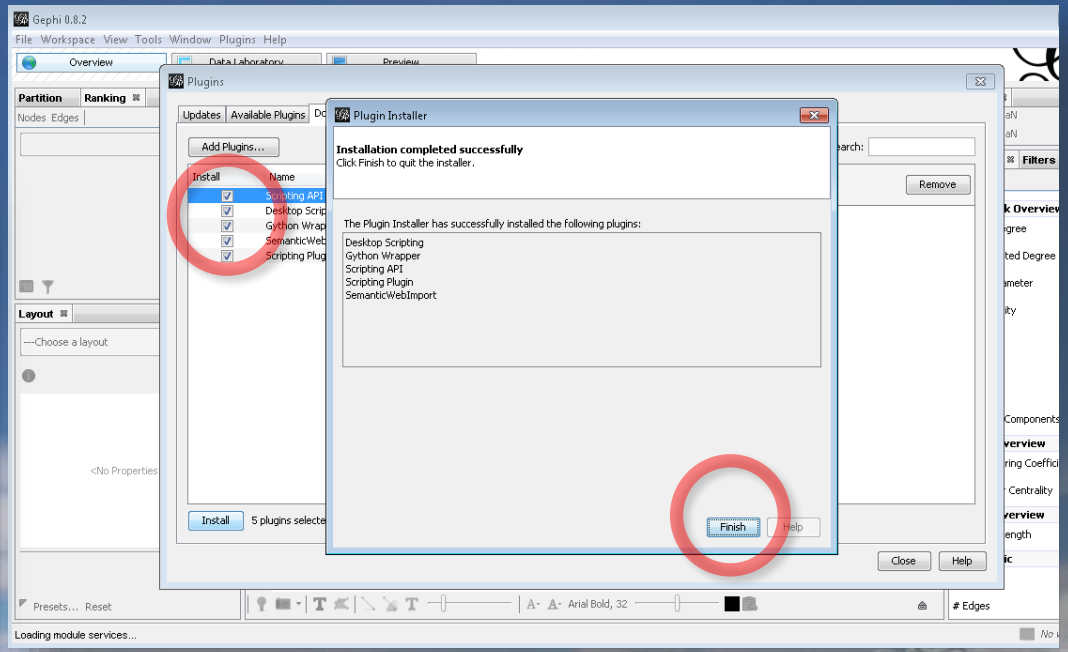


© @marianorico

Gephi

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



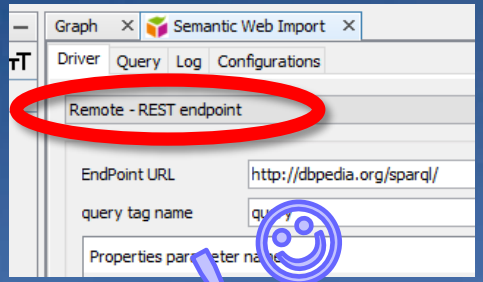


© @marianorico

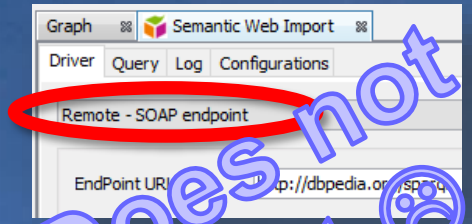
Gephi 0.9.2

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

Pay attention to the ending /

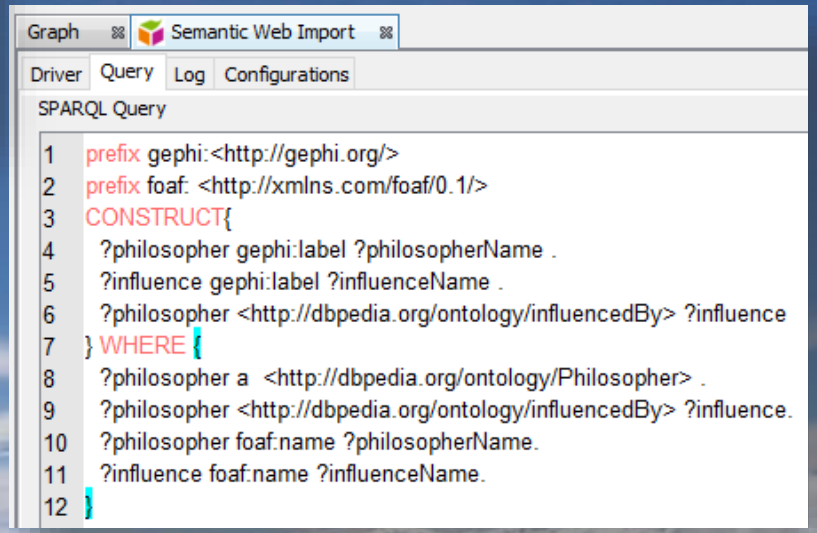


Yes! ☺



Does not exist ☹

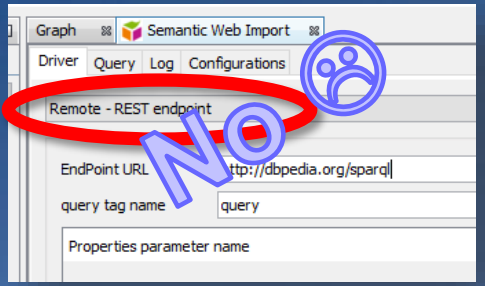
  - 2) Make a SPARQL **CONSTRUCT** query

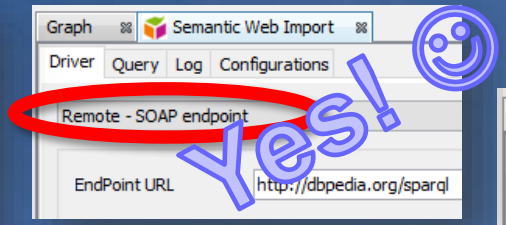


© @marianorico

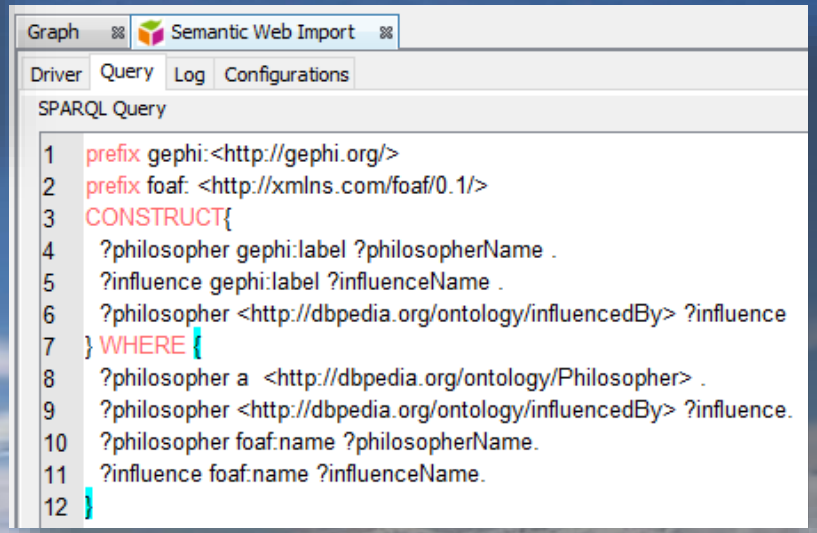
Gephi 0.8.2

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





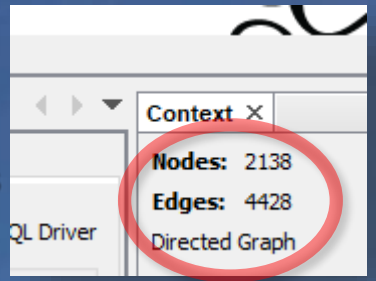
  - 2) Make a SPARQL **CONSTRUCT** query

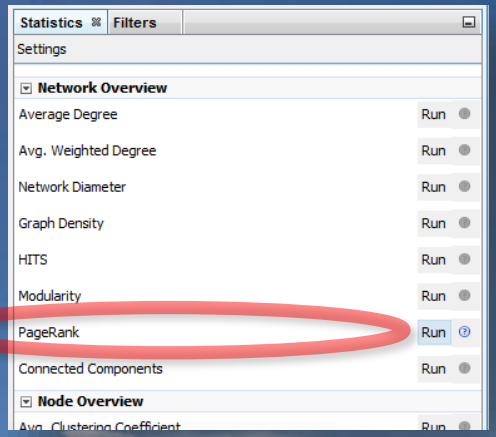


© @marianorico

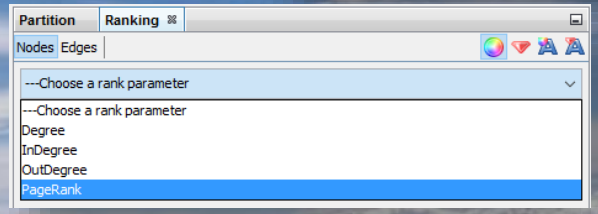
Gephi

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



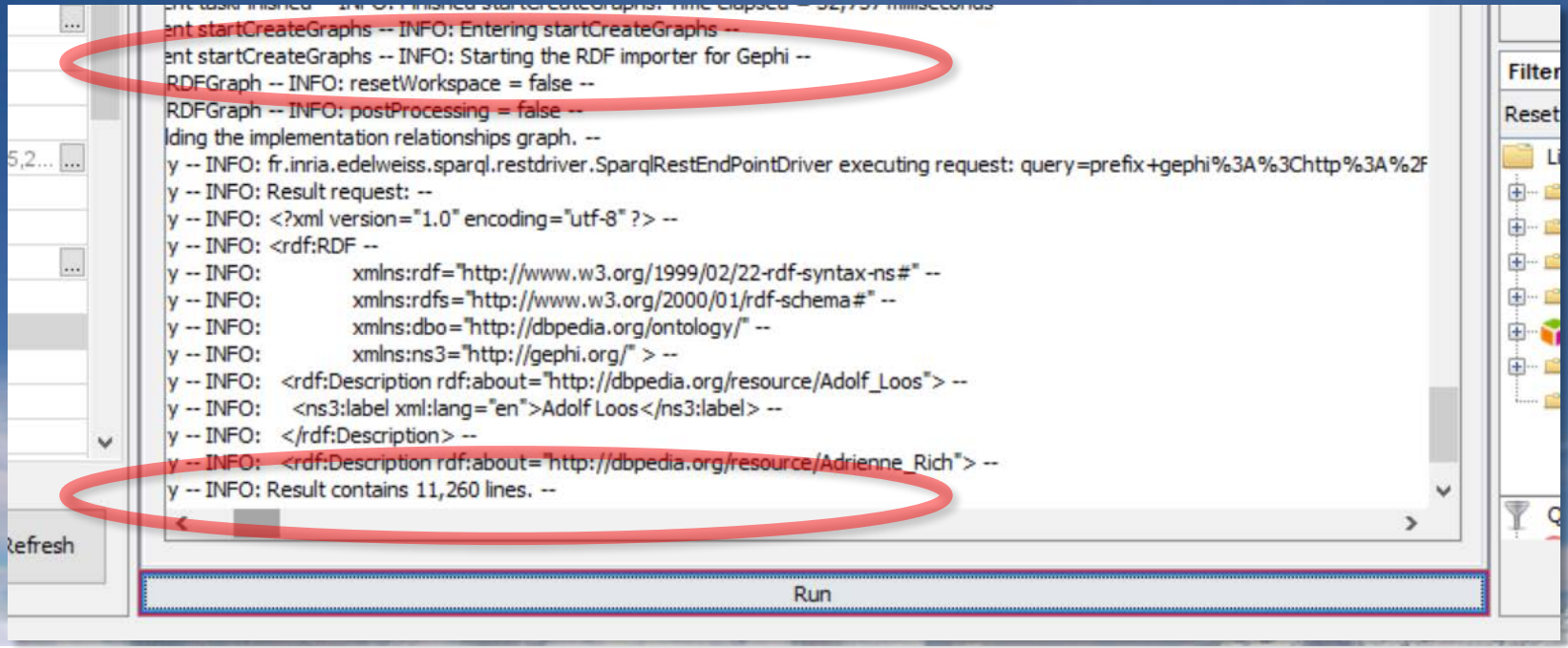


- Select Appearance → Nodes →Ranking
  Rank parameter = PageRank → Apply



Gephi

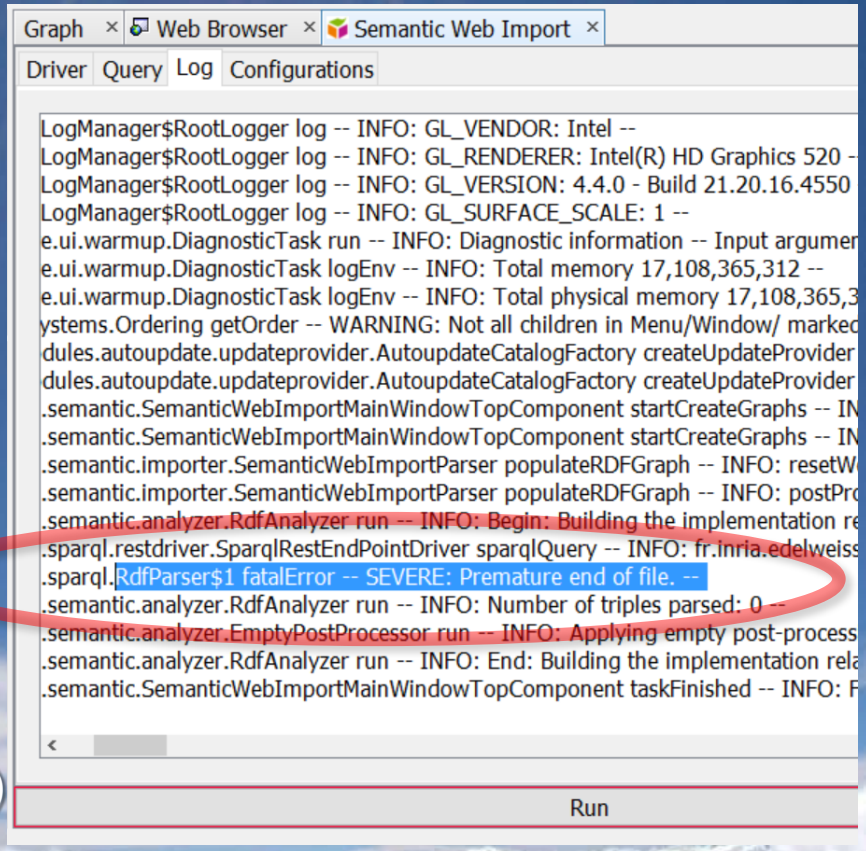
- You should get something like this in the “log” tab



© @marianorico

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

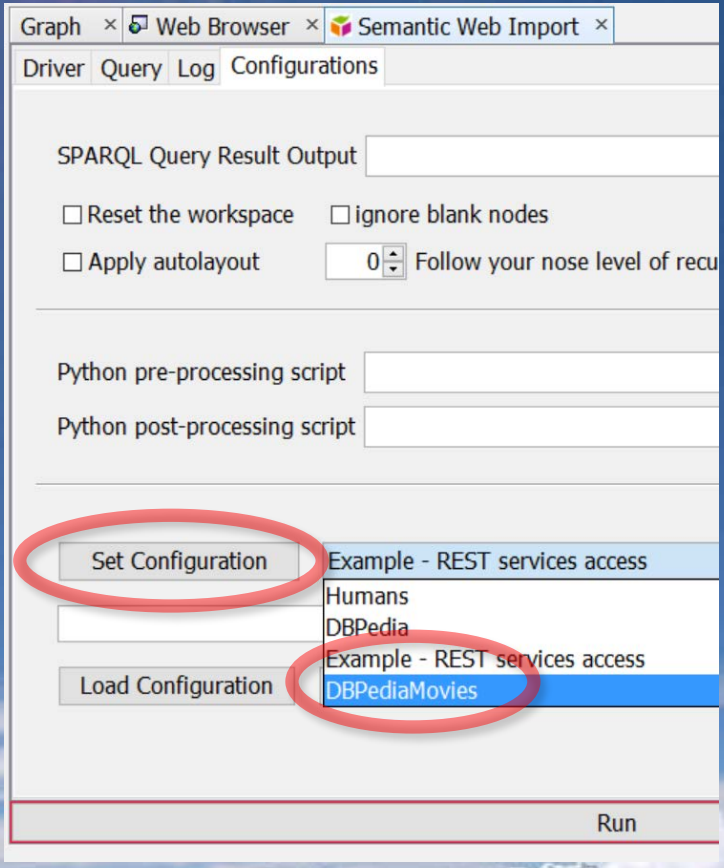
Look for invalid chars in the query (e.g. tabs)

May be you have to update the plugins

© @marianorico

Gephi

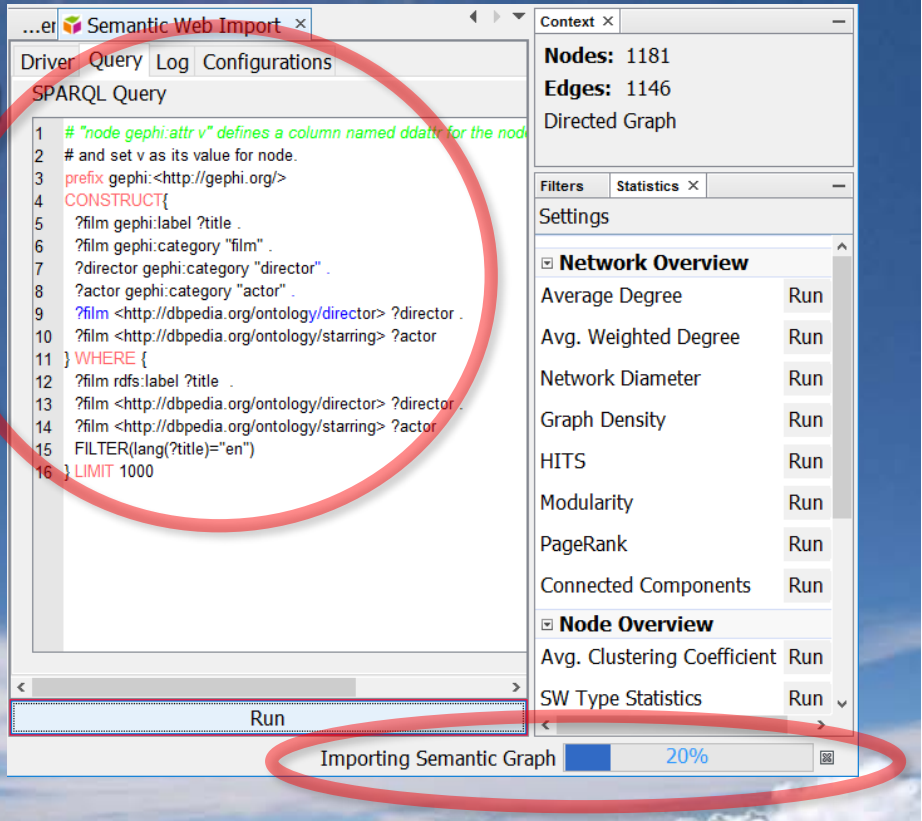
- If you cannot make it run
  - Use a “pre configuration”
    - Select DBPediaMovies
    - Press button “Set configuration” (you can see the CONSTRUCT query in the “Query” tab)



© @marianorico

Gephi

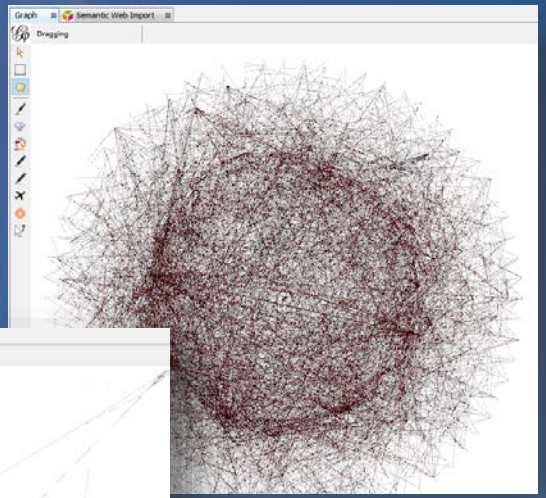
- If you cannot make it run
  - Use a “pre configuration”
    - Select DBPediaMovies
    - Press button “Set configuration” (you can see the CONSTRUCT query in the “Query” tab)



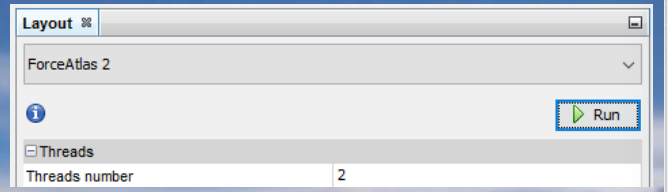
© @marianorico

Gephi

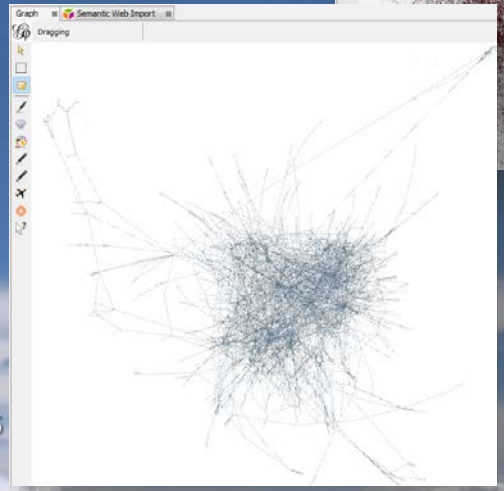
- Importing data from DBpedia
  - Check that you have something like this (Show/Hide the Graph tab with Window → Graph)



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

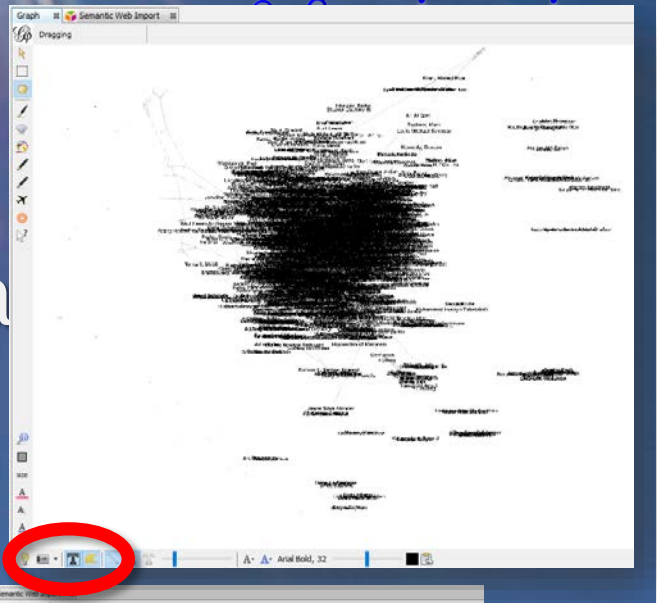


  - You should get some thing like this

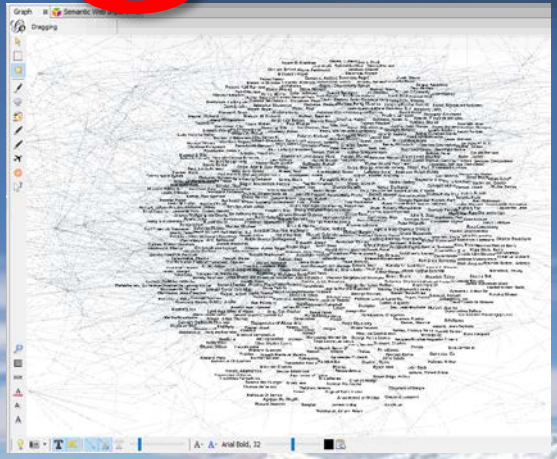


Gephi

- Importing data from DBpedia
  - Show node labels
    (Clic icon in the bottom of the Graph tab)



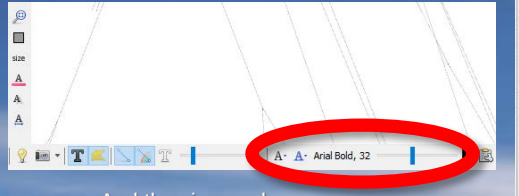
  - Layout → Expansion → Run three times
  - Layout → Label Adjust → Run ~10 seconds
  - You should get some thing like this



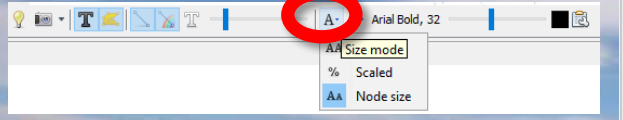
© @marianorico

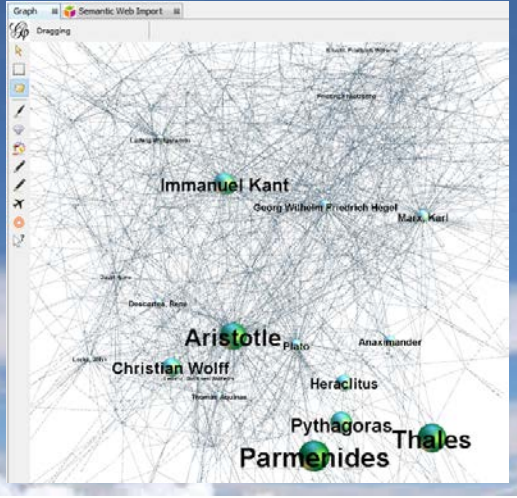
Gephi

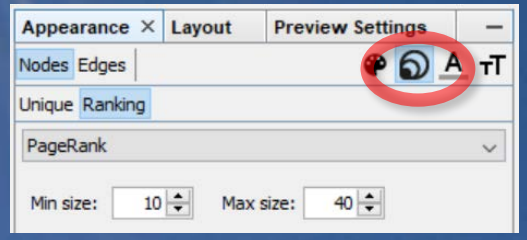
- Importing data from DBpedia
  - Node size proportional to ranking (Ranking→Nodes→ clic icon → set min & max →Apply)
  - Node label size proportional to ranking (Ranking→Nodes→clic icon → set min & max →Apply)
  - You should get some thing like this
    - Play with the font size slider

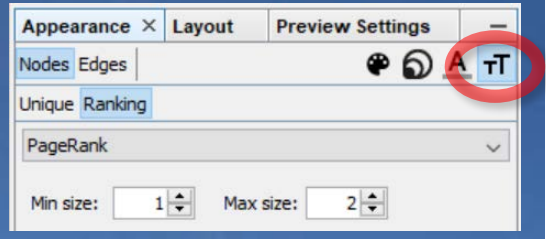


    - And the size mode





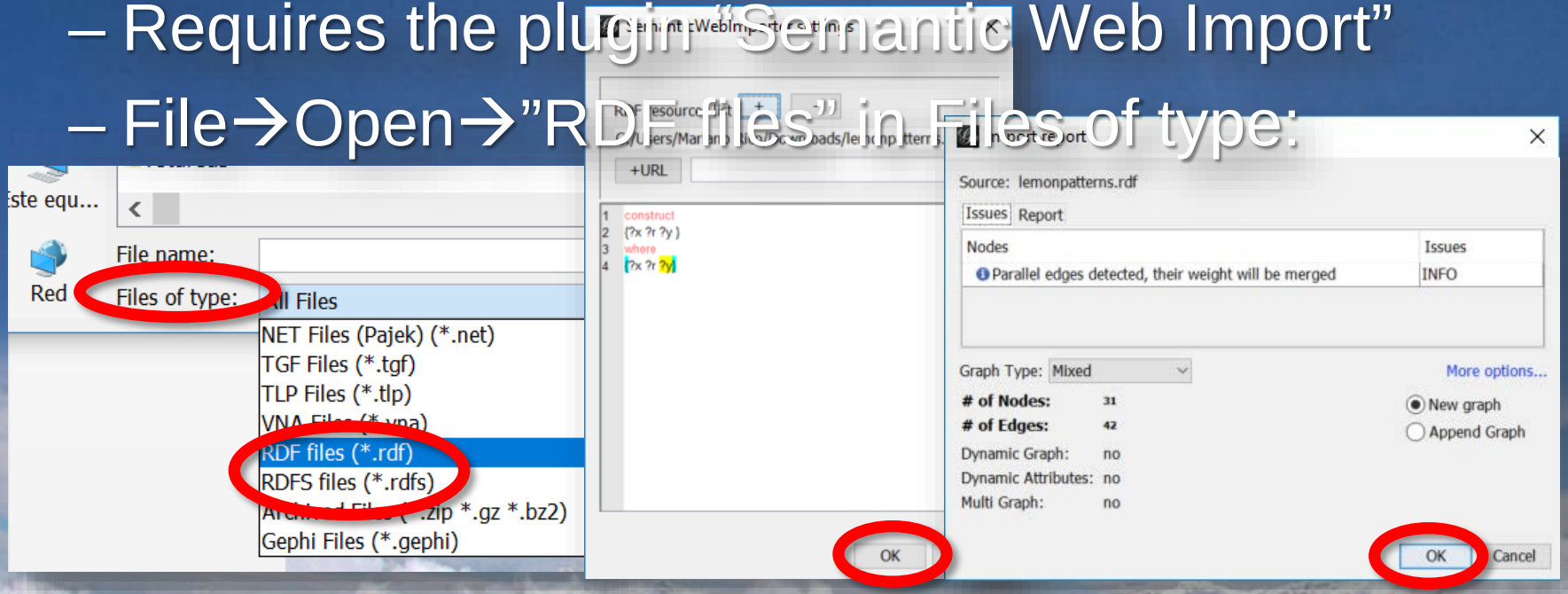




© @marianorico

Gephi

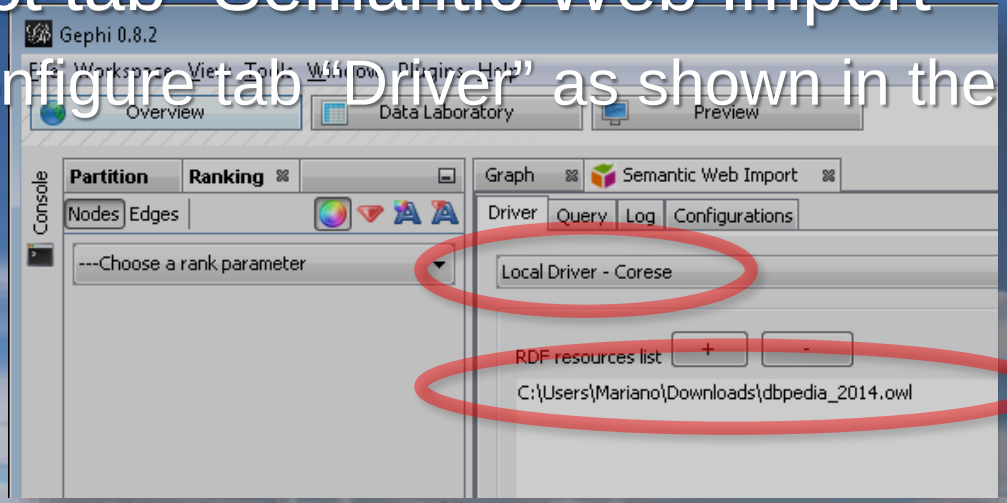
- Gephi can import RDF (not .owl, see later)
  - Requires the plugin "Semantic Web Import"
  - File→Open→"RDF files" in Files of type:



© @marianorico

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



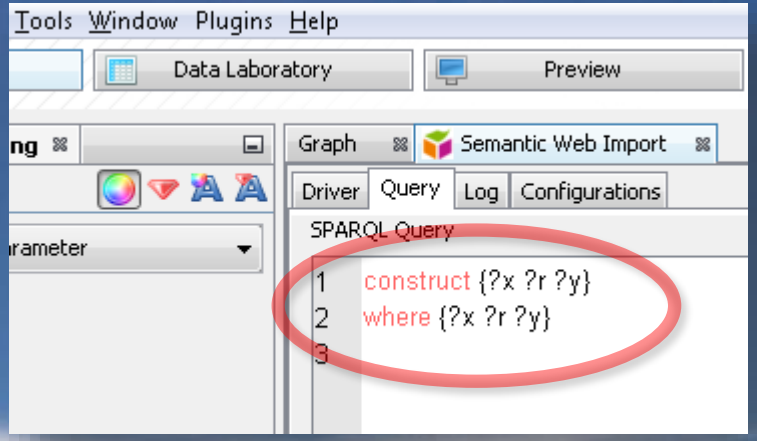
© @marianorico

Gephi

- Load local file (dbpedia_2014.owl)

  2) Click tab “Query”

  - Remove the triples limit (100 by default)



© @marianorico

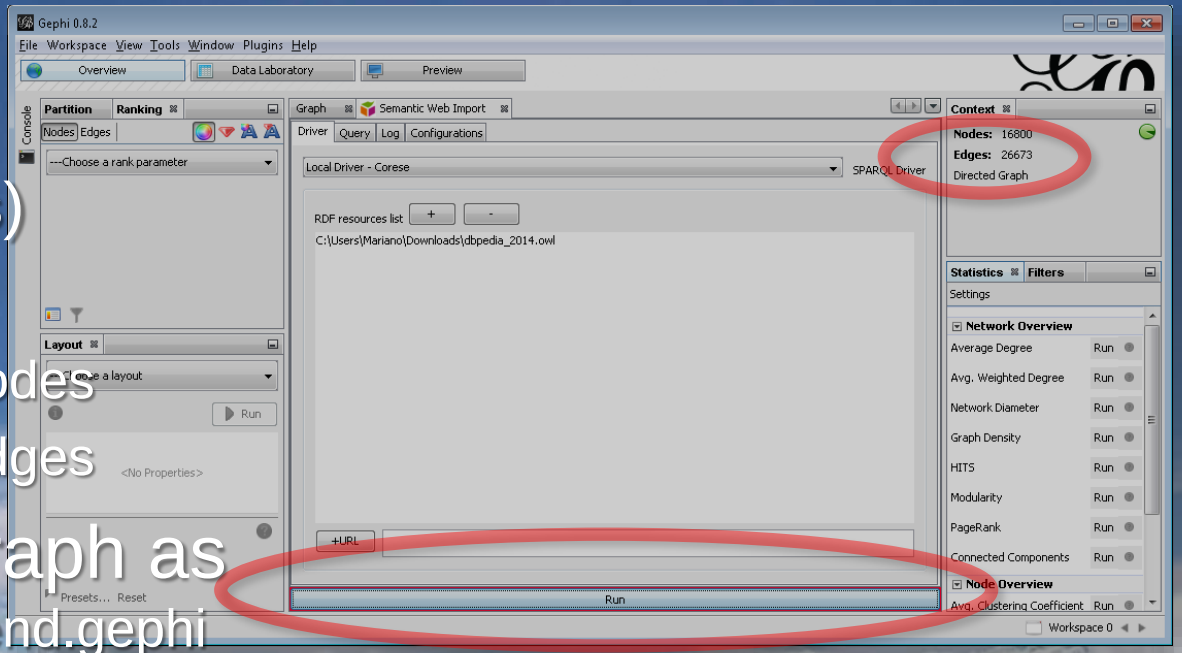
Gephi

- 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

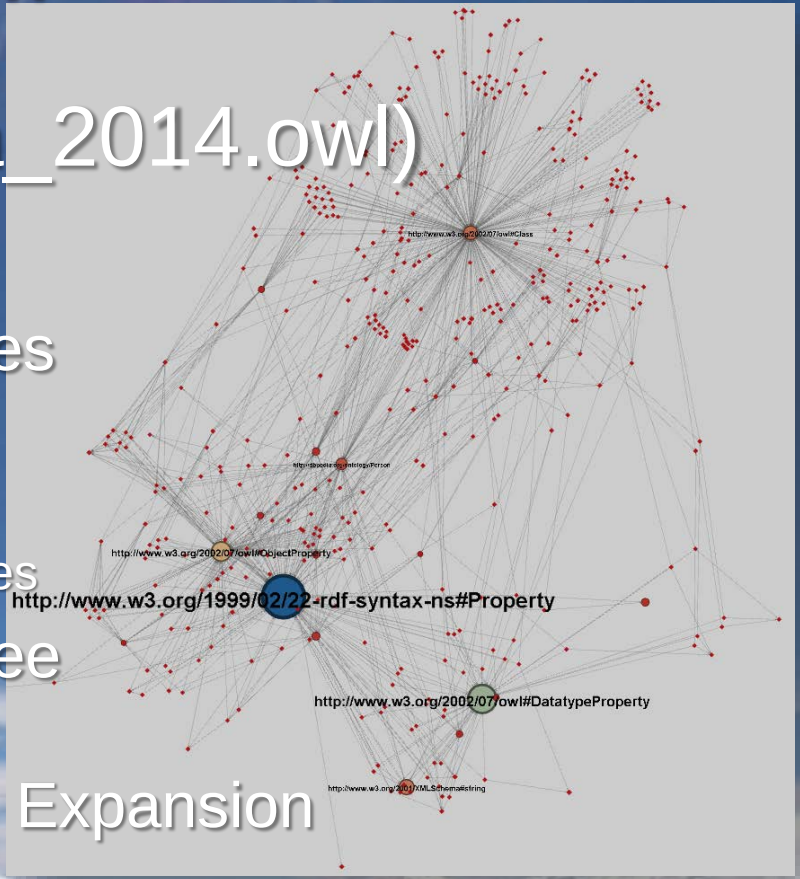
  – Filter nodes. “Remove” nodes with degree < 10

    – 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



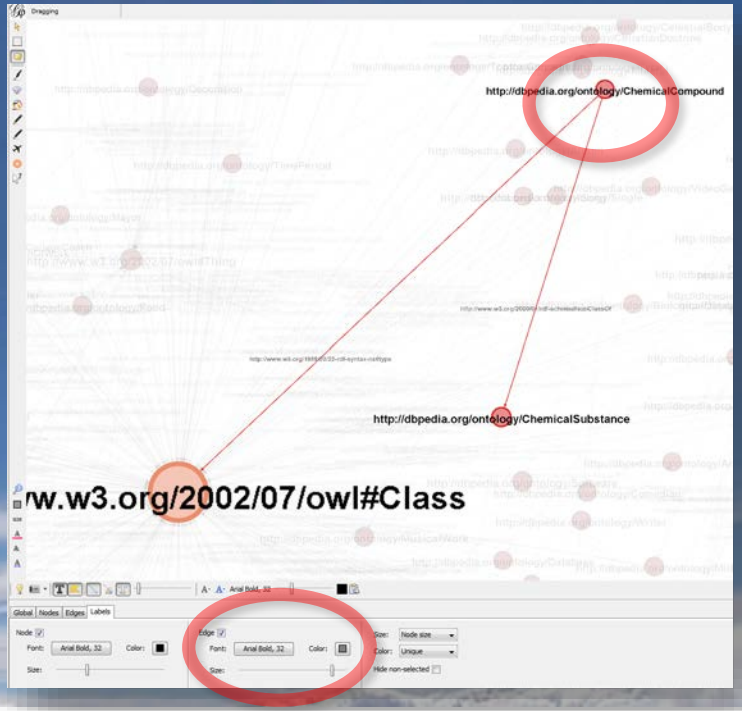
© @marianorico

Gephi

- Load local file (dbpedia_2014.owl)

  Result analysis

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



© @marianorico

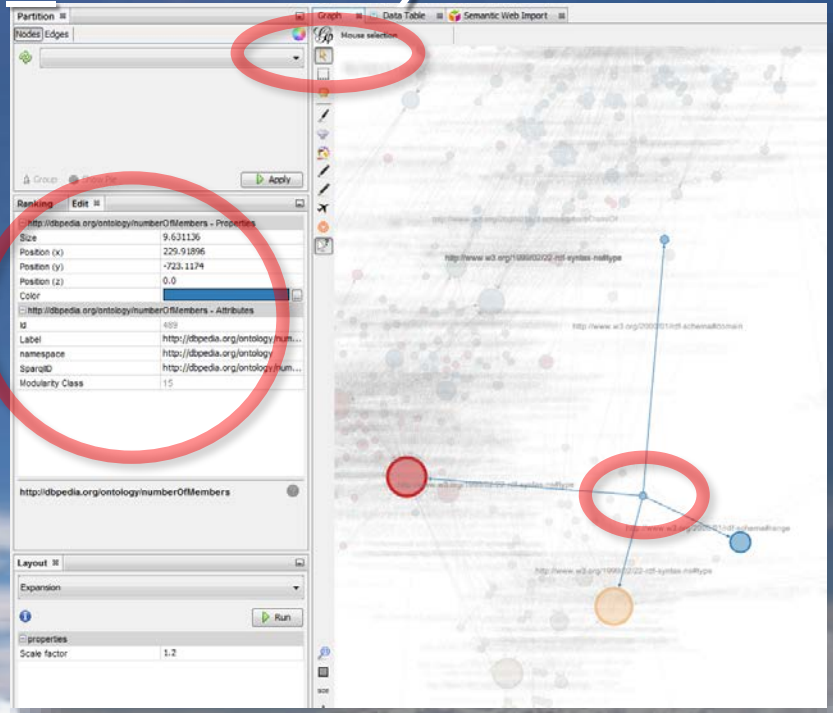
Gephi

- Load local file (dbpedia_2014.owl)

  Result analysis

  – Use the selection tool

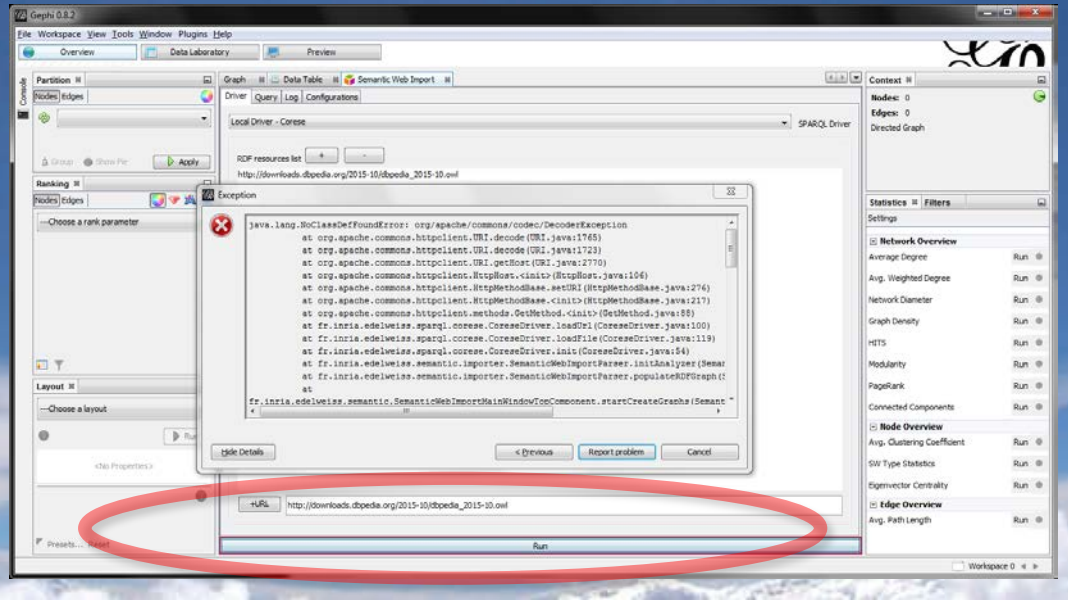
    - Click on a node to see its properties in a tab (“Edit tab”)
    - Hide node labels
    - Increase edges’ label size
    - Now you see the node`s name in the “Edit tab” and its edges labels



© @marianorico

Gephi

- Load online file
  - e.g. http://downloads.dbpedia.org/2015-10/dbpedia_2015-10.owl
- Execution error
  - At lest on Windows 7 ☹



© @marianorico

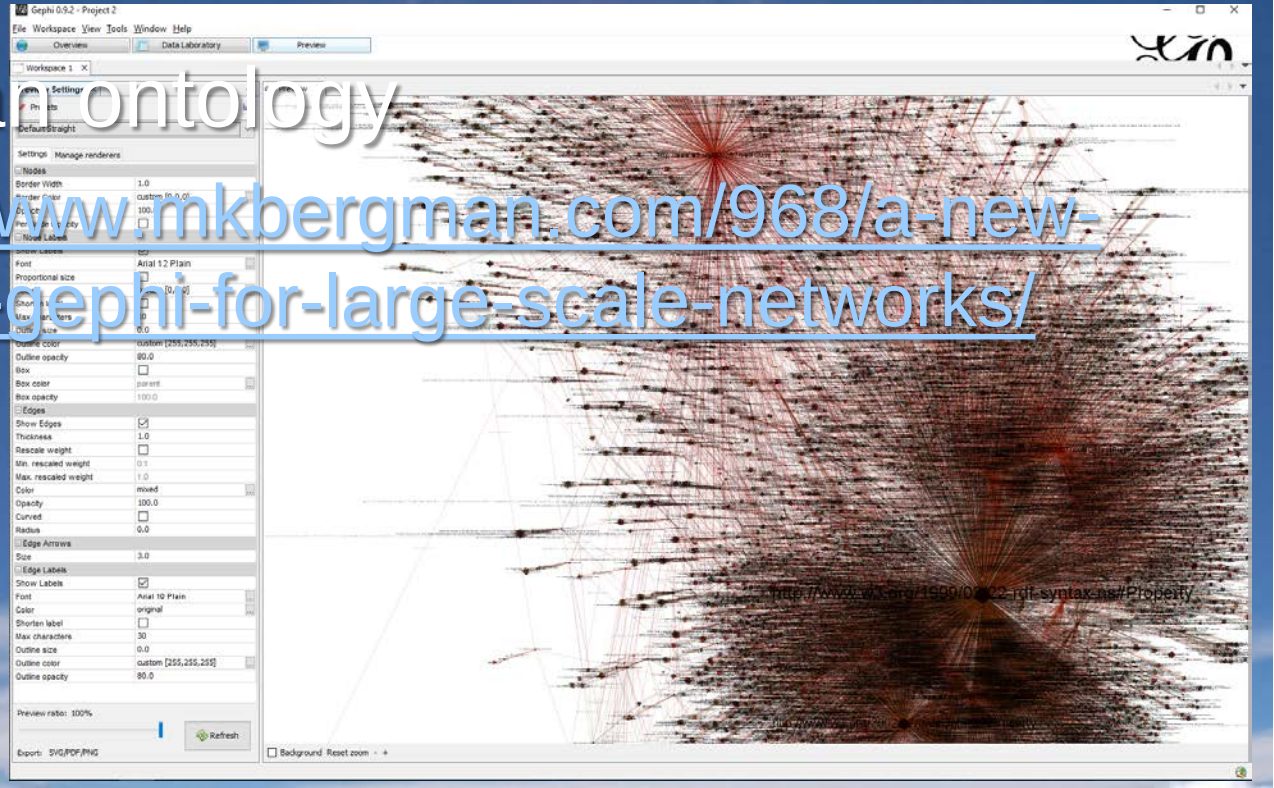
Gephi

- Load online file
  - e.g. http://downloads.dbpedia.org/2016-10/dbpedia_2016-10.owl
- Goes fine
  - At lest on Windows 10 ☺
    although it takes ~10 min to load ☹

© @marianorico

Gephi

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



© @marianorico

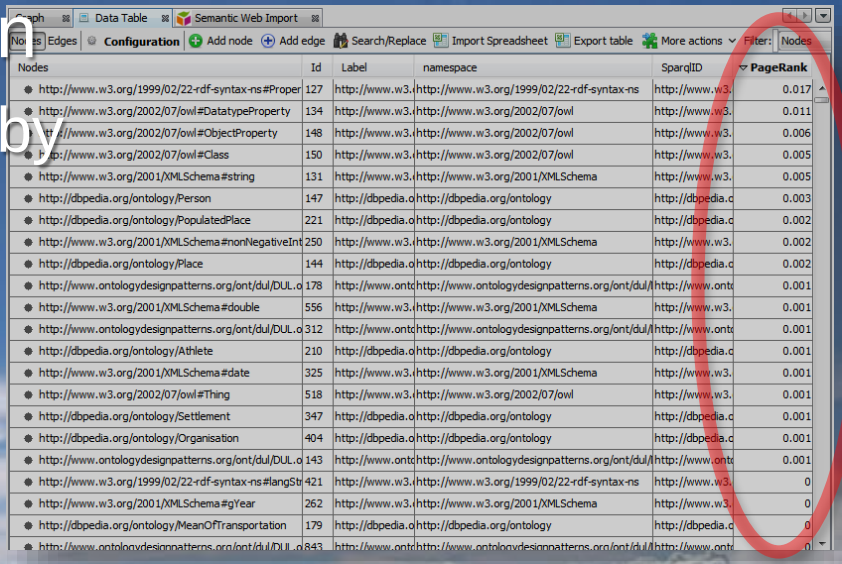
Gephi

- 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**

© @marianorico

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**

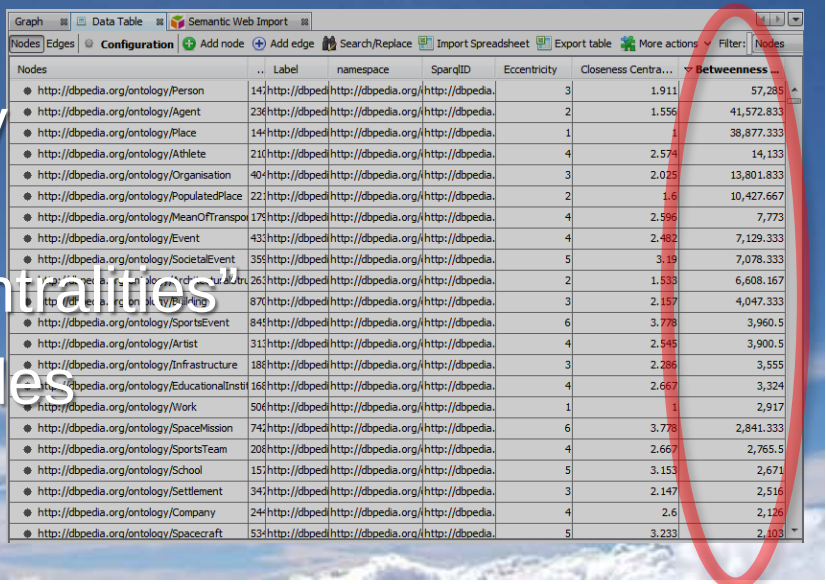


Nodes	Id	Label	namespace	SparqlID	PageRank
http://www.w3.org/1999/02/22-rdf-syntax-ns#Proper	127	http://www.w3.	http://www.w3.org/1999/02/22-rdf-syntax-ns	http://www.w3.	0.017
http://www.w3.org/2002/07/owl#DatatypeProperty	134	http://www.w3.	http://www.w3.org/2002/07/owl	http://www.w3.	0.011
http://www.w3.org/2002/07/owl#ObjectProperty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

© @marianorico

Gephi

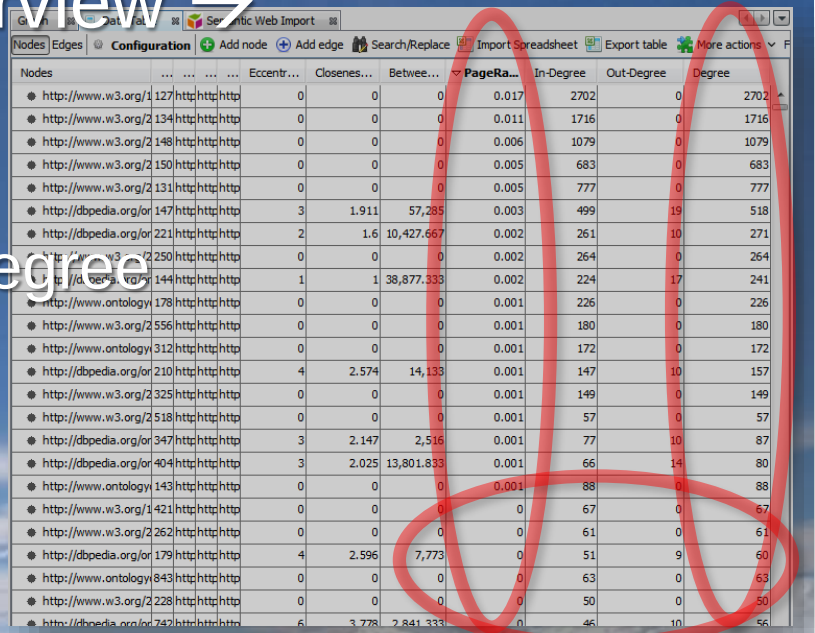
- The DBpedia Ontology use case
  - Statistics →Edge Overview → Avg. Path Length→ Run
    - See the data table, sort by Betweenness Centrality
    - Do NOT tic "normalize centralities"
    - Notice that now, more nodes have a value != 0 → **Do NOT normalize!!**



© @marianorico

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 ☺

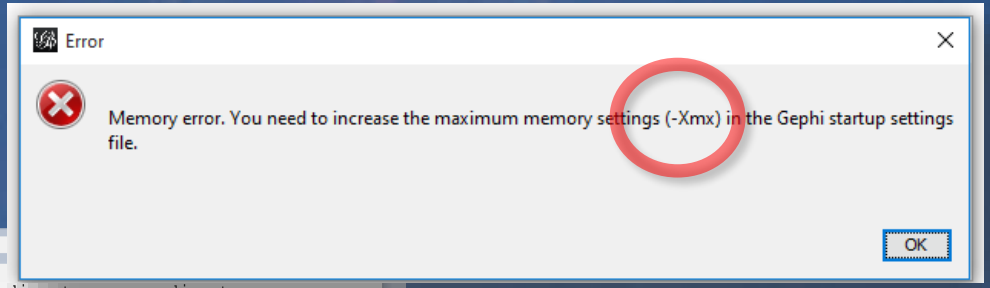


Nodes	...	...	...	...	Eccentr...	Closenes...	Betwee...	PageRa...	In-Degree	Out-Degree	Degree
http://www.w3.org/1	127	http	http	http	0	0	0	0.017	2702	0	2702
http://www.w3.org/2	134	http	http	http	0	0	0	0.011	1716	0	1716
http://www.w3.org/2	148	http	http	http	0	0	0	0.006	1079	0	1079
http://www.w3.org/2	150	http	http	http	0	0	0	0.005	683	0	683
http://www.w3.org/2	131	http	http	http	0	0	0	0.005	777	0	777
http://dbpedia.org/or	147	http	http	http	3	1.911	57,285	0.003	499	19	518
http://dbpedia.org/or	221	http	http	http	2	1.6	10,427.667	0.002	261	10	271
http://www.w3.org/2	250	http	http	http	0	0	0	0.002	264	0	264
http://dbpedia.org/or	144	http	http	http	1	1	38,877.333	0.002	224	17	241
http://www.ontology	178	http	http	http	0	0	0	0.001	226	0	226
http://www.w3.org/2	556	http	http	http	0	0	0	0.001	180	0	180
http://www.ontology	312	http	http	http	0	0	0	0.001	172	0	172
http://dbpedia.org/or	210	http	http	http	4	2.574	14,133	0.001	147	10	157
http://www.w3.org/2	325	http	http	http	0	0	0	0.001	149	0	149
http://www.w3.org/2	518	http	http	http	0	0	0	0.001	57	0	57
http://dbpedia.org/or	347	http	http	http	3	2.147	2,516	0.001	77	10	87
http://dbpedia.org/or	404	http	http	http	3	2.025	13,801.833	0.001	66	14	80
http://www.ontology	143	http	http	http	0	0	0	0.001	88	0	88
http://www.w3.org/1	421	http	http	http	0	0	0	0	67	0	67
http://www.w3.org/2	262	http	http	http	0	0	0	0	61	0	61
http://dbpedia.org/or	179	http	http	http	4	2.596	7,773	0	51	9	60
http://www.ontology	843	http	http	http	0	0	0	0	63	0	63
http://www.w3.org/2	228	http	http	http	0	0	0	0	50	0	50
http://dbpedia.org/or	742	http	http	http	6	3.778	2,841.333	0	46	10	56

© @marianorico

Gephi limitations

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

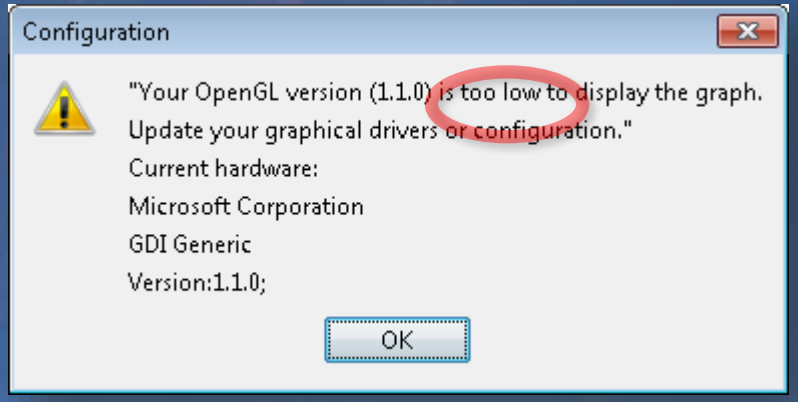



```
# ${HOME} will be replaced by user home directory according to
platform
default_userdir="${HOME}/.${APPNAME}/0.8.2/dev"
default_mac_userdir="${HOME}/Library/Application Support/
${APPNAME}/0.8.2/dev"

# 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.noddraw=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=
```

© @marianorico

Gephi limitations

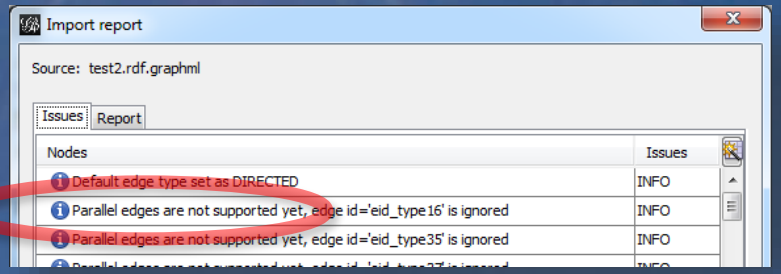
- OpenGL
  - Update your graph drivers
    - For Intel go here

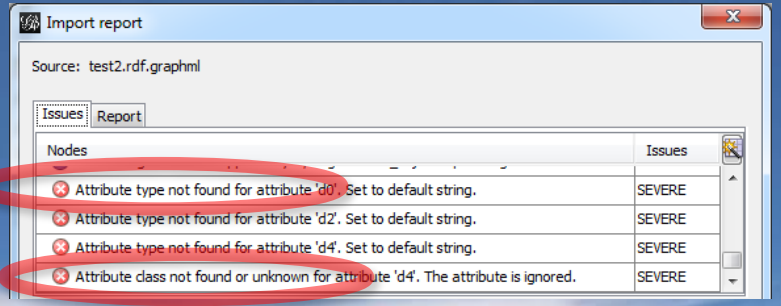


© @marianorico

Gephi limitations

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

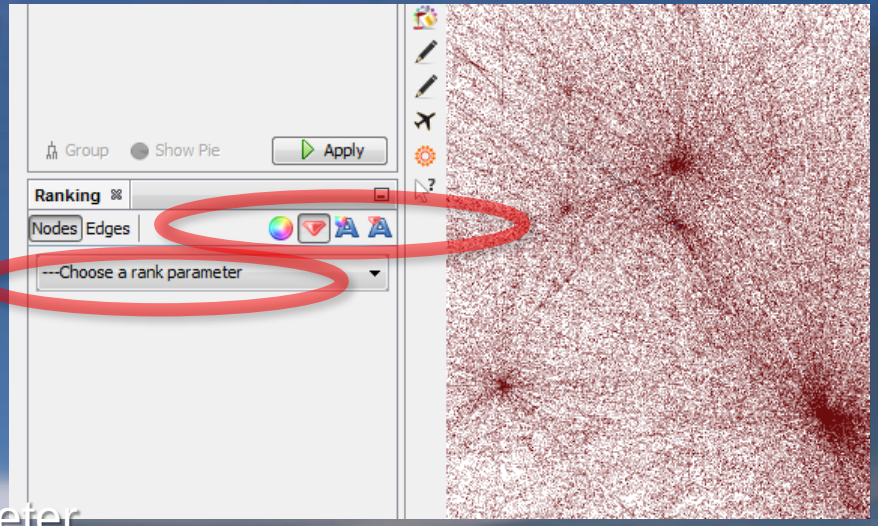




© @marianorico

Gephi limitations

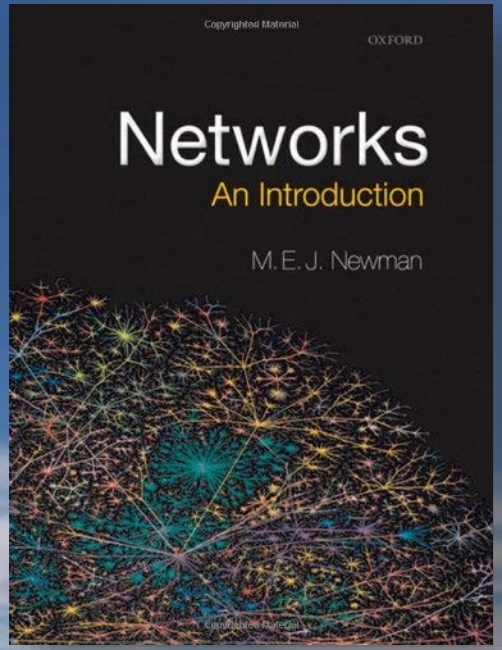
- Usability
  - UI selectors are confusing
    - Always show the last used option
    - They do not use the current parameter ☹
    - E.g. load a graph
      - In the “Ranking” tab you cannot know which parameter has been used for node size, color, etc.

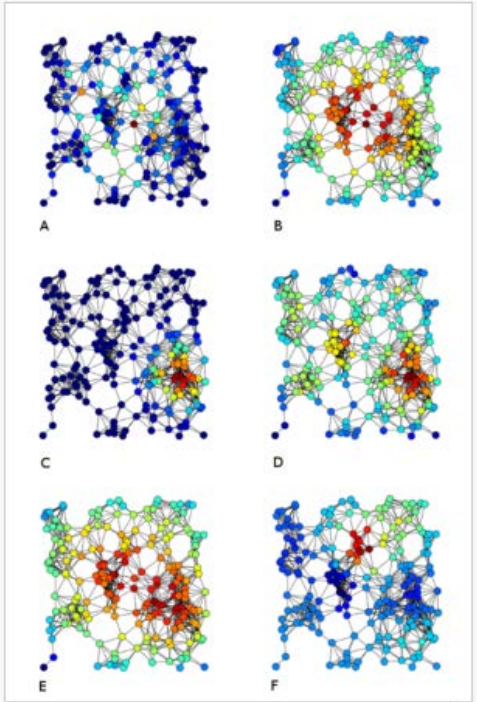


© @marianorico

More on graph theory

- Centrality
  - Wikipedia
- Graphs
  - Newman's book, 2010.





Examples of A) Betweenness centrality, B) Closeness centrality, C) Eigenvector centrality, D) Degree centrality, E) Harmonic centrality and F) Katz centrality of the same graph.

© @marianorico

More on Gephi Layouts

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

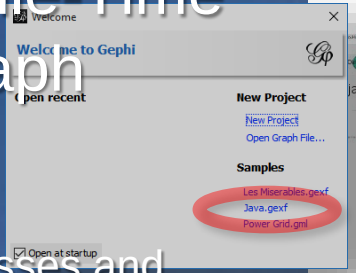
Selecting the Layout

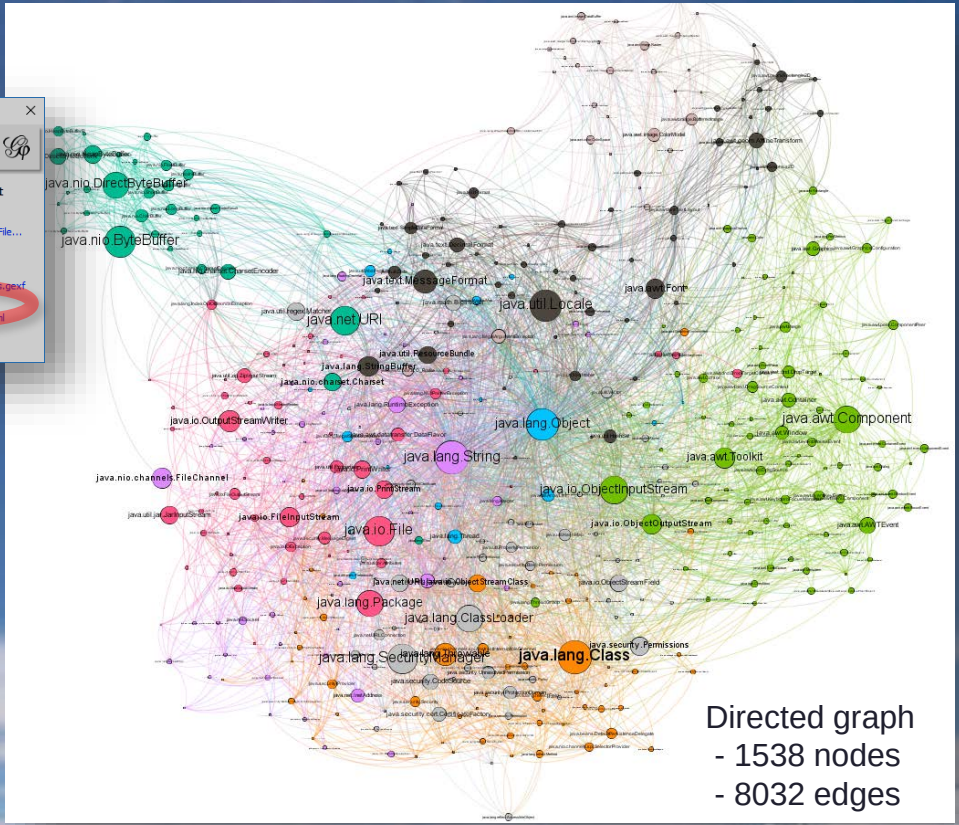
Algorithm name	Type	Strengths	Weaknesses	When to use
DAG layout	Tree	Ordering hierarchical data	This is impractical for very large networks	This can be used in cases where you wish to see levels of data in a top to bottom order
Dual Circle layout	Circular	This has the ability to focus on a group of nodes 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

© @marianorico

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



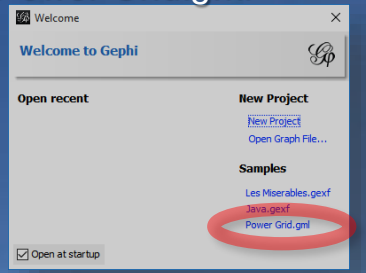


Directed graph
- 1538 nodes
- 8032 edges

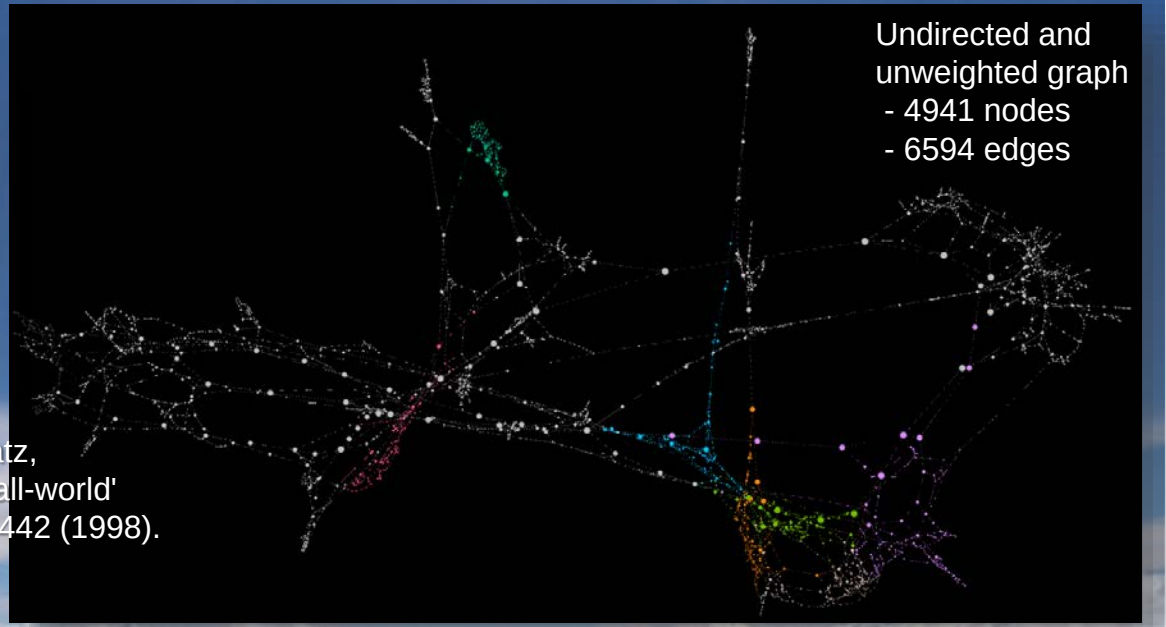
© @marianorico

More examples

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



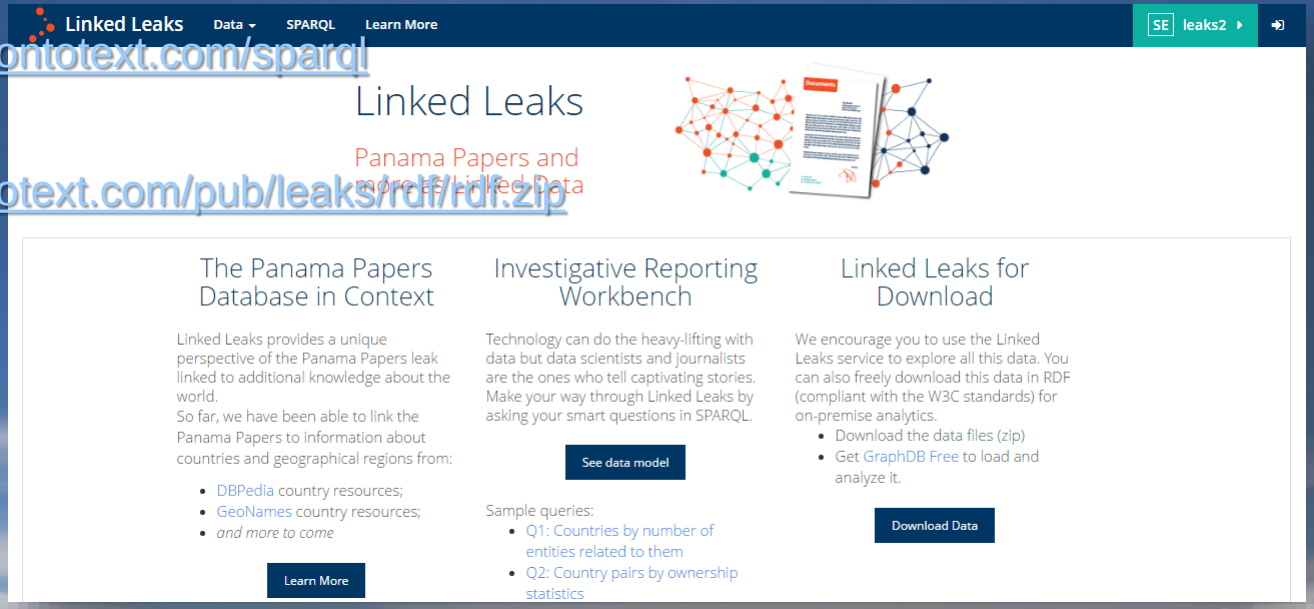
  - Original paper citation
    - D. J. Watts and S. H. Strogatz, "Collective dynamics of `small-world' networks", Nature 393, 440-442 (1998).



© @marianorico

More SPARQL examples

- Panama papers
  - EP
    - http://data.ontotext.com/sparql
  - in RDF
    - ftp://ftp.ontotext.com/pub/leaks/rdf/rdf.zip



© @marianorico

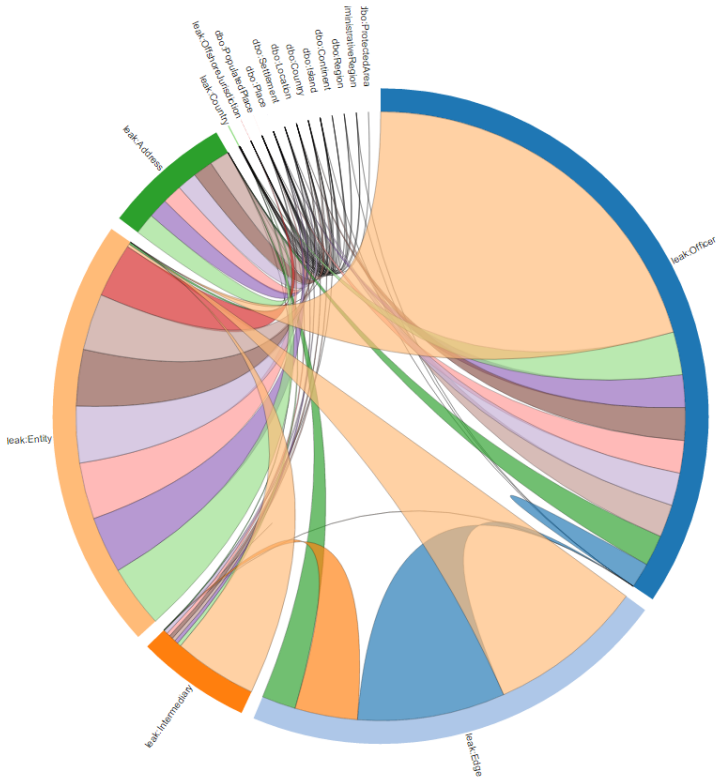
More visualizations

Linked Leaks Data SPARQL Learn More

Class relationships

Class relationships

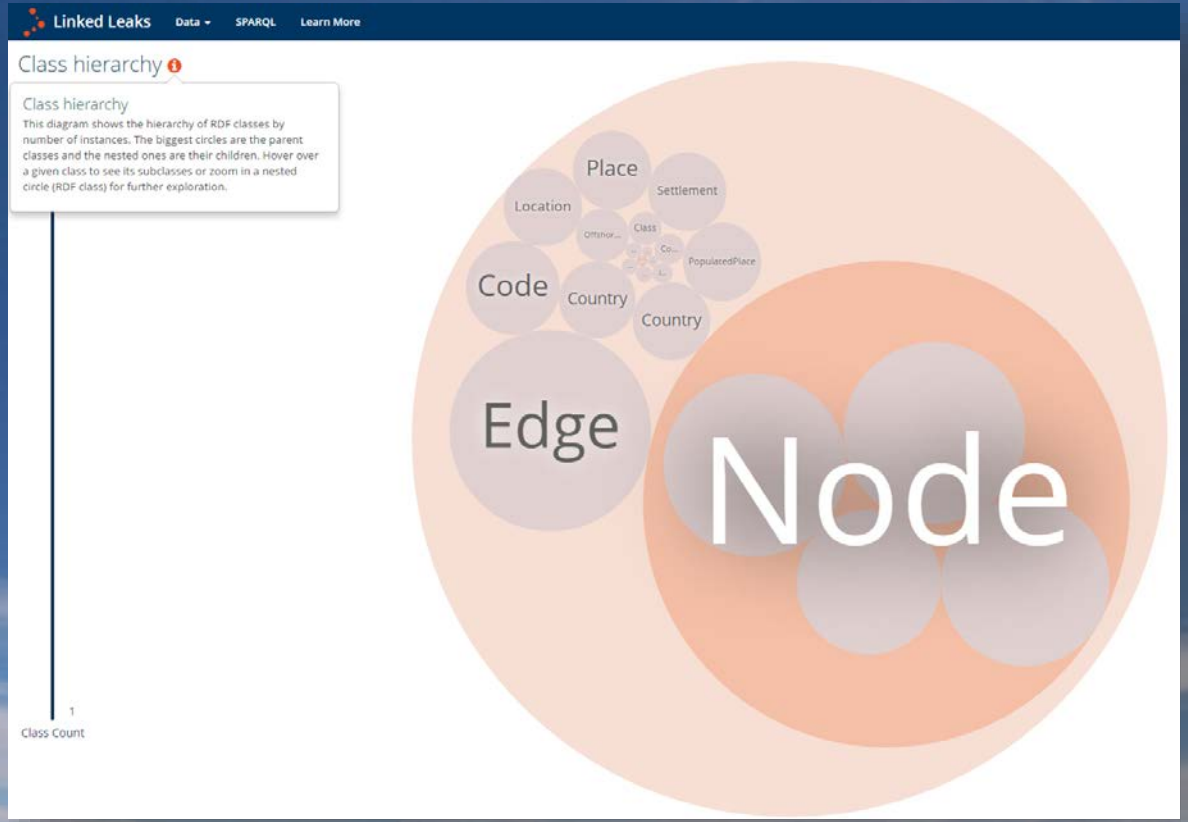
This diagram shows the relationships between RDF classes, where a relationship is represented by links between the individual instances of two classes. Each link is an RDF statement where the subject is an instance of one class, the object is an instance of another class, and the link is the predicate. Depending on the number of links between the instances of two classes, the bundle can be thicker or thinner and it gets the colour of the class with more incoming links. The links can be in both directions.



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

© @marianorico

More visualizations

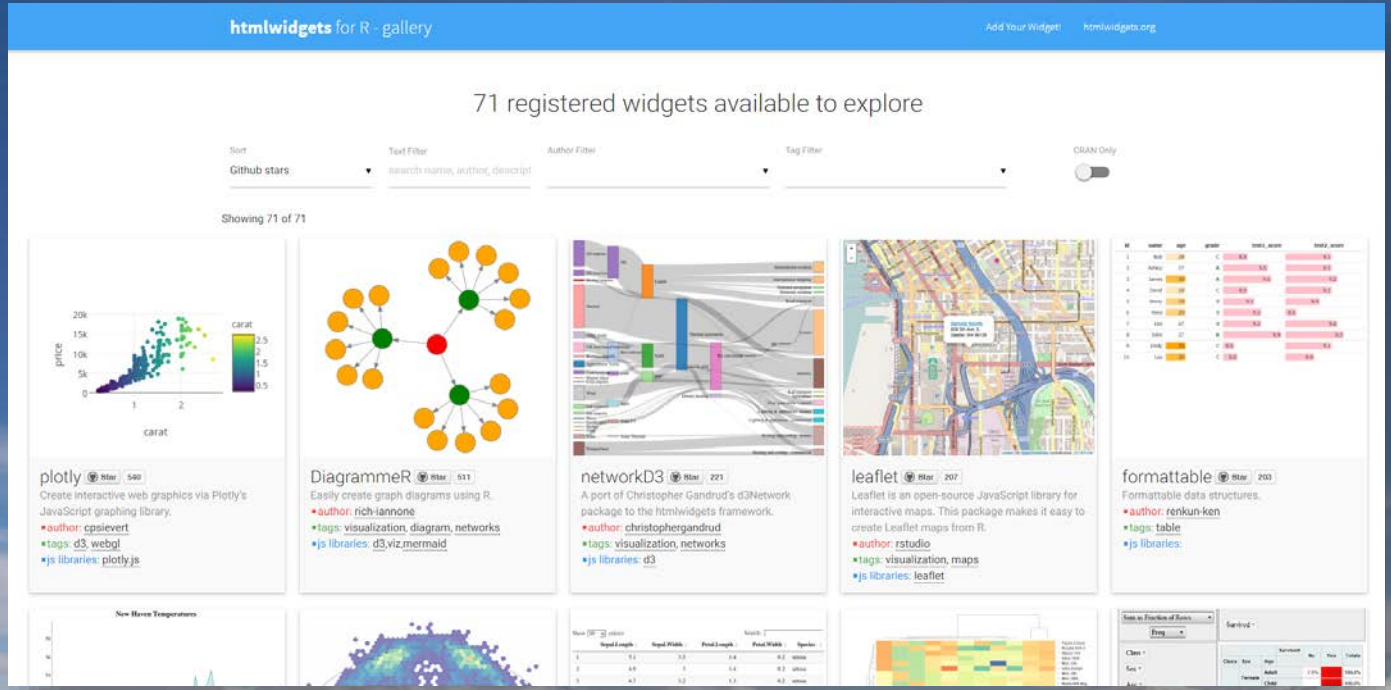


See http://data.ontotext.com/sparql

© @marianorico

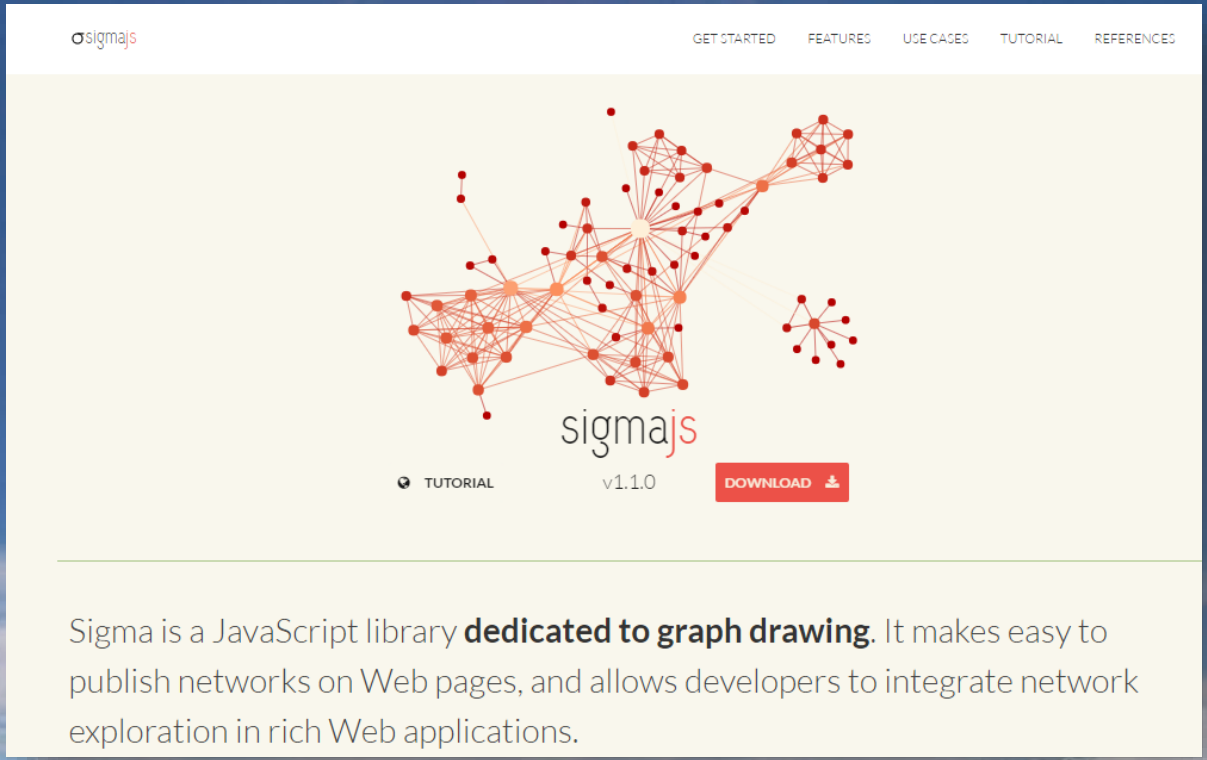
More visualizations

- The coolest for R: package htmlwidgets



Displaying on a web page

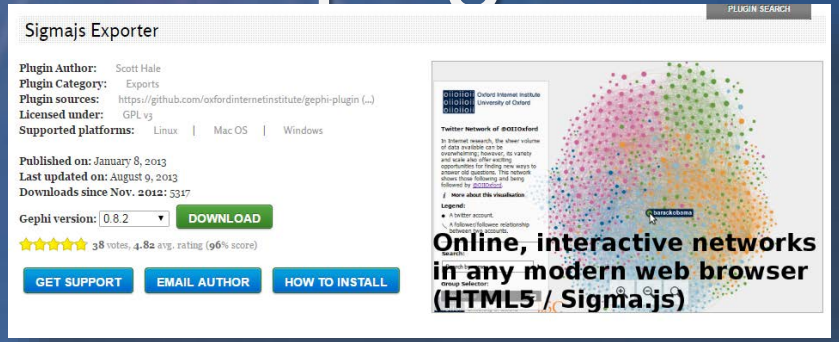
- Sigma
  sigmajs.org
  - Javascript
  - Reads .gexf files ☺
  - R package
    - github.com/jjallaire/sigma



© @marianorico

Displaying on a web page

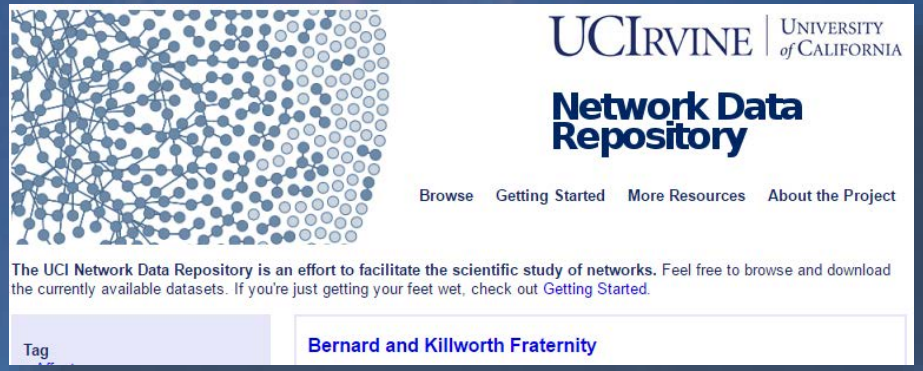
- Gephi exporter for Sigma
  - Generates all the html/js/json code in a given folder
    - Requires a html server (like Tomcat)
      - Does not work as a html file on the browser
  - Limitation: No labels ☹
  - Installation
    - Gephi 0.8.2 vía Marketplace
    - Gephi 0.9.1 vía Tools→Plugins→ Available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



© @marianorico

More datasets

- UCI Network data repository
  - https://networkdata.ics.uci.edu/

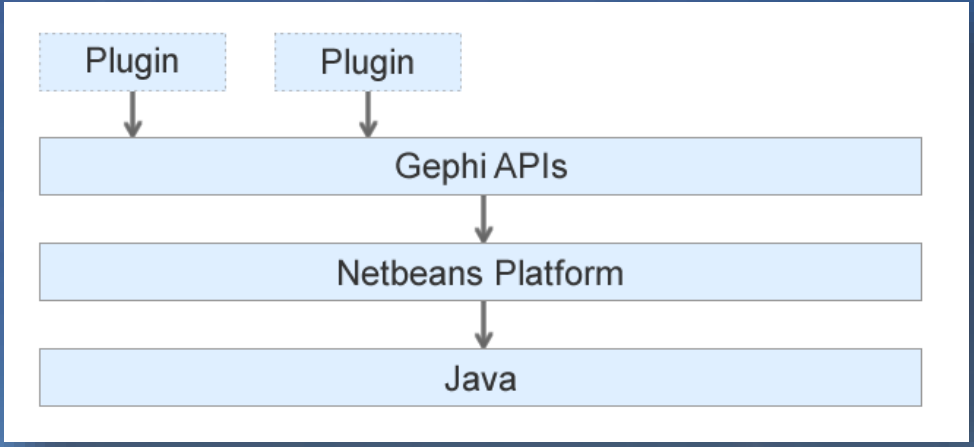


- Nexus (igraph.org)
  - http://nexus.igraph.org/api/dataset_info

© @marianorico

Gephi API

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



- More at Gephi.org (https://github.com/gephi/gephi/wiki/How-to-build-Gephi)

© @marianorico

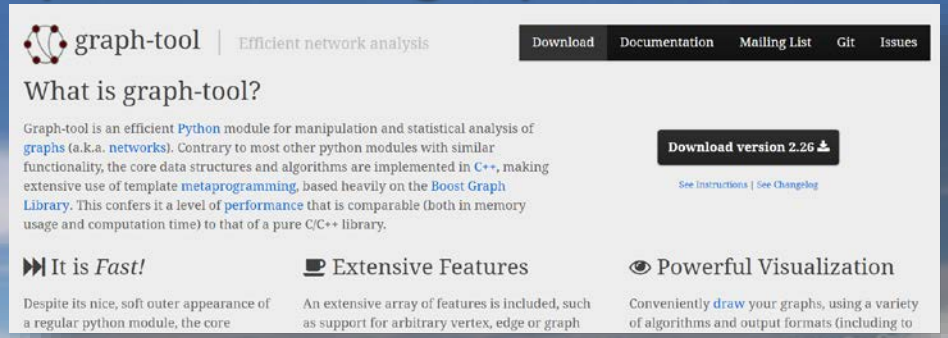
Gephi & R

- 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

© @marianorico

For Python fans

- Igraph
  - Also available for R fans
- graph-tool
  - ToDo: make a comparison to Igraph



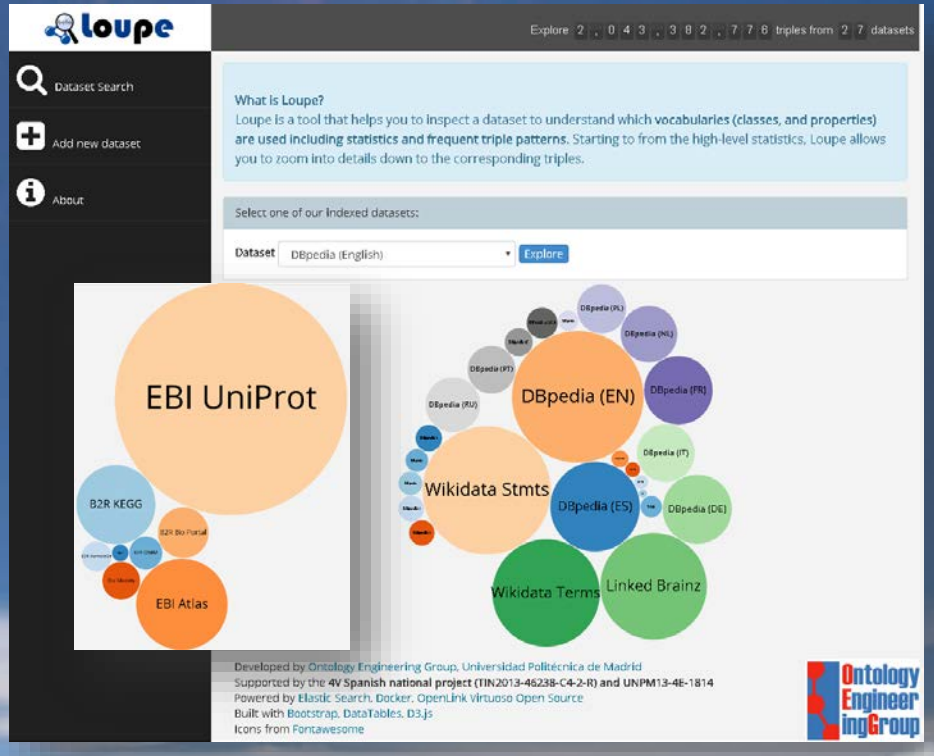
Exploring a given dataset

LOUPE

© @marianorico

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



© @marianorico

Loupe

- Better an online demo ☺

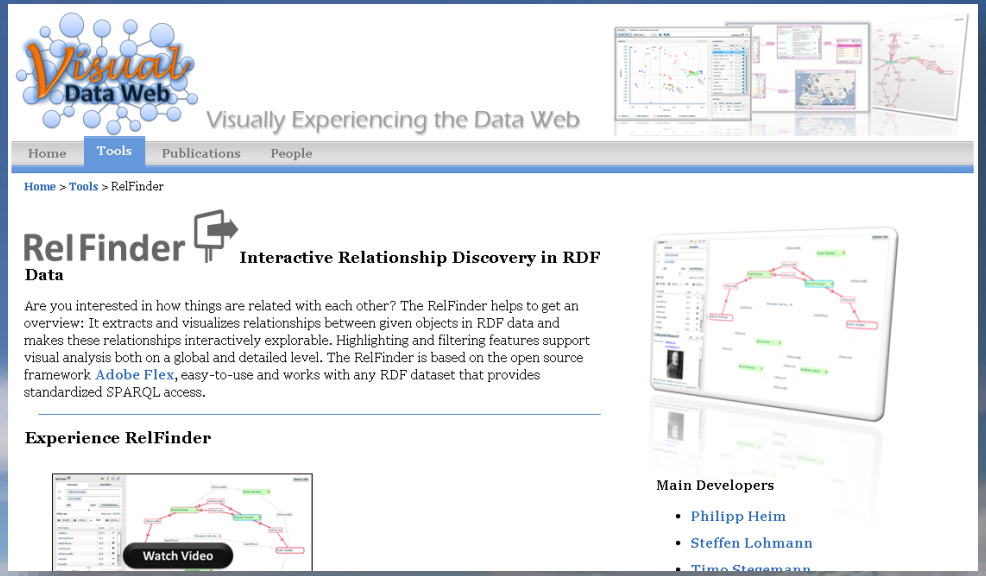
A first glimpse of graphs with

RELFINDER

© @marianorico

RelFinder

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



© @marianorico

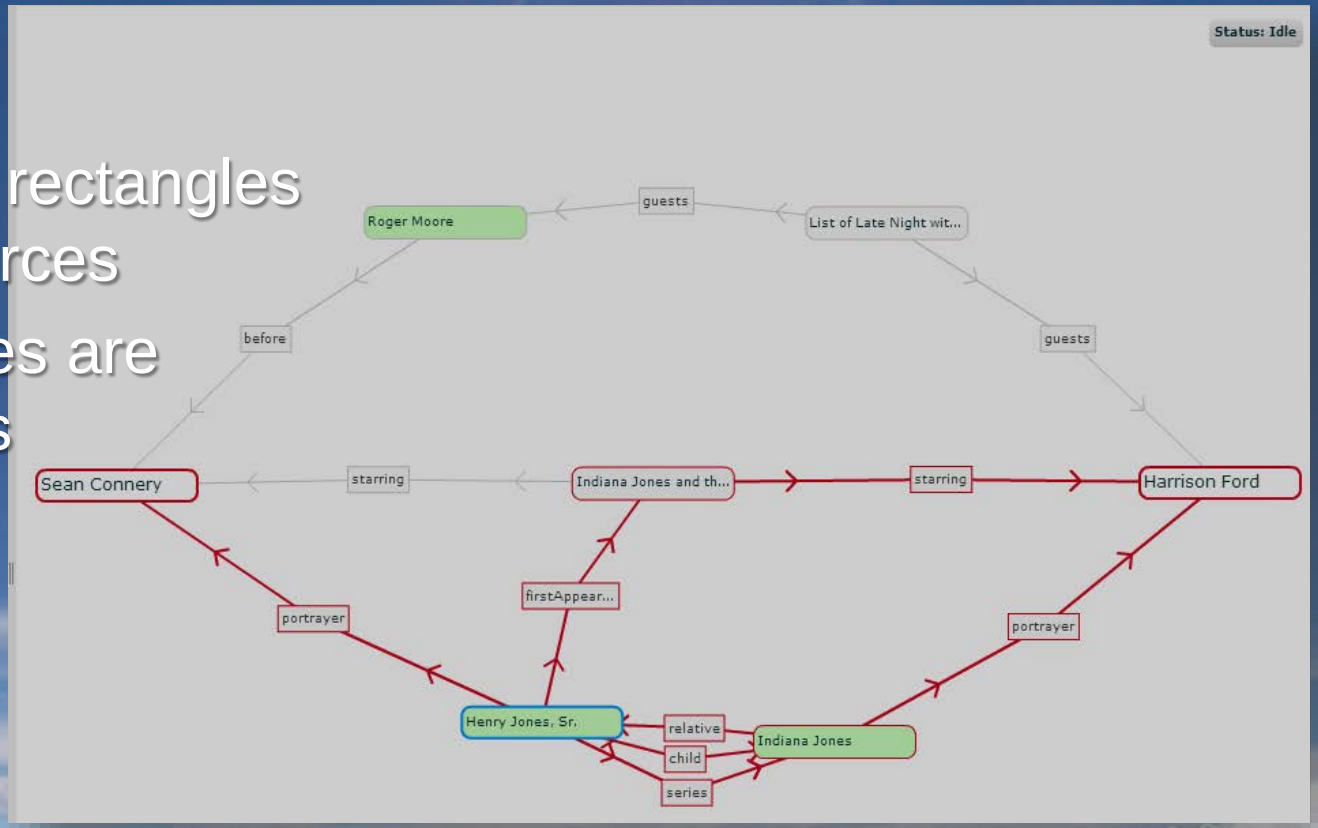
RelFinder

- Look for the relationship between Sean Connery and Harrison Ford

© @marianorico

RelFinder

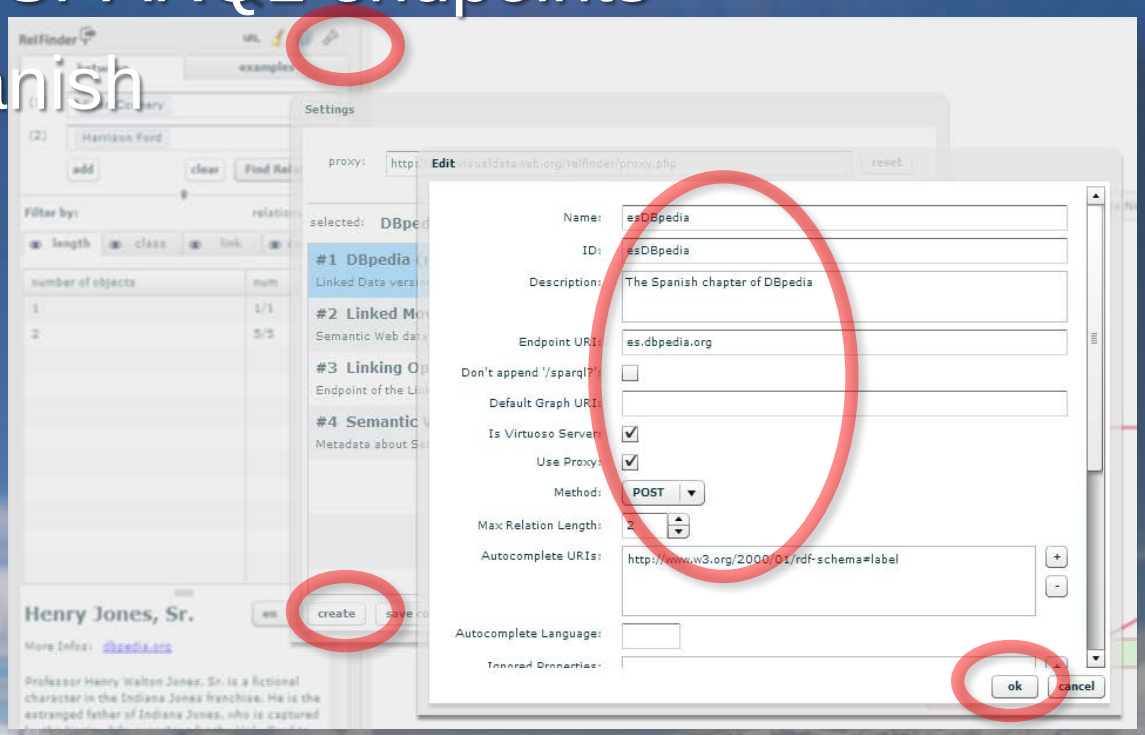
- Result
  - Rounded rectangles are resources
  - Rectangles are properties



© @marianorico

RelFinder

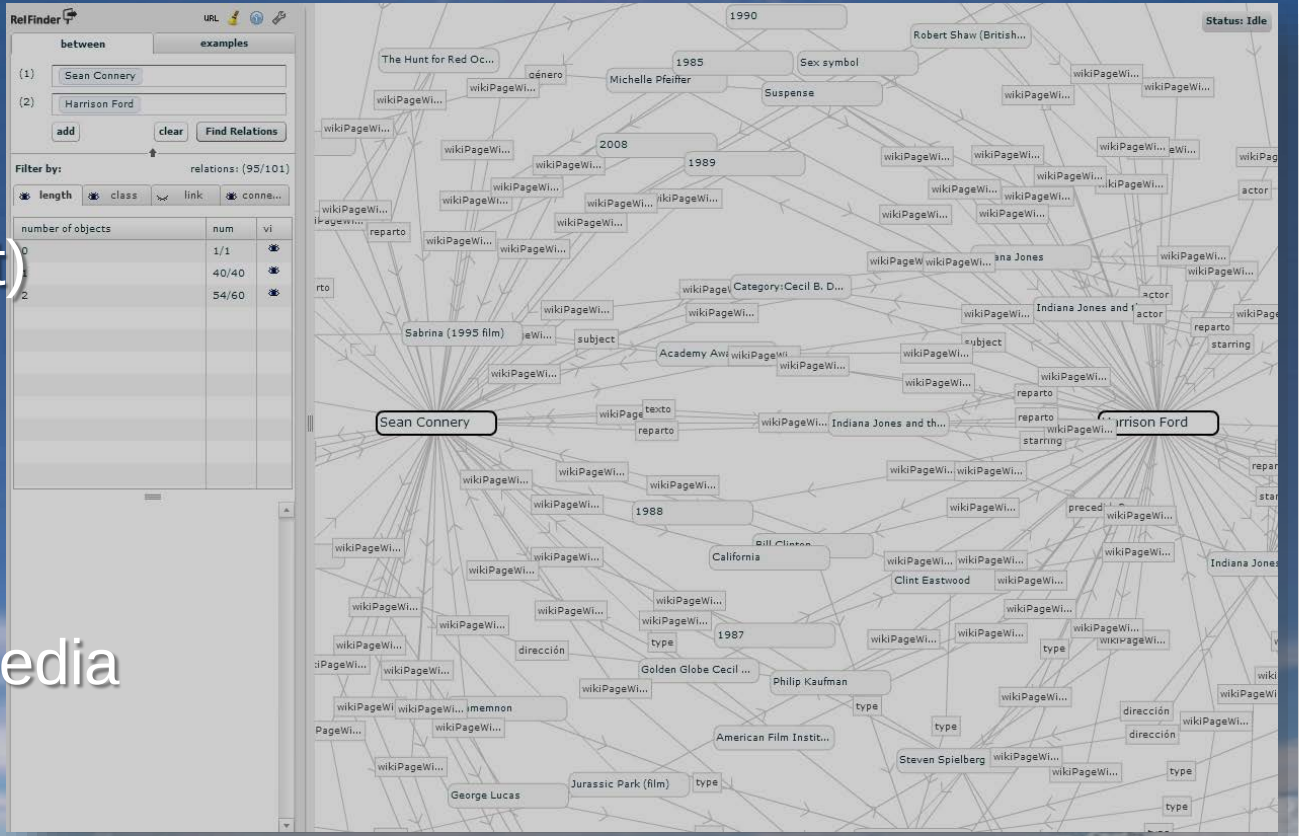
- Setup for other SPARQL endpoints
  - E.g. The Spanish DBpedia



© @marianorico

RelFinder

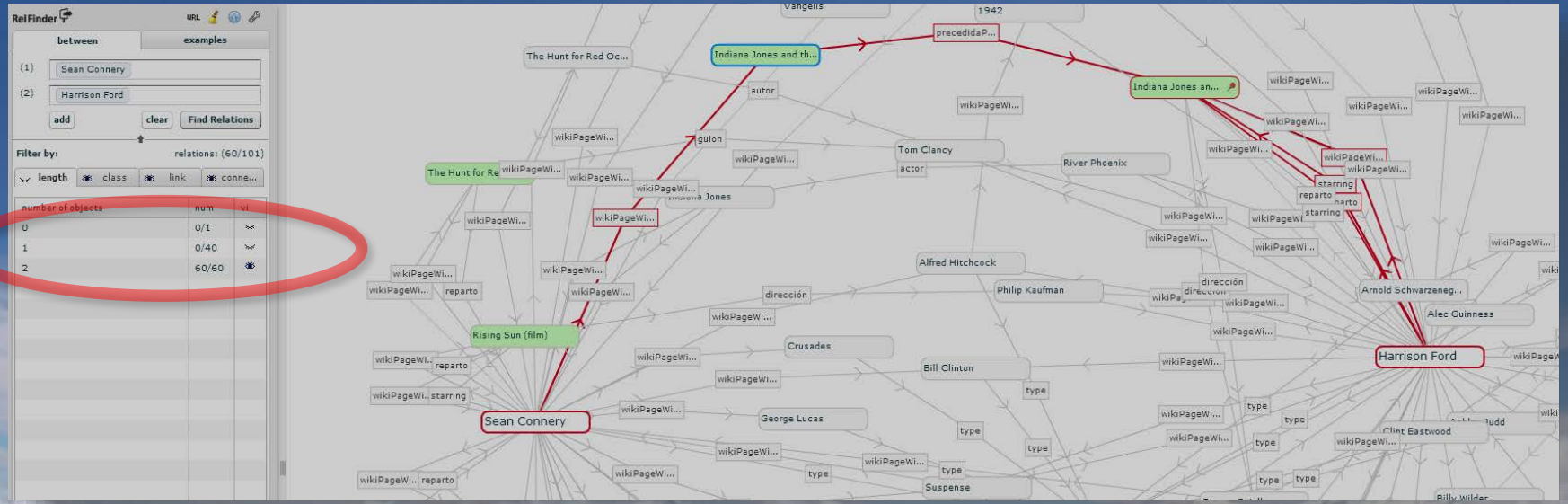
- Result
  - Much more than (default) DBpedia
  - Hypothesis: They use a old private copy of DBpedia



© @marianorico

RelFinder

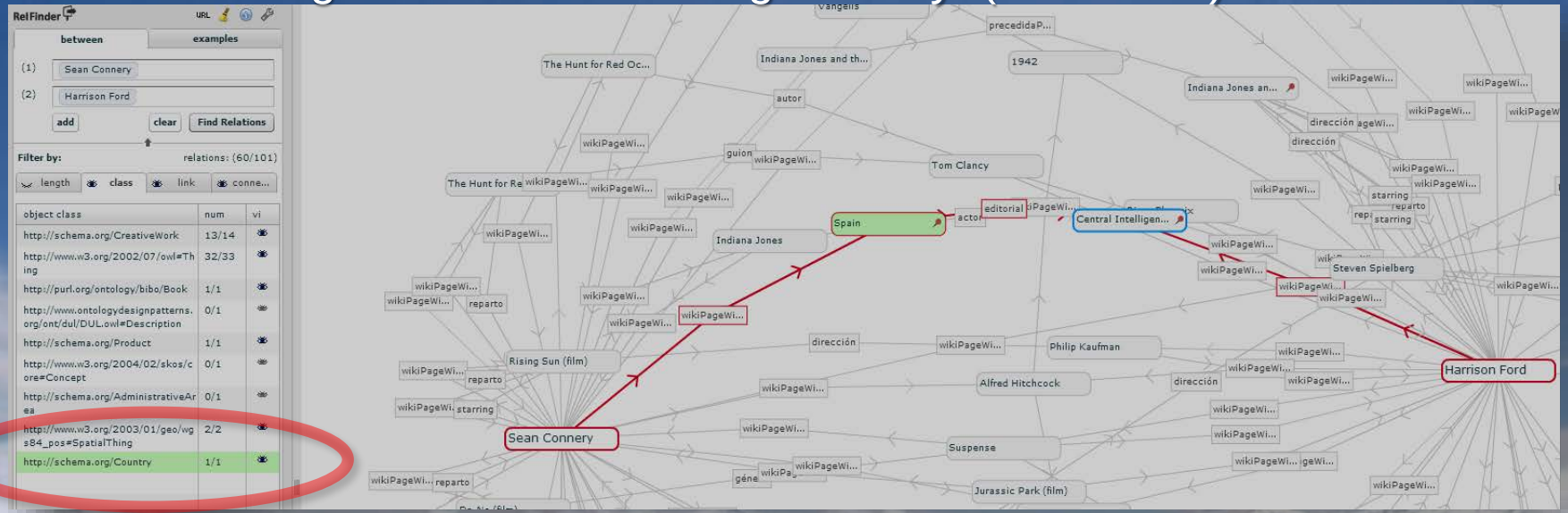
- Filter by
  - Length (number of resources in between)
    - E.g. length = 2 ➔ 2 resources in between



© @marianorico

RelFinder

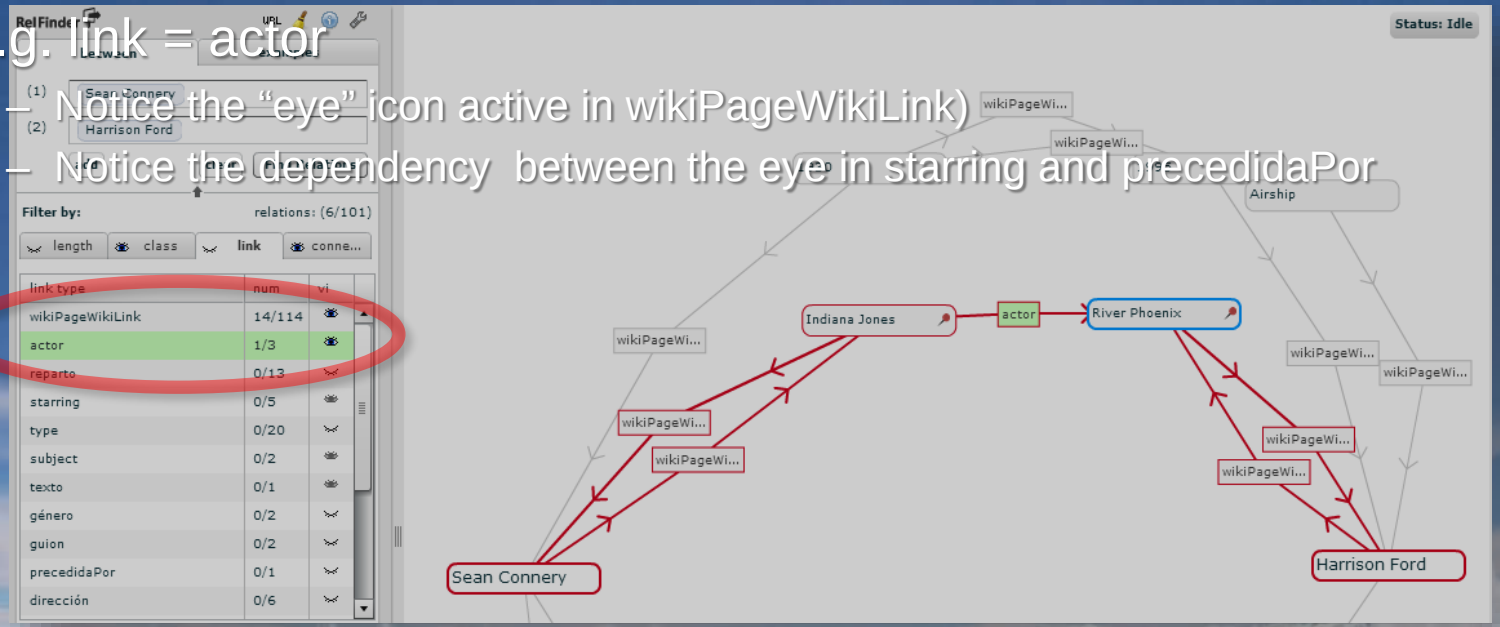
- Filter by
  - Class (list of resource's class)
    - E.g. class = schema.org/Country (num = 1/1)



© @marianorico

RelFinder

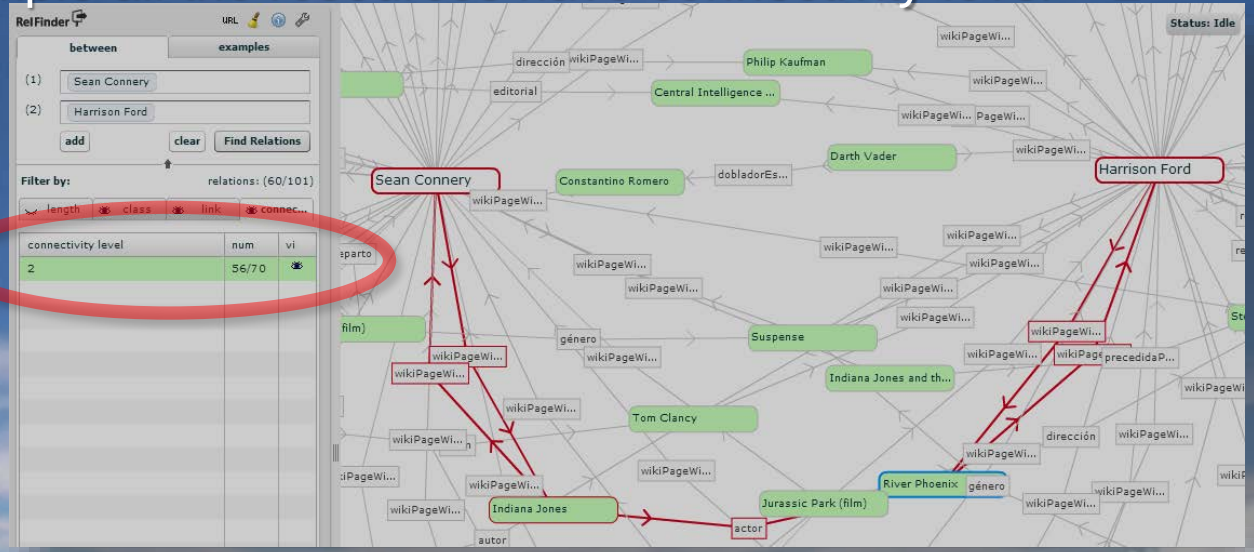
- Filter by
  - Link (list of resource's properties)
    - E.g. link = actor
      - Notice the "eye" icon active in wikiPageWikiLink)
      - Notice the dependency between the eye in starring and precedidaPor



© @marianorico

RelFinder

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



© @marianorico

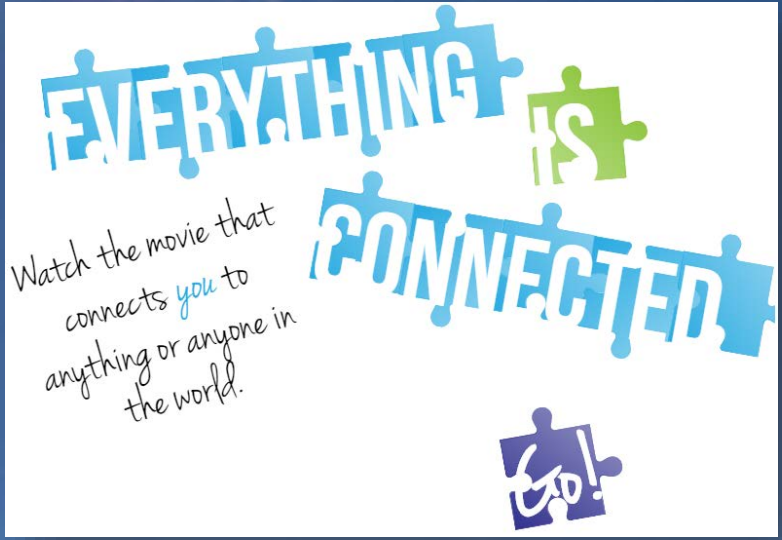
RelFinder

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

© @marianorico

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

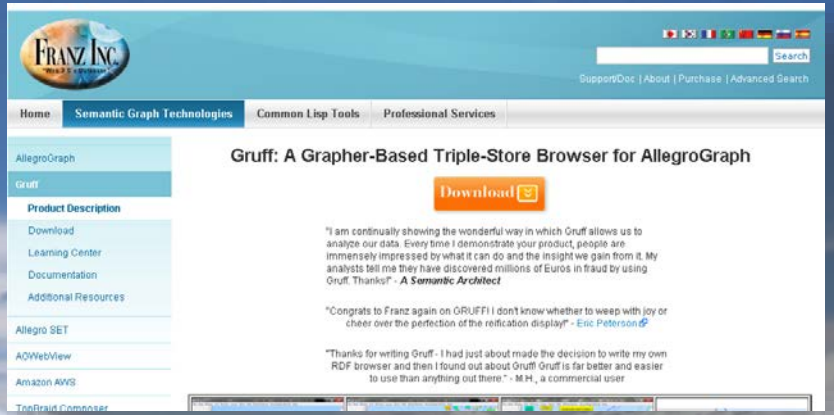
Ordered small graphs

GRUFF

© @marianorico

Gruff

- Private product (evaluation license)
- Last version includes access to EPs
  - You have to request a month-key
    - July 2016 key: longproton
- Windows/Linux/Mac



© @marianorico

Gruff

- Windows/Linux/Mac

Gruff for AllegroGraph 3.3

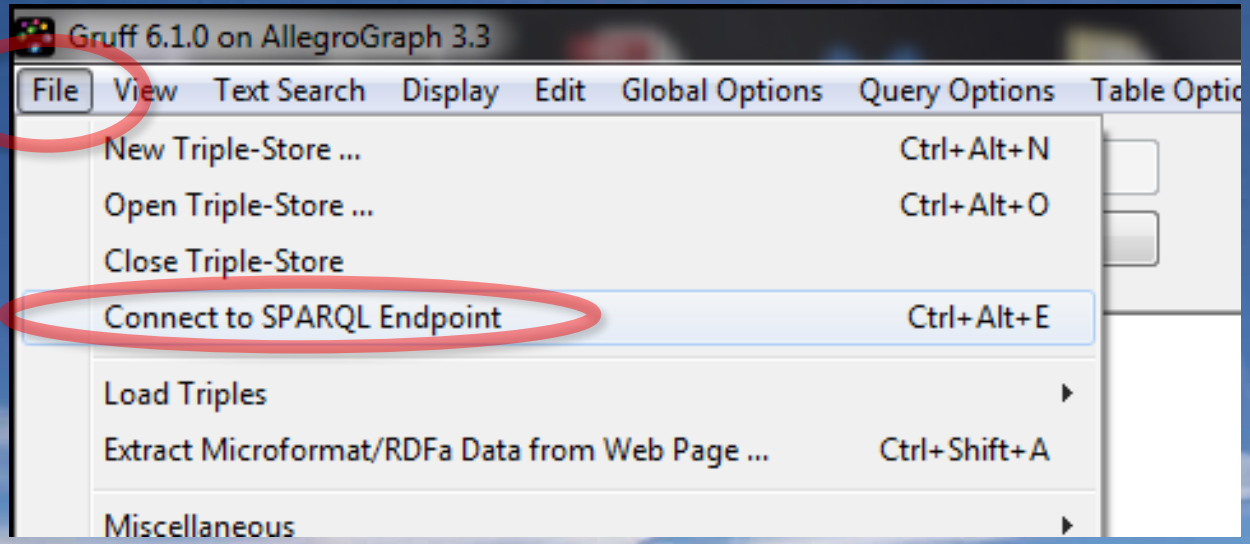
A Standalone Gruff with built-in AllegroGraph 3.3, **no Additional Download Required**:

Platform	Download	System Requirements
64-bit Linux	Gruff v6.1.0-AG3.3	glibc 2.4 or newer -- Installation instructions
32-bit Linux	Gruff v6.1.0-AG3.3	glibc 2.3 or newer -- Installation instructions
64-bit Mac OS X	Gruff v6.1.0-AG3.3	Mac OS X 10.6 or newer -- Installation instructions
64-bit Windows	Gruff v6.1.0-AG3.3	Windows Vista or newer -- Installation instructions
32-bit Windows	Gruff v6.1.0-AG3.3	Windows Vista or newer -- Installation instructions

© @marianorico

Gruff

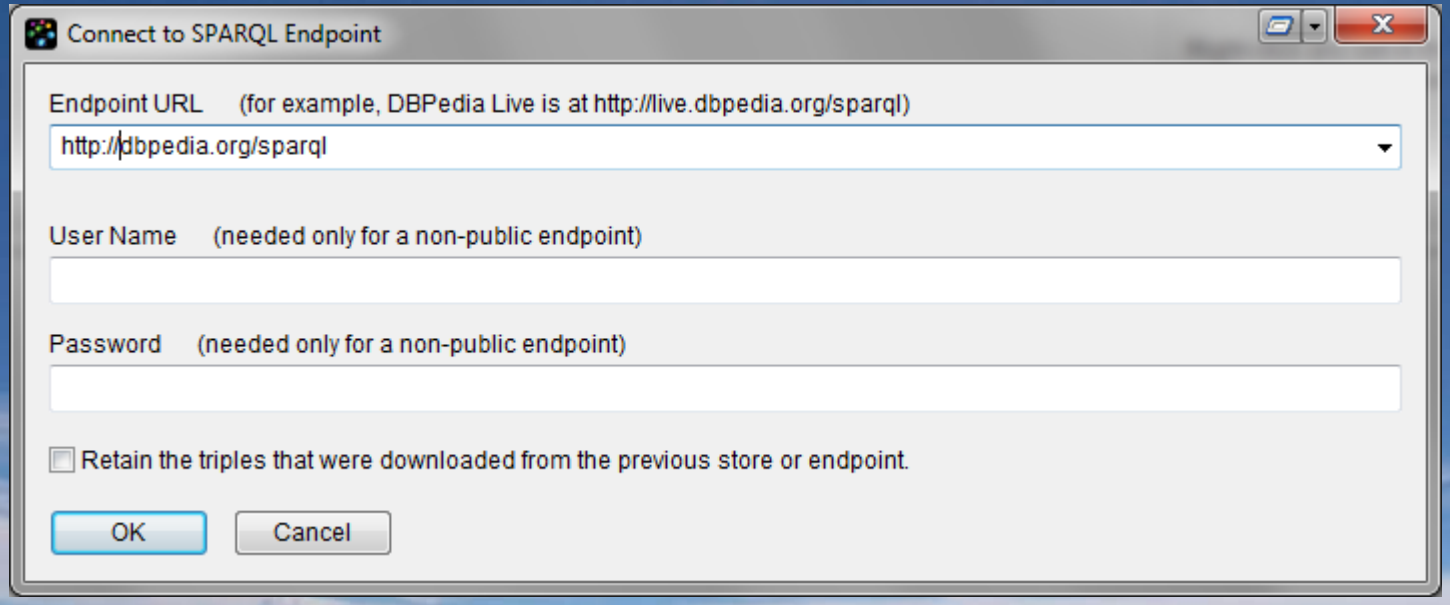
- Specify the EP



© @marianorico

Gruff

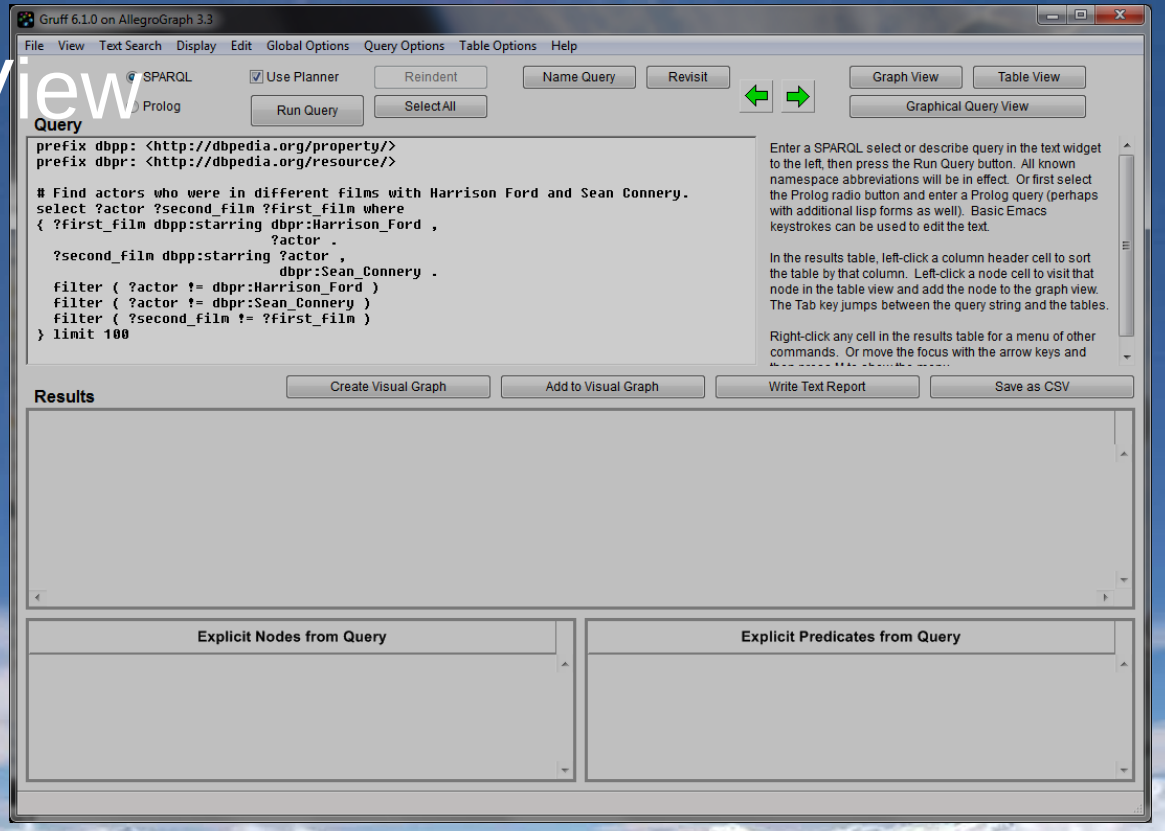
- Specify the EP



© @marianorico

Gruff

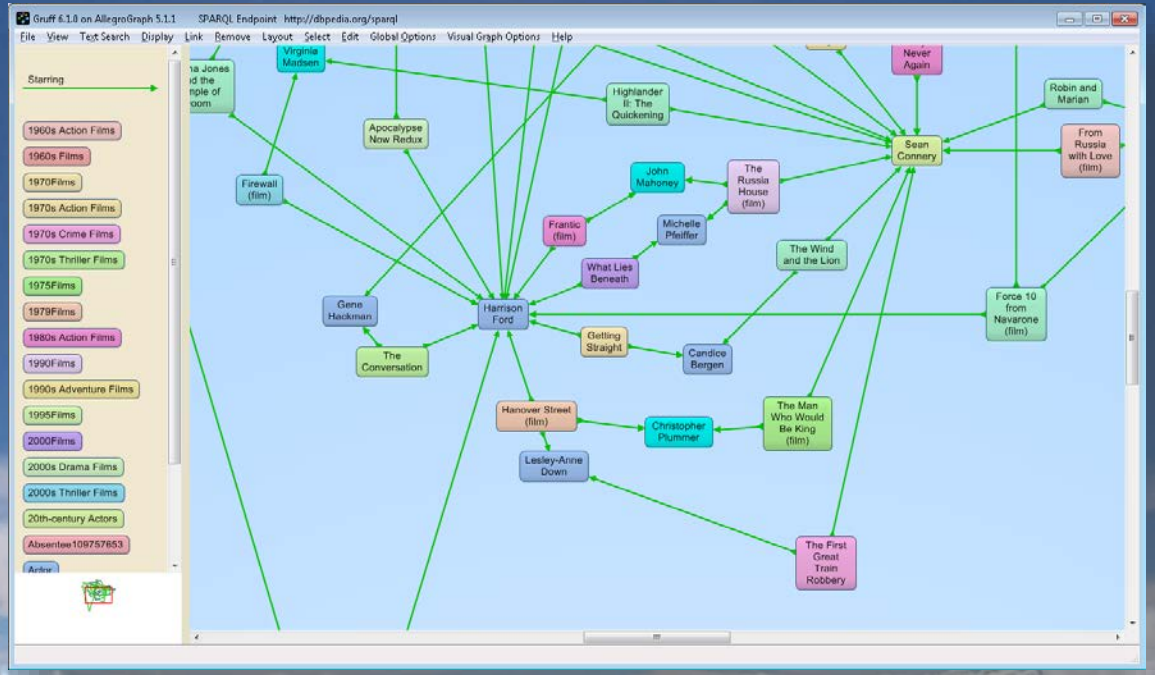
- View→Query View



© @marianorico

Gruff

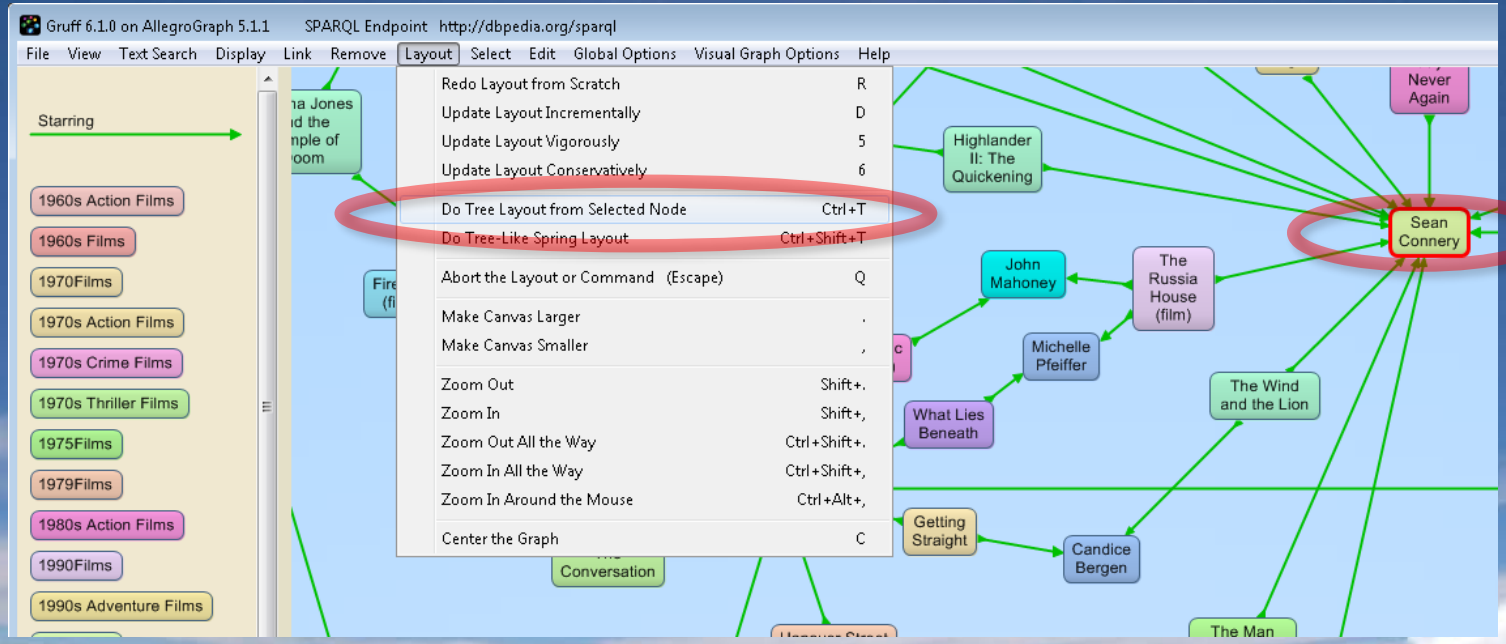
- SPARQL query results as a graph



© @marianorico

Gruff

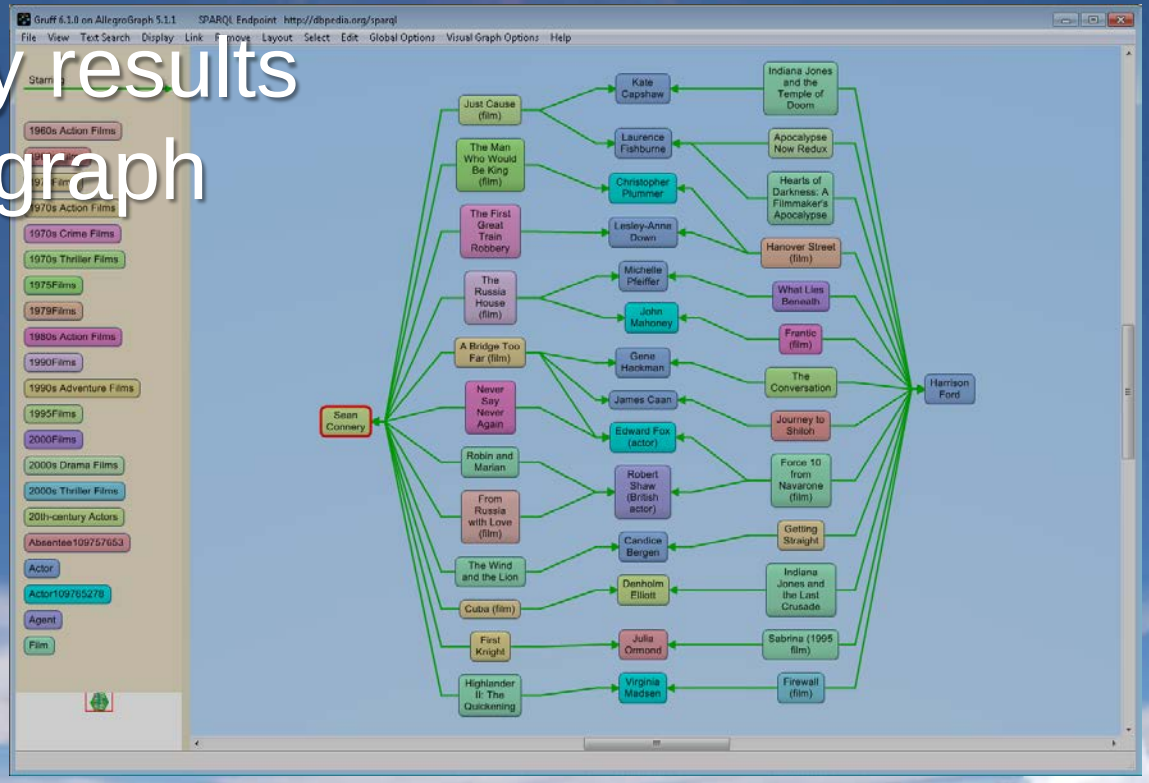
- SPARQL query results as a graph



© @marianorico

Gruff

- SPARQL query results as an ordered graph



© @marianorico

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 → Maximun Samples Triple to Display (e.g. dbpedia_2014.owl has 27.063 triples)

© @marianorico

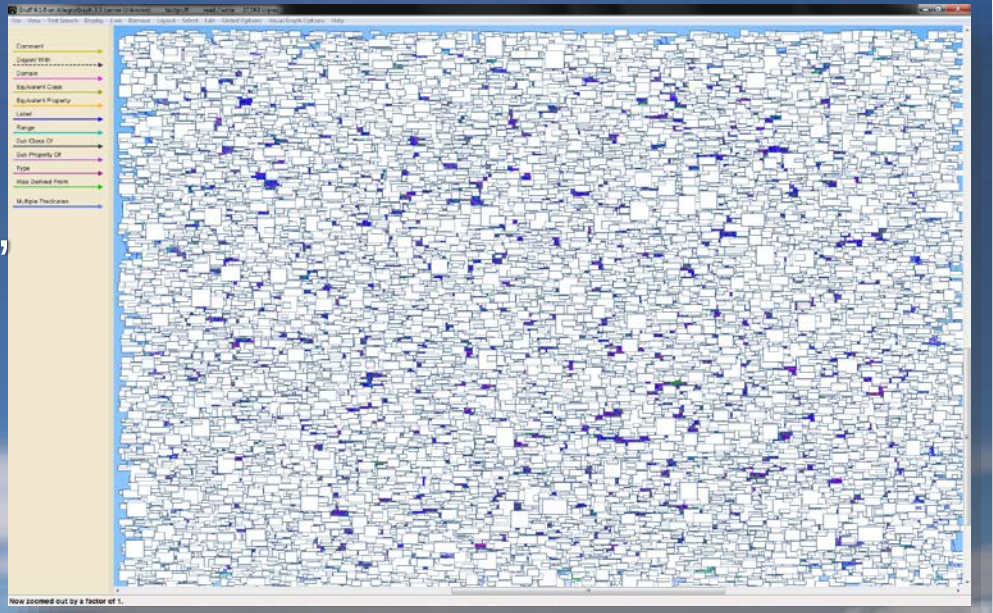
Gruff

- 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….

© @marianorico

Gruff

- Trying to visualize the DBpedia ontology
  - And…
  - Wait almost 1 **hour** ☹ later you get an almost "impossible to manage" graph ☹



© @marianorico

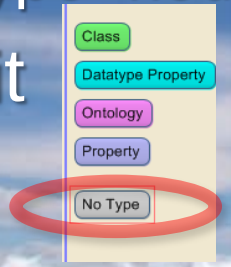
Gruff

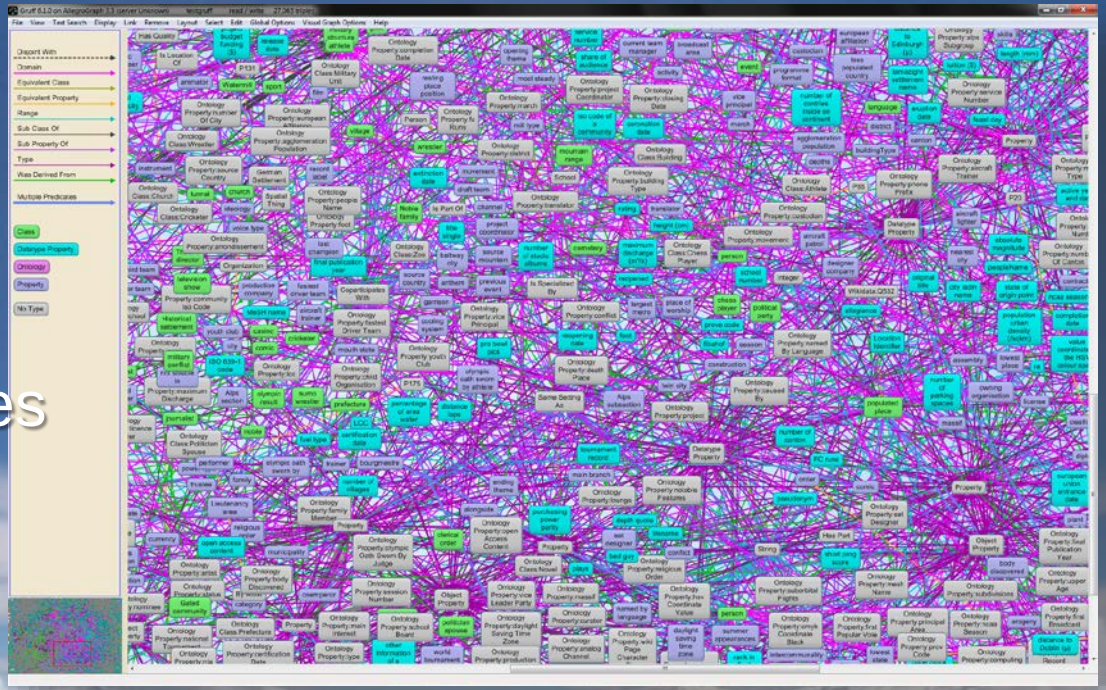
- Trying to visualize the DBpedia ontology
  - Zoom in
  - Remove again Literals
    - Wait for hours ☹ again
  - And…

© @marianorico

Gruff

- Trying to visualize the DBpedia ontology
  - And…
  - You get this
  - Keep removing
    - Now remove "No Type" nodes
  - And wait

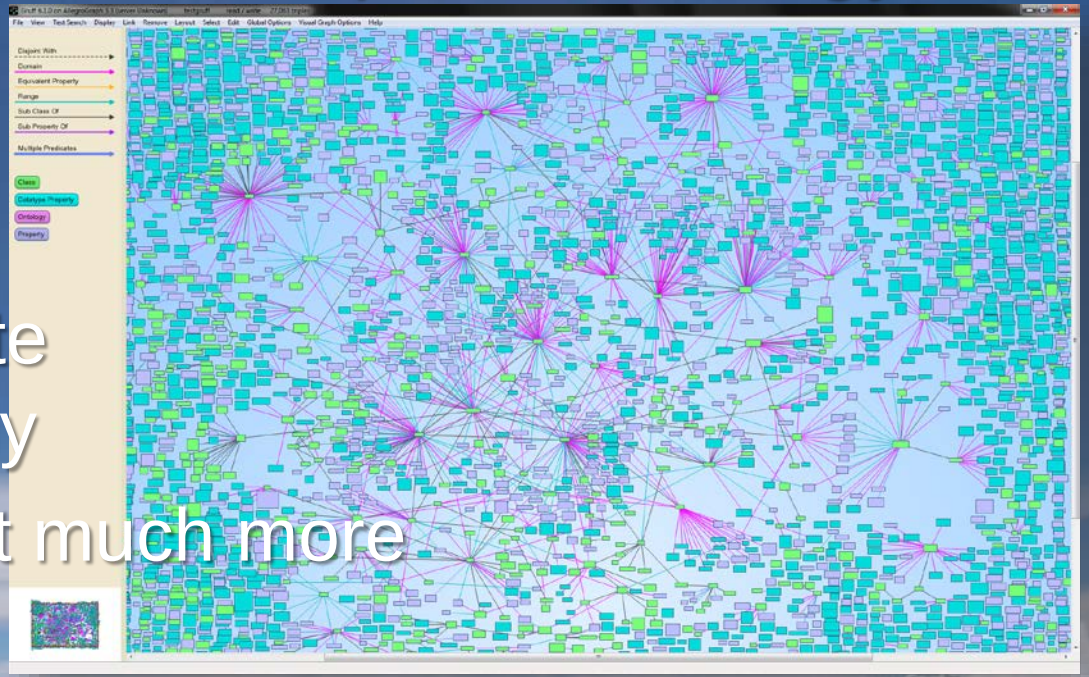




© @marianorico

Gruff

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



© @marianorico

Homework

- Your own proposals
- My proposals
  - Social graph (e.g. Facebook)
    - 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 in proportional to the number of triples using that property)

© @marianorico

Homework

- A written work made by up to 4 people. Describe clearly the work done by each author.
- Deadline: March 22nd

- Procedure
  - Upload it to Moodle and
  - Send an email to 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

#VizLinkedData

Thanks for your attention

Mariano.Rico@upm.es

Ontology Engineering Group