UNIVERSITÉ DE TECHNOLOGIE DE TROYES
CENTRE DE RECHERCHES ET D'ETUDES INTERDISCIPLINAIRES SUR LE DEVELOPPEMENT DURABLE (CREIDD)

Analysis and Visualization of Co-Authorship Network in Life Cycle Assessment Research Area: A case study of the International Journal of Life Cycle Assessment

by

SORIA MATEO, PABLO

28/02/2013

Supervisor:
Junbeum KIM, Ph.D. Assistant Professor
University of Technology of Troyes, France

1. Abstract

This document focuses on the methods to study the collaboration patterns in co-authorship networks. As well as in other fields, no studies of collaboration patterns have been done in industrial ecology field. In this document we will collect data from scientific publications in this field for and then analyze it. Since there are a large amount of data, we will create a program for their extraction and placement in a database. Database can be used to create graphs with Gephi and statistics with Excel using the necessary data. We base our analysis on a model of undirected weighted graph to represent a collaborative network and to extract from it various network parameters.
Results show that 53% of the graph forms a large cluster and the other 47% works in small communities and / or individually.

2. Summary

3. Introduction

As we move towards to a society in which no one works in an isolated way, it is essential to understand the systems and the way of how people and companies interact. These systems can be represented as networks in which the entities are nodes and the interactions between them are represented by edges. More generally, a node can be a neuron, an individual, a group, an organization, or even a country, while links can take the form of friendship, communication, collaboration, alliance or trade, to name only some of them.

Network theory has been developed considerably since it began in the 1950s. It has been used widely in electrical engineering, computer sciences, economics, social sciences, logistics and biology, where it has proven to be an increasingly powerful tool allowing the understanding of complex processes. In this document, we are using this network concept to analyze the relationship between authors of the same scientific field, such as industrial ecology.

Scientific publications are one of the most tangible and well documented forms of scientific collaboration. Many scientific collaboration networks have been extensively studied from different perspectives, such as degree analysis or centrality. However there is any study of the scientific collaboration network in industrial ecology research. For this reason, we will collect data from publications between 1996 and early 2013 from the journal "International Journal of Life Cycle Assessment" and analyze and visualize them using a network analysis software.

4. Background and related works

We can say that the study of social networks began in 1930 when the Hungarian writer Frigyes Karinthy proposed that there are five degrees of separation between any person in a short story called Chains. This would lead to the theory of six degrees of separation which consists in that any person is separated from another arbitrary person by six people on average, the six degrees of separation.

Later the Austrian mathematician Manfred Kochen and politician Ithiel de Sola Pool wrote a manuscript entitled mathematical Contacts and Influences. The manuscript postulated formally the mechanics of social networking, and explored mathematical consequences (including their degree of connectivity).

The manuscript left unresolved several significant questions about social networks, one of them is the number of degrees of separation in real social networks. Stanley Milgram took the baton leading the experiments reported in the article "The Small World Problem", in the popular science magazine Psychology Today.

Milgram's experiment was designed to measure the length between nodes, developing a method to count the number of nodes between two people.

Since then, the social network analysis has not stopped to grow and play an important role in many disciplines. The most recent study in the line of Milgram was made in 2011 when Facebook made a study called "Anatomy of Facebook" with all active users of its website at that time, 721 million members (about 10% of the world population) and was analyzed the set of mutual friends, to get the average of how many nodes there are between any pair of nodes.
From this test celebrities and famous people were excluded. The results showed that 99.6% of couple's users were connected by 5 degrees of separation. This is the closest test of the theory to today's date and gives an approximate result of 4.75 links.

4.1. Social network analysis

The social network analysis is based on the premise that relationships between entities can be described by a graph. The nodes in the graph represent entities, edges connect pairs of nodes, thus representing social interactions. This representation allows applying graph theory.
From the graph we can describe its properties in two levels: overall statistics of the graph and by individual properties of each node. Global indicators describe the characteristics of a social network as a whole, for example the network's diameter, the average distance's network, the number of components (fully connected sub graphs), groups, etc. The node properties characterize each node individually, for example, by the centrality, degree, length of the middle path we can see what nodes are better connected.

4.2. Co-authorship networks

Co-authorship networks have been widely used to determine the structure of scientific collaborations and the status of individual researchers. An example of co-authorship network is The Oracle of Bacon which determines the number of nodes between the actor Kevin Bacon and any other celebrity. The co-authorship network analysis has also been applied to various fields such as mathematics, neuroscience and information systems.

5. Collecting data

An important part for making possible this work has been to obtain the data. We have obtained the following information about each publication from March 1996 to January 2013:

- Title of publication
- Date
- Authors
    - Country of origin
    - City of origin
    - Organism they work for
- Keywords associated with the publication
- Volume of the journal
- Issue of the journal

Due to the large volume of information to be processed (1060 publications and 1901 authors) and that for in the future this task will be automatic, a program has been made in order to extract this information automatically from the website of the journal in question and keep it in a database. The advantage of working with a database is that we have all the information well-structured and without redundancies. This allows to extract then the data that we need for analyze the network and create statistics.

5.1. Technologies used

The program has been made in PHP due to prior knowledge of the author in the field and has been chosen the following tools for the gratuitous character thereof:

- WAMP, local server with:
    - Apache 2.4.2
    - Mysql 5.5.24
    - PHP 5.4.3
    - XDebug 2.1.2
    - XDC 1.5
    - PhpMyadmin 3.4.10.1
    - SQLBuddy 1.3.3
    - webGrind 1.0

All you need is to download the program: http://www.wampserver.com/en/ install and start the server. With this, we will have a local web server capable of interpreting PHP and a database server. We will also use the web interface of Mysql, PhpMyadmin to manage the database.

5.2. Design of the database

Designing the database is an important task because it will determine the speed, security and storage efficiency. We will explain the process without going into details:

1. Think about what information you want to keep.

2. Define the type of the data.

3. Establish what tables are necessary so the database will be normalized and will not contain redundant information. This step is the most elaborated and requires an understanding of relational databases such as MySQL.

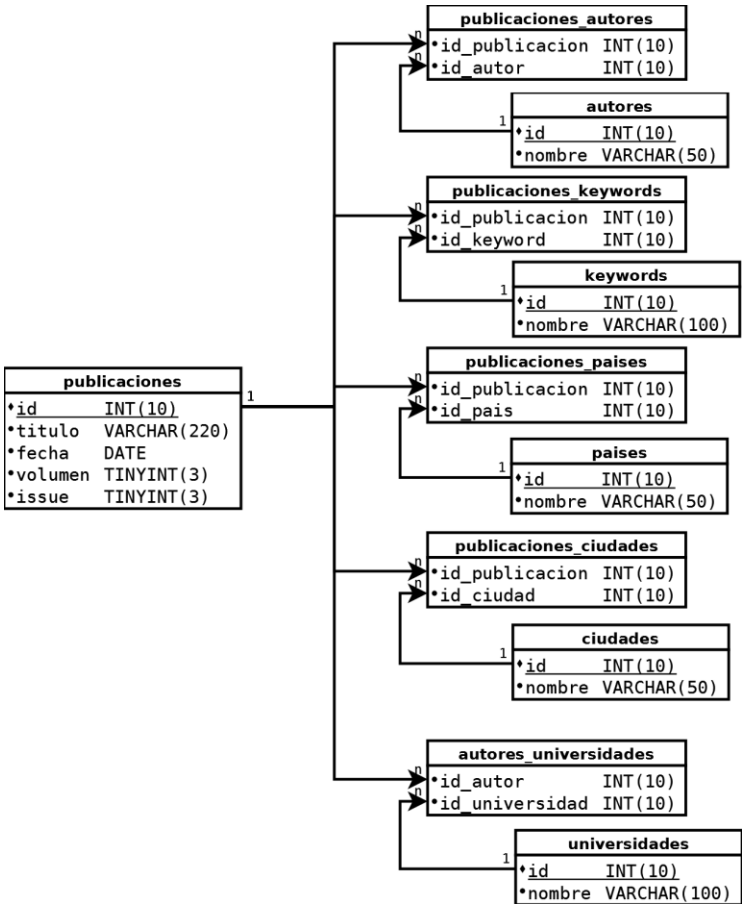
4. Establish the storage engine, it takes care of storing, managing and retrieving information. The most popular engines are MyISAM and InnoDB, we chose InnoDB although it would work as well with MyISAM.

5. Establish the encoding to use, in this case UTF8.

6. Establish the fields that will be indexes.

7. Write the code to create the database and tables or otherwise use any GUI like phpMyAdmin.

The diagram of the database has been drawn with the DIA program, which allows the creation of diagrams and is free. The code for creating the database is in the annexes. A diagram is shown below:



5.3. Algorithm

This is the most complex part of the work, the software programming that will process the information. Now we are going to explain how does the program works, you can find the complete code in the annexes.

Firstly, the url is provided where to start scanning:
$uIni = "http://link.springer.com/journal/volumesAndIssues/11367";
Secondly we define the section (piece of text) to be identified on the web. Download the web in a file called "file1.html" and begins to go through the text and find sections. After finding each one make a random waiting time between 3 and 5 seconds to not make too many requests in a short time to the magazine's server. This is done to keep caution, so connections are not interpreted as an attack on the server, basically it consists in simulating human behavior as far as requests per minute relates. With each section that has been found it gets a web address that sends it to the Prods function, which proceeds to download on the file "www2.html". New sections are defined,

one for finding paging and another one for finding the urls of the next level that we will access. Note that paging is repeated 2 times in the web. The particularity of this function is that when it finds paging number 2, it follows it and calls itself, it's a recursive function. So it goes through all the pages of that level. The next step is to process the pages from the next level, the Parse function does this, do not forget that after each request we continue waiting. With Parse function we can finally access to the publication itself, where are all the data we want: authors, date, keywords, country, etc. Sections are defined in order to obtain this data and it proceeds to download the web on the file “www3.html”. After that, the file is read searching the sections considering that in some cases, as dates, they can have more than one different section. This is due to the scanned website not always maintains the same scheme. It also detects whether the scanned website contains data or is private and not useful. Then the data obtained are processed to avoid problems of coding, each register is checked in order to see if they have the data that we want and then they proceed to integrate in the database and in comma separated files. CSV files at the end of this project have not been used because it is more practical to work only with the database. Add2File function saves all data in a CSV called backup.csv. Add2FileAutores function saves all possible combinations of two authors who are in a publication, this was done initially for working with Ghepi without the database, currently is not used. Finally AddDB function is responsible for checking if the registries already exist in the database and if not to add them.

5.4. About the coding

Data extracted from the web are in UTF8, so have they been treated in this way at all the times, do not forget to define the "charset" when connect to the database:
mysqli_set_charset($con, "utf8");
And set the tables and the collate of the connection to the server in the same type of coding.
These small errors can often produce you more than a headache.

6. Making the graph

6.1. Installing Gephi

Download the software on https://gephi.org/users/download/ and follow the installation steps. The version used in this document (0.8.2 beta) needs to update plugins so connection to the database will not fail.

6.2. Importing data to Gephi

Ghepi can import data in multiple formats:

- CSV
- GDF
- GEXF
- GraphML
- Pajek NET
- GML
- Tulip TLP
- Netdraw VNA
- Database

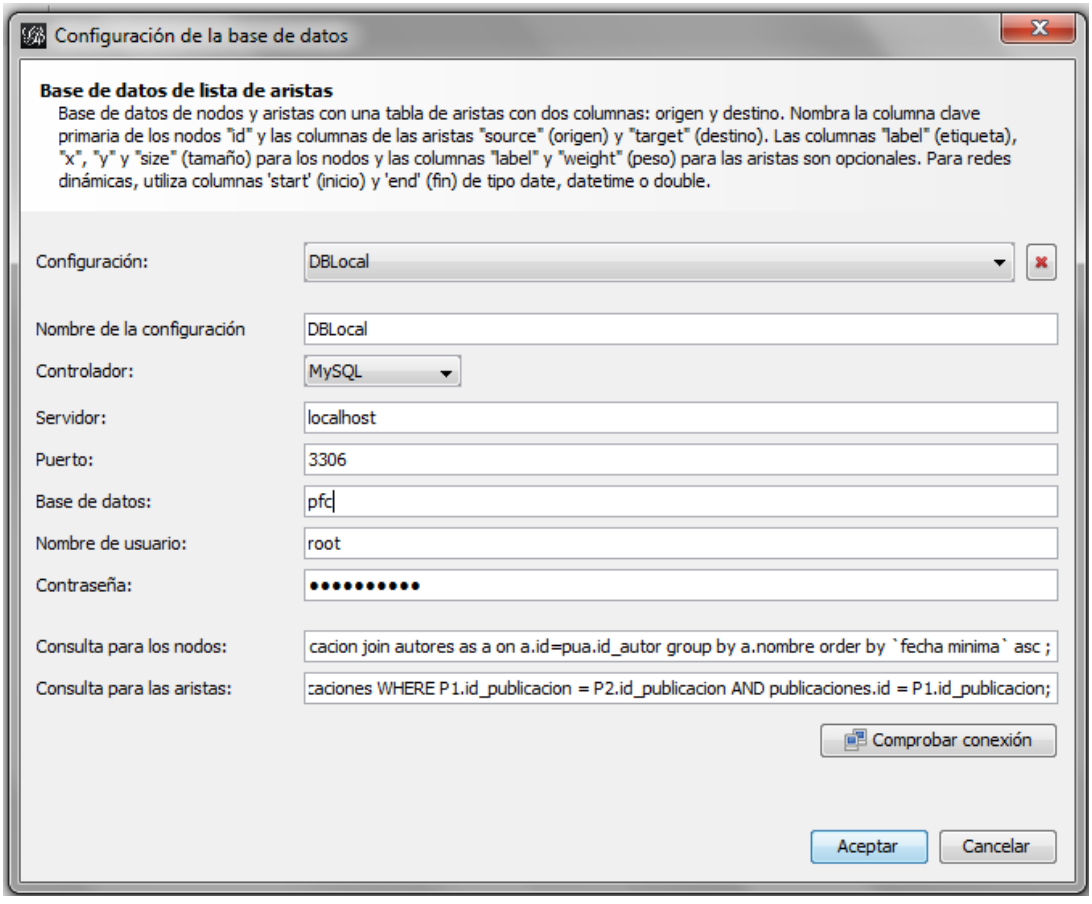
To obtain the data from a database, firstly we need to configure the type of the database, the address of the server, the server port, the name of the database, the user name and password and finally the queries for the nodes and edges, considering that we need to define the nodes with the "id" and edges as "source" and "target".
Thus the query to the nodes is:

```
SELECT a.id AS id,a.nombre AS label,MIN(pu.fecha) AS `fecha minima`
from publicaciones AS pu
JOIN publicaciones_autores AS pua
on pu.id=pua.id_publicacion
JOIN autores AS a
on a.id=pua.id_autor
group by a.nombre
order by `fecha minima` ASC;
```

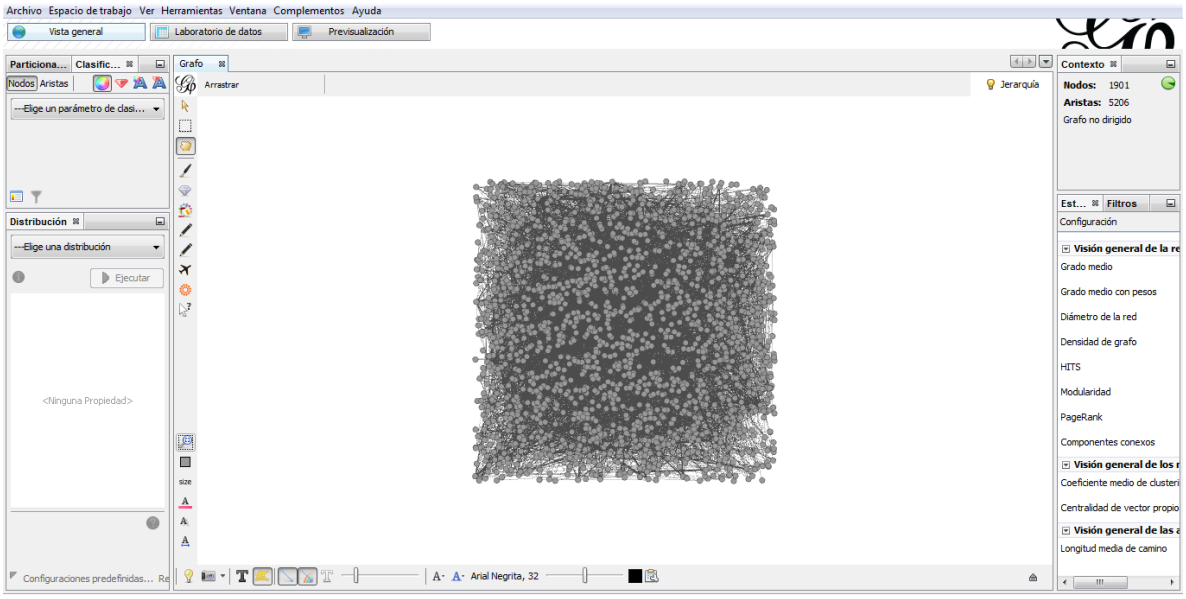
And the query to the edges is:

```
SELECT P1.id_autor as source, P2.id_autor as target,
publicaciones.fecha
FROM publicaciones_autores AS P1, publicaciones_autores AS P2,
publicaciones
WHERE P1.id_publicacion = P2.id_publicacion
AND publicaciones.id = P1.id_publicacion
AND P1.id_autor < P2.id_autor;
```

Should be noted that alongside the nodes's column also we extract the column of dates in which first appears each author. With the edges has been done the same, is also extracted the emerged date of the publication. This will help us later to see the evolution of the graph along the time.
In Ghepi looks like:



Then we choose "undirected" because it is an undirected graph and we should get the unprocessed graph:



6.3. Layout algorithms

Gephi has several algorithms for sorting the graph, most of them are based on physical systems in which interconnected nodes are attracted while isolates are repelled. It is explained below.

6.3.1. ForceAtlas

ForceAtlas is used to shape "small worlds" or scale-free networks, in other words, real data. The purpose of ForceAtlas is not the performance, but the quality. Some directed force algorithms seek the profitability (Yifan Hu, OpenOrd), while others seek quality (LinLog, ForceAtlas). ForceAtlas has been empirically developed to allow a rigorous interpretation of the graph and good readability, although it is slow. It has a complexity of $O(N^2)$, stands until 10.000 nodes and takes into consideration the weight of the edges.

6.3.2. Fruchterman-Reingold

It simulates the graph like a system of particles with mass. Nodes are the particles and the edges are springs that connect them. The algorithm tries to minimize the energy of the physical system. This has become a standard, but is very slow. It has a complexity of $O(N^2)$, stands until 1,000 nodes and does not take into consideration the weight of the edges.

6.3.3. Yifan Hu Multilevel

It's a very fast algorithm and with good quality for large graphs. It combines a directed force model with multilevel techniques to reduce complexity. It has a complexity of O(N*log(N)), stands until 100.000 nodes and does not take into consideration the weight of the edges. The algorithm stops automatically when it finishes.

6.3.4. OpenOrd

This algorithm hopes weighted undirected graphs and aims to distinguish the groups (clusters). This algorithm is based on Fruchterman-Reingold and operates with a fixed number of iterations. The long edges are cut to permit separation of the clusters. Has a complexity of O(N*log(N)), stands until 1.000.000 nodes and takes into consideration the weight of the edges. The algorithm stops automatically when it finishes.

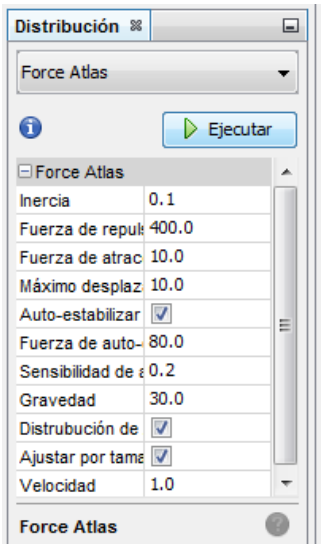
6.3.5. ForceAtlas2

It is an improved version of ForceAtlas to manipulate large graphs maintaining a good quality. The repulsion of the nodes is estimated by calculation of Barnes-Hut, this reduces the complexity of the algorithm. Has a complexity of O(N*log(N)), stands until 1.000.000 nodes and takes into consideration the weight of the edges.

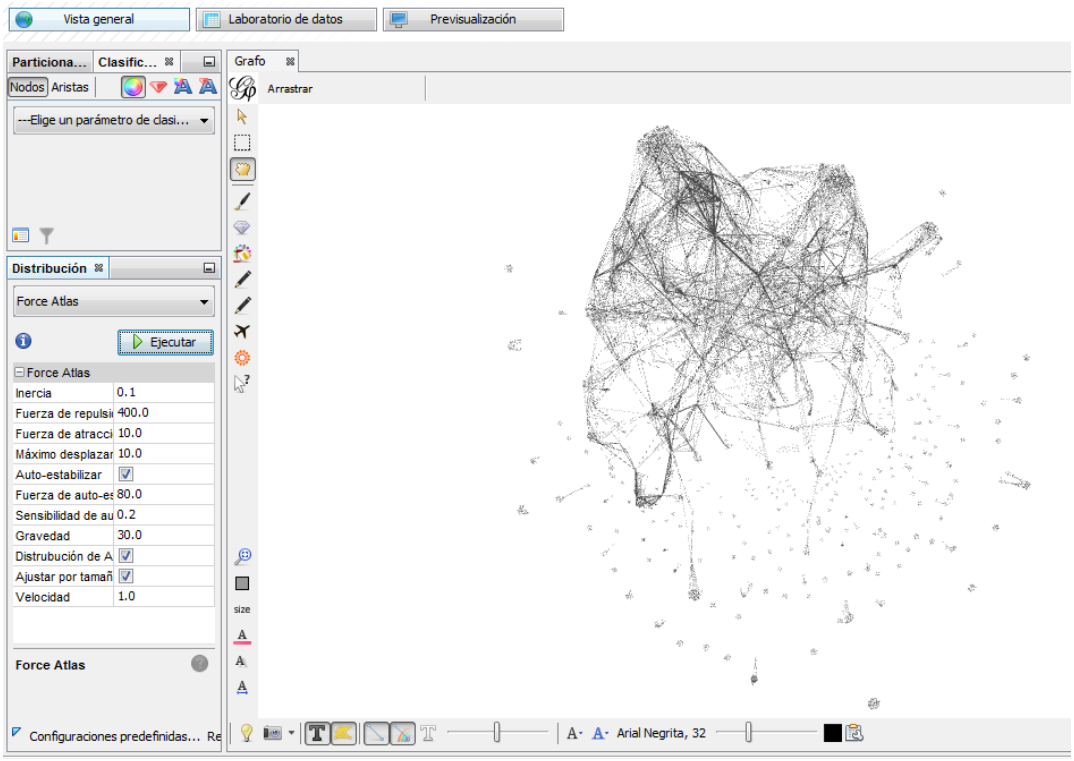
6.3.6. GeoLayout

The algorithm uses the latitude and longitude coordinates to establish the position of the nodes in the network. It has a complexity of O(N) and stands until 1.000.000 nodes.

To choose the appropriate algorithm we can consider that OpenOrd emphasizes clusters or communities. ForceAtlas, Yifan Hu and Fruchterman-Reingold emphasizes complementarities and GeoLayout emphasizes geographical distribution. In this work we have used ForceAtlas because it fits with our requirements.
We configure the algorithm parameters and execute:



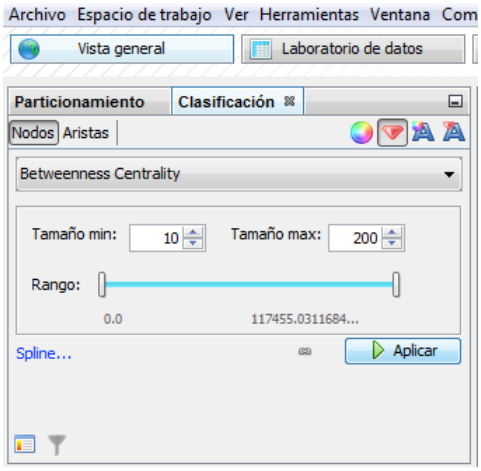
We let the algorithm ForceAtlas works and when it is finished we will get something like this:



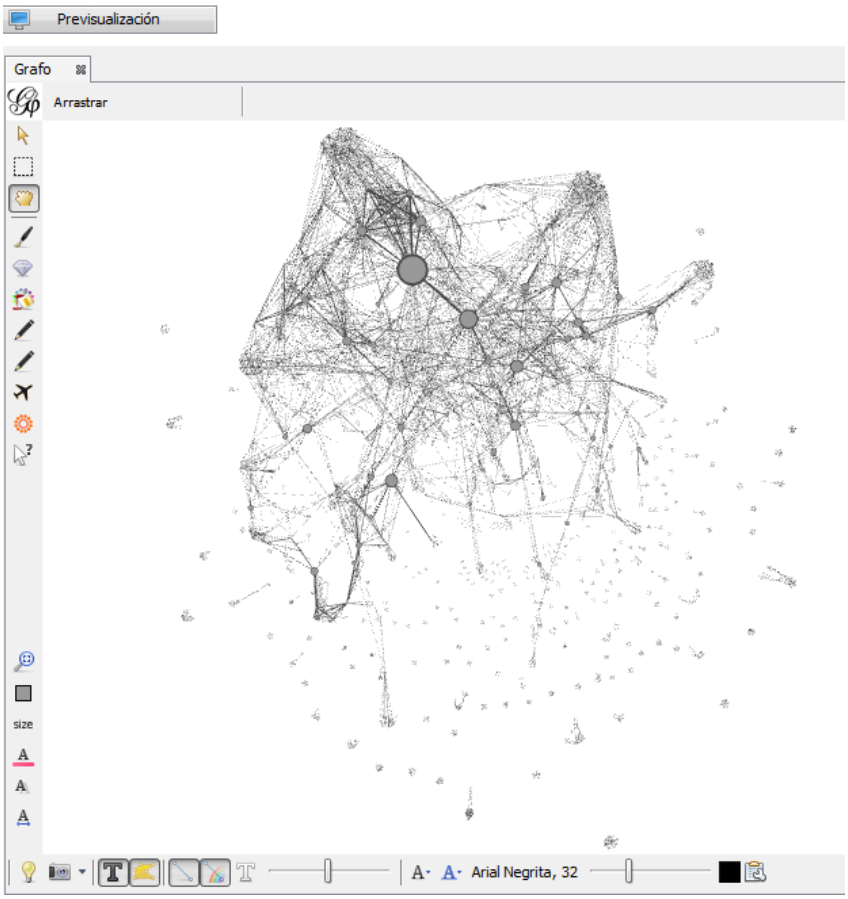
6.4. Color, size and communities

The next step is to provide size and color to the nodes and edges according to their properties. We calculate the network diameter, with this we get three measures: Betweenness Centrality, Closeness Centrality and Eccentricity those which will be explained later. We will assign node size according to Betweenness Centrality.

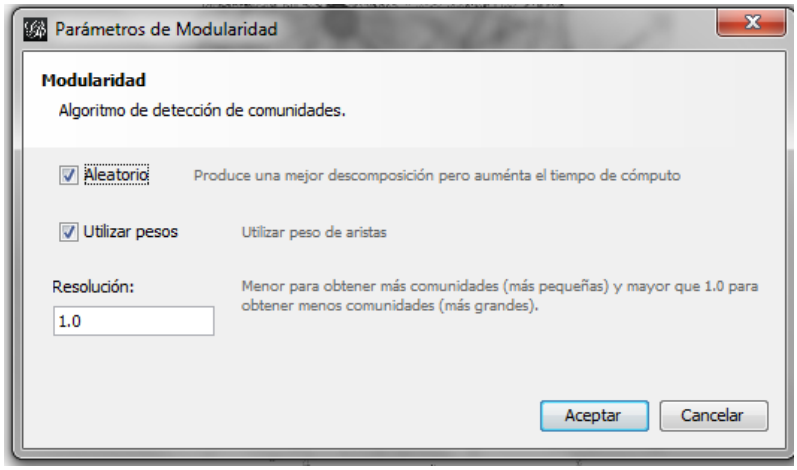
We apply and re-run ForceAtlas.



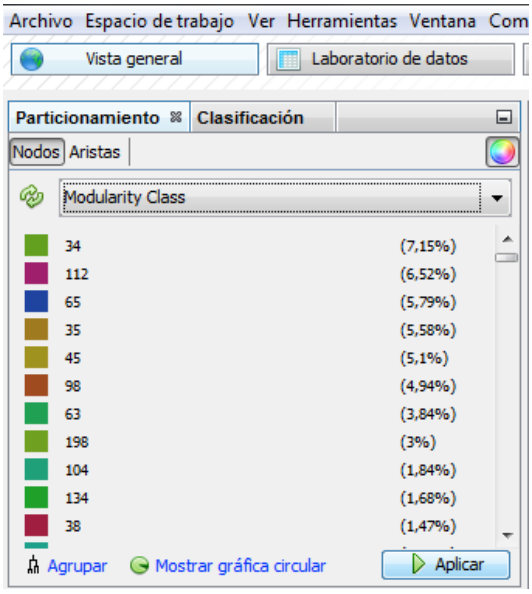
We obtain the following distribution in the space:



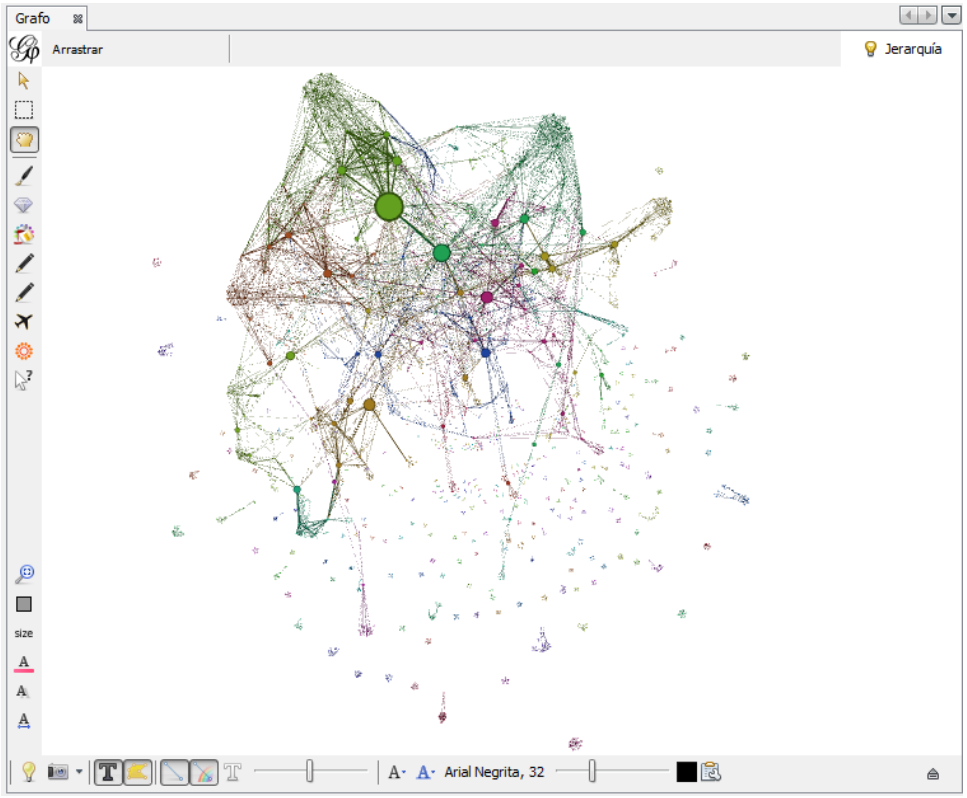
To detect the communities we calculate the modularity:



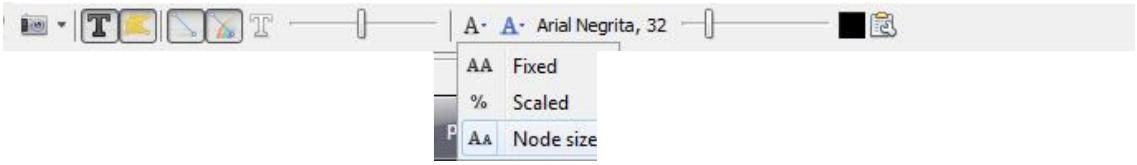
With this we get the property "Modularity Class" and we proceed to partitioning with it:



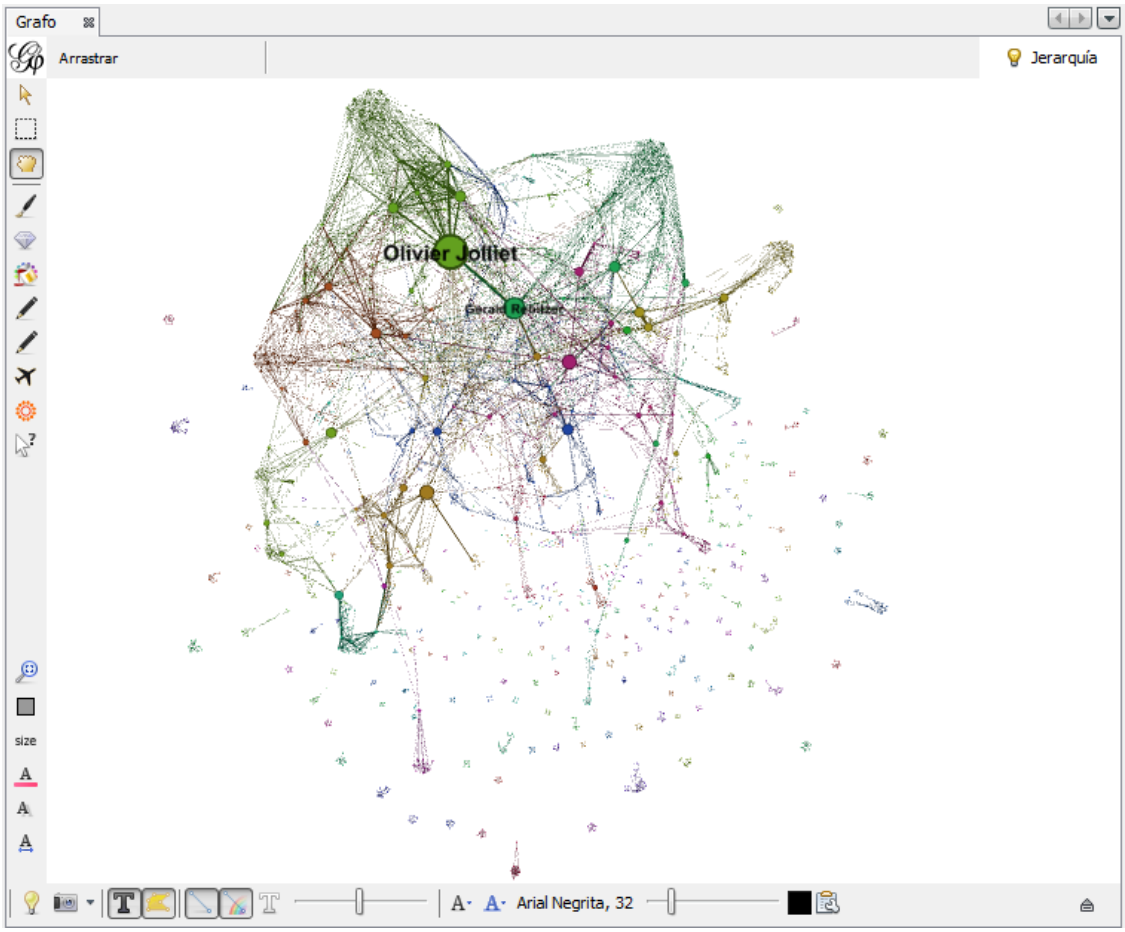
With this we get colored communities in different tones:



We activate the labels on each node and we make them proportional to the size of the node:

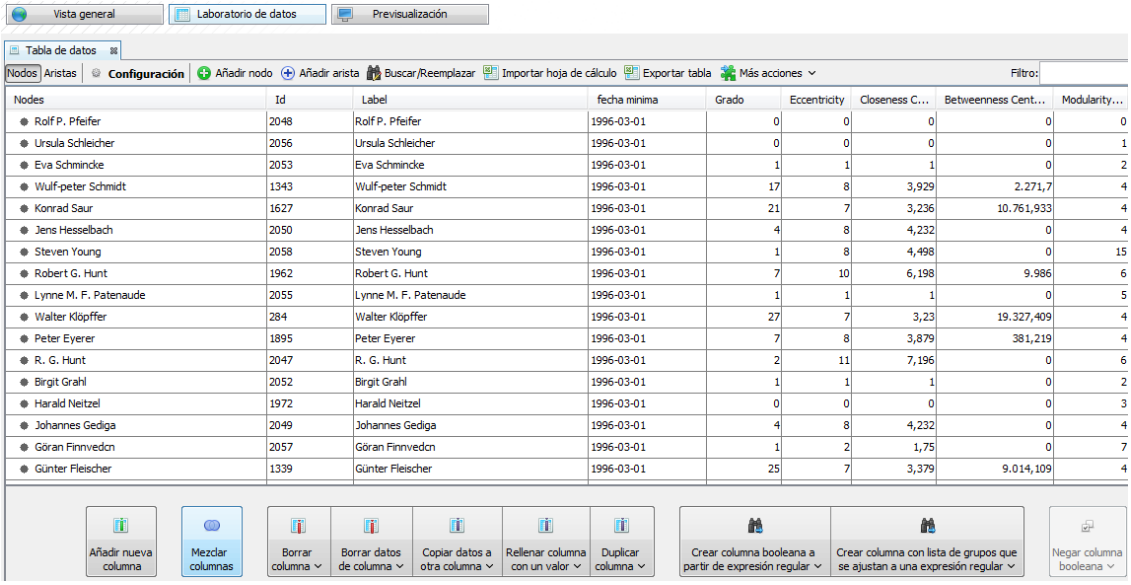


The result until now is the following and is composed of 296 communities:



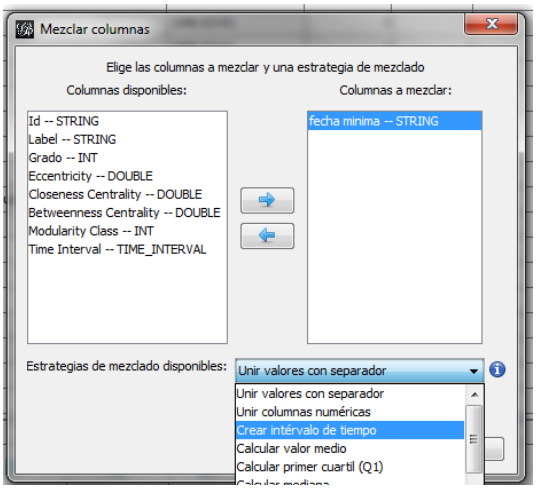
6.5. Timeline

To study the evolution of the graph is necessary to create a dynamic graph, to do this we need to go to the data laboratory and create a type of column "Time Interval" for both nodes and edges. Let's start with the nodes, in the nodes data table we select ‘merge columns’:

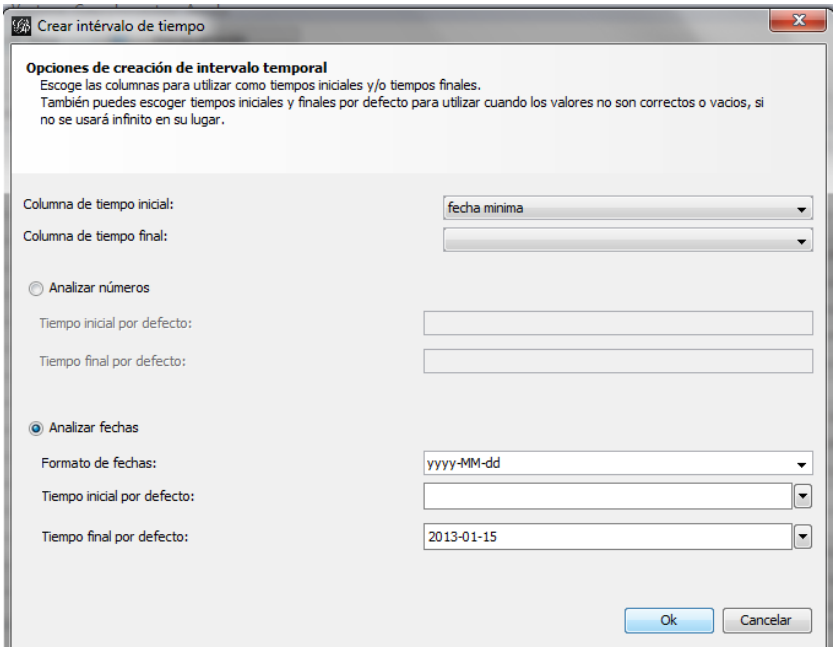


Nodes	Id	Label	fecha minima	Grado	Eccentricity	Closeness C...	Betweenness Cent...	Modularity...
Rolf P. Pfeifer	2048	Rolf P. Pfeifer	1996-03-01	0	0	0	0	0
Ursula Schleicher	2056	Ursula Schleicher	1996-03-01	0	0	0	0	1
Eva Schmincke	2053	Eva Schmincke	1996-03-01	1	1	1	0	2
Wulf-peter Schmidt	1343	Wulf-peter Schmidt	1996-03-01	17	8	3,929	2.271,7	4
Konrad Saur	1627	Konrad Saur	1996-03-01	21	7	3,236	10.761,933	4
Jens Hesselbach	2050	Jens Hesselbach	1996-03-01	4	8	4,232	0	4
Steven Young	2058	Steven Young	1996-03-01	1	8	4,498	0	15
Robert G. Hunt	1962	Robert G. Hunt	1996-03-01	7	10	6,198	9.986	6
Lynne M. F. Patenaude	2055	Lynne M. F. Patenaude	1996-03-01	1	1	1	0	5
Walter Klöpffer	284	Walter Klöpffer	1996-03-01	27	7	3,23	19.327,409	4
Peter Eyerer	1895	Peter Eyerer	1996-03-01	7	8	3,879	381,219	4
R. G. Hunt	2047	R. G. Hunt	1996-03-01	2	11	7,196	0	6
Birgit Grahl	2052	Birgit Grahl	1996-03-01	1	1	1	0	2
Harald Neitzel	1972	Harald Neitzel	1996-03-01	0	0	0	0	3
Johannes Gediga	2049	Johannes Gediga	1996-03-01	4	8	4,232	0	4
Göran Finnvedcn	2057	Göran Finnvedcn	1996-03-01	1	2	1,75	0	7
Günter Fleischer	1339	Günter Fleischer	1996-03-01	25	7	3,379	9.014,109	4

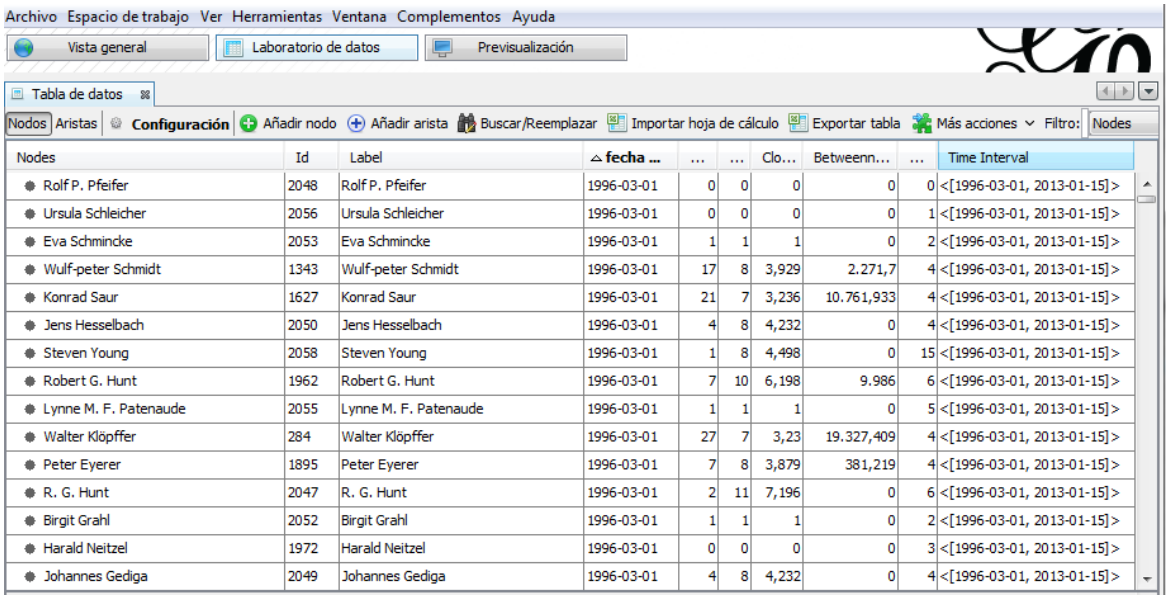
In the new window we select the minimum date column and we chose create a time interval. This will create for each node a date of appearance and disappearance. In the case of the nodes, these will appear when the author of that node will write and publish his first article. Once created, the node will stay until the end, that means that we accumulate new and existing publications.



We set the format in which the date has to be recognized and we establish by default a final date, in this case 2013-01-15 because extracted publications belong up to that date.

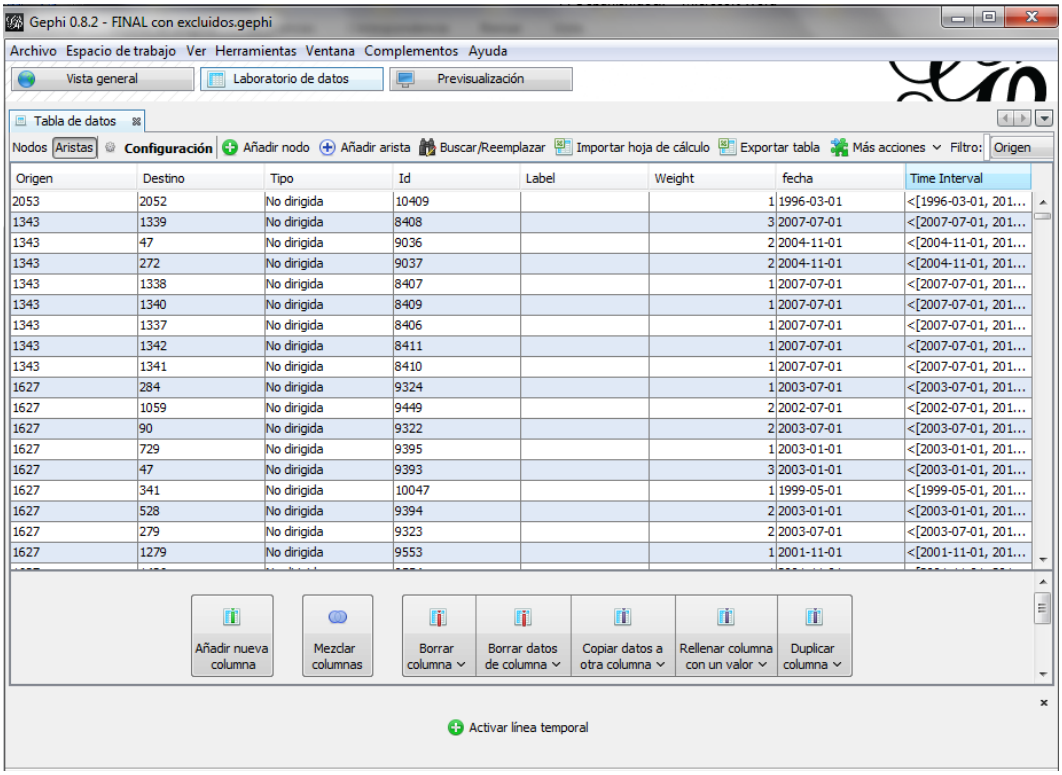


Now there should be a new column called "Time Interval" which will help us to create the dynamic graph:



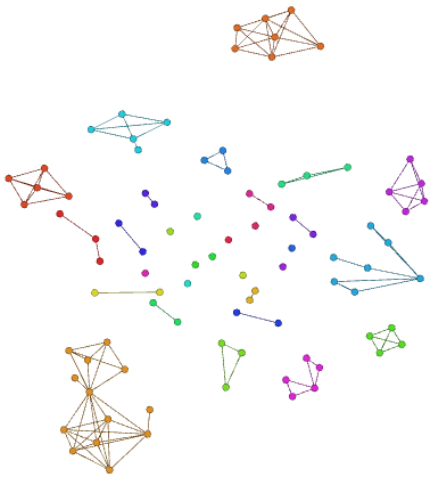
Nodes	Id	Label	fecha ...	...	...	Clo...	Betweenn...	...	Time Interval
Rolf P. Pfeifer	2048	Rolf P. Pfeifer	1996-03-01	0	0	0	0	0	<[1996-03-01, 2013-01-15]>
Ursula Schleicher	2056	Ursula Schleicher	1996-03-01	0	0	0	0	1	<[1996-03-01, 2013-01-15]>
Eva Schmincke	2053	Eva Schmincke	1996-03-01	1	1	1	0	2	<[1996-03-01, 2013-01-15]>
Wulf-peter Schmidt	1343	Wulf-peter Schmidt	1996-03-01	17	8	3,929	2.271,7	4	<[1996-03-01, 2013-01-15]>
Konrad Saur	1627	Konrad Saur	1996-03-01	21	7	3,236	10.761,933	4	<[1996-03-01, 2013-01-15]>
Jens Hesselbach	2050	Jens Hesselbach	1996-03-01	4	8	4,232	0	4	<[1996-03-01, 2013-01-15]>
Steven Young	2058	Steven Young	1996-03-01	1	8	4,498	0	15	<[1996-03-01, 2013-01-15]>
Robert G. Hunt	1962	Robert G. Hunt	1996-03-01	7	10	6,198	9.986	6	<[1996-03-01, 2013-01-15]>
Lynne M. F. Patenaude	2055	Lynne M. F. Patenaude	1996-03-01	1	1	1	0	5	<[1996-03-01, 2013-01-15]>
Walter Klöpffer	284	Walter Klöpffer	1996-03-01	27	7	3,23	19.327,409	4	<[1996-03-01, 2013-01-15]>
Peter Eyerer	1895	Peter Eyerer	1996-03-01	7	8	3,879	381,219	4	<[1996-03-01, 2013-01-15]>
R. G. Hunt	2047	R. G. Hunt	1996-03-01	2	11	7,196	0	6	<[1996-03-01, 2013-01-15]>
Birgit Grahl	2052	Birgit Grahl	1996-03-01	1	1	1	0	2	<[1996-03-01, 2013-01-15]>
Harald Neitzel	1972	Harald Neitzel	1996-03-01	0	0	0	0	3	<[1996-03-01, 2013-01-15]>
Johannes Gediga	2049	Johannes Gediga	1996-03-01	4	8	4,232	0	4	<[1996-03-01, 2013-01-15]>

We proceed to do the same in the edges table, we will get a new column as in the picture:

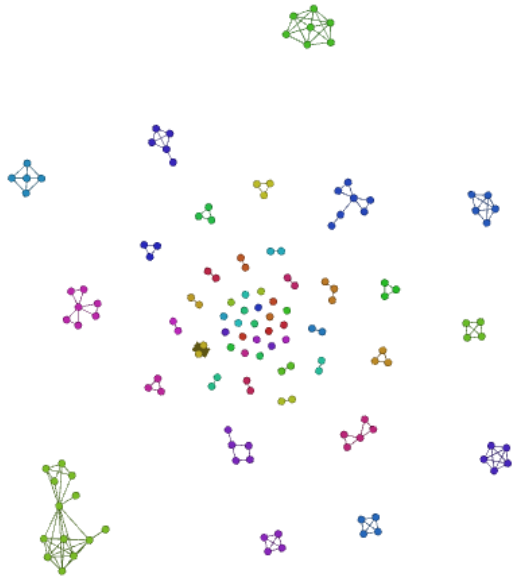


Then, we activate the timeline and go to overview, we establish a time frame about a year long. Now we can press play and observe the evolution.

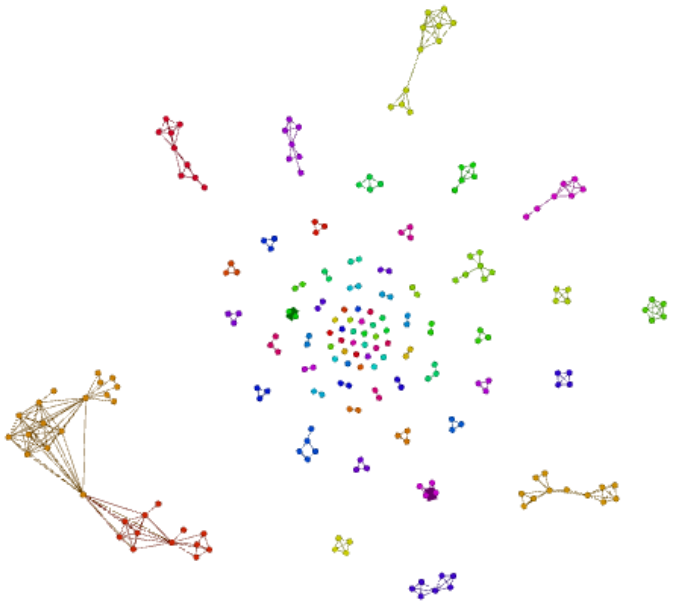
In 1997 we have:



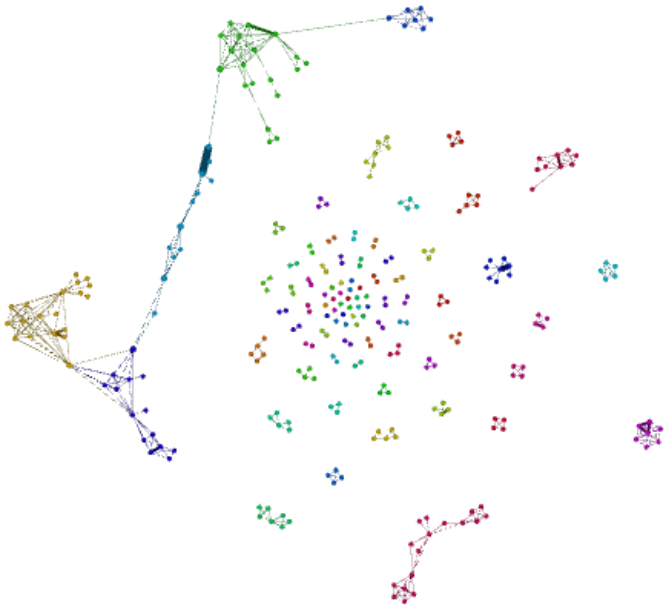
In 1998 we have:



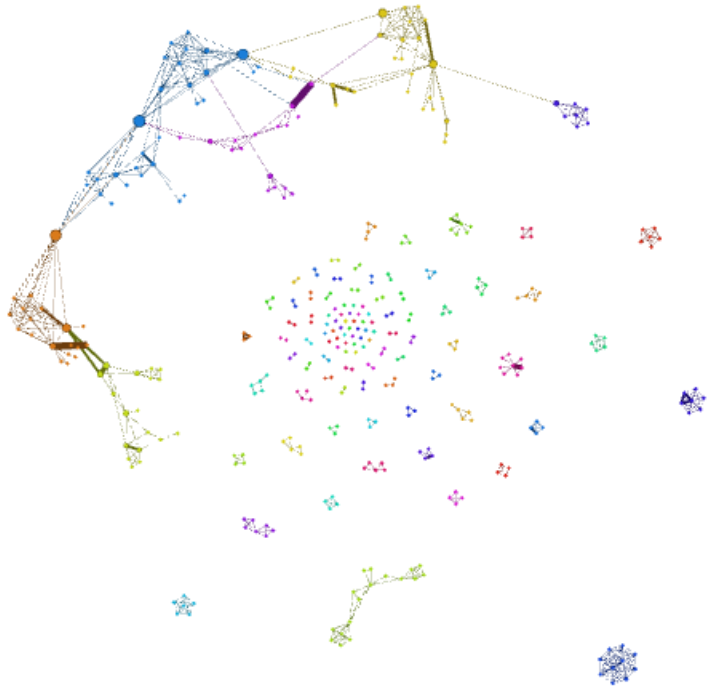
In 1999 we have:



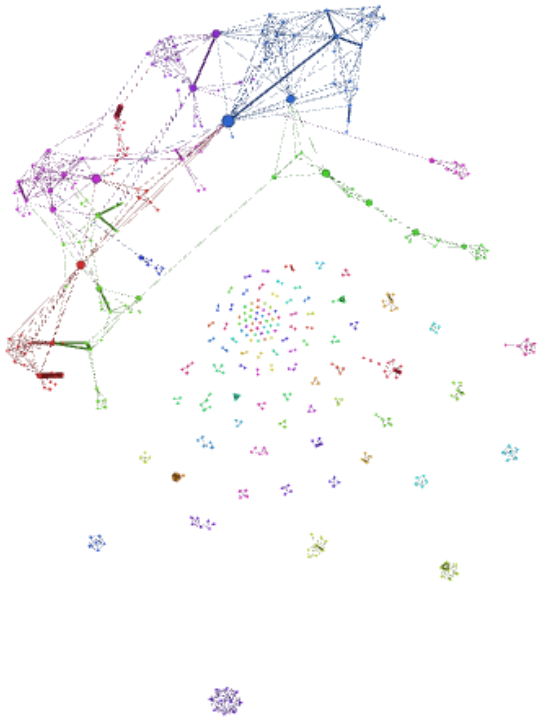
In 2000 we have:



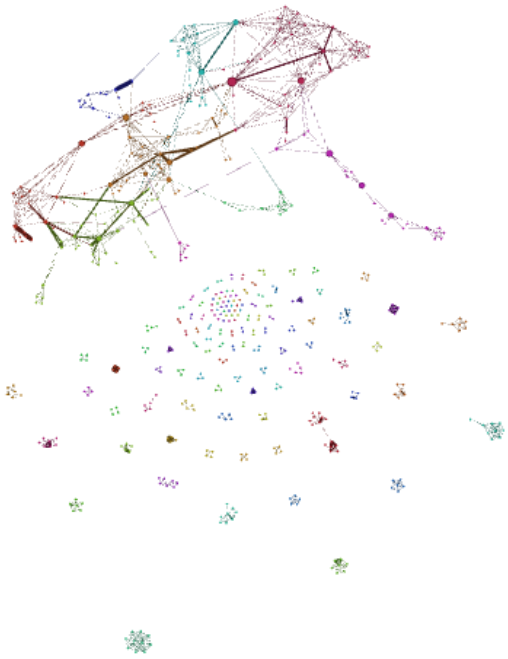
In 2001 we have:



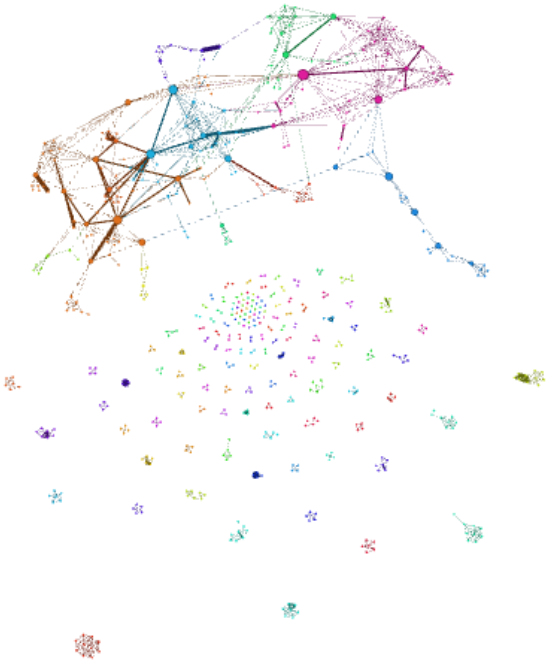
In 2002 we have:



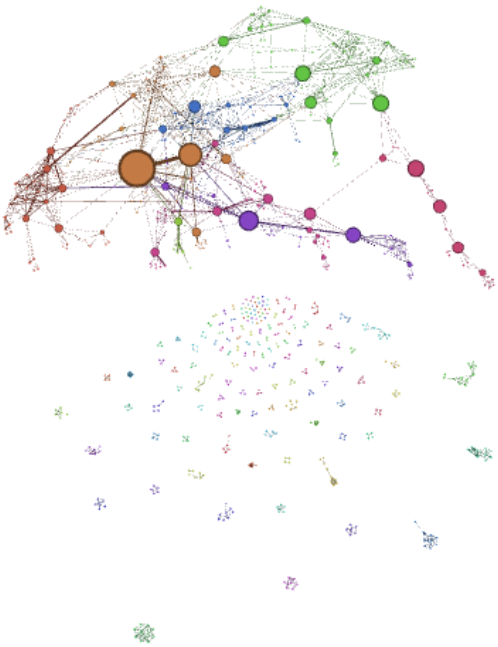
In 2003 we have:



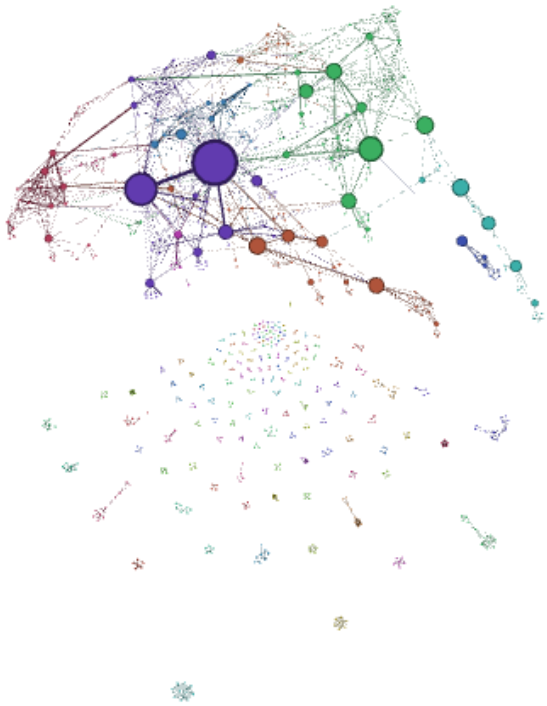
In 2004 we have:



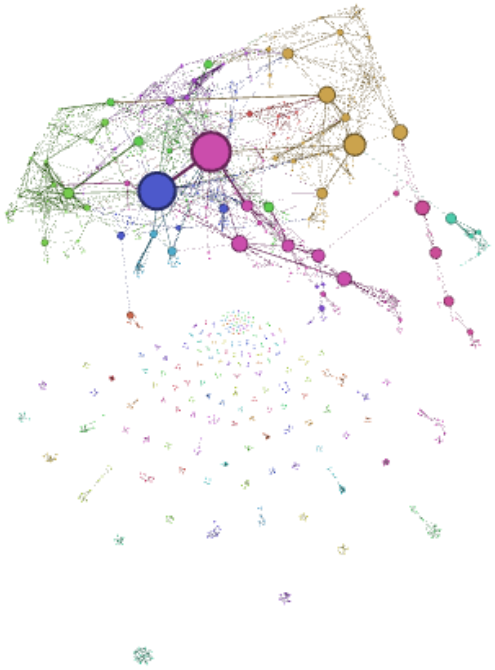
In 2005 we have:



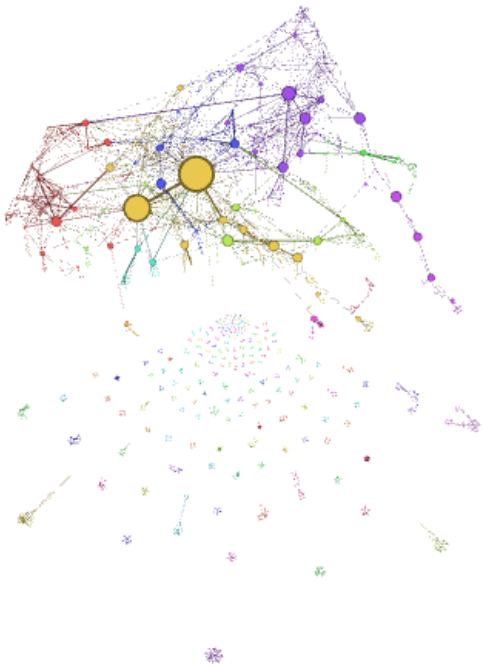
In 2006 we have:



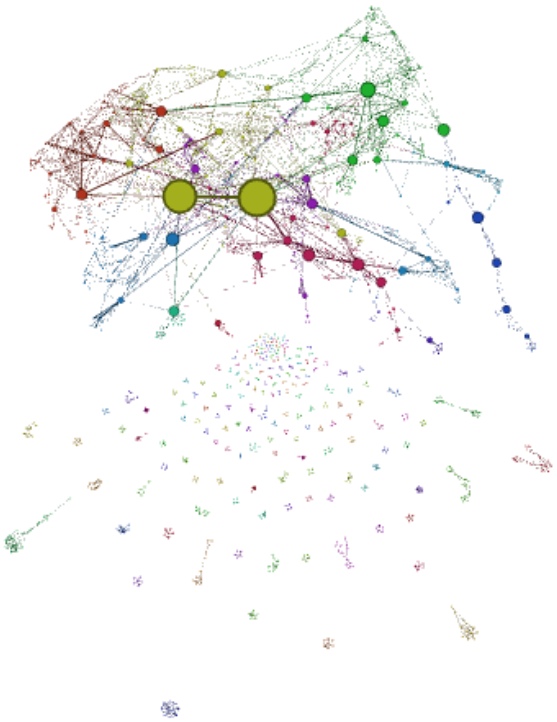
In 2007 we have:



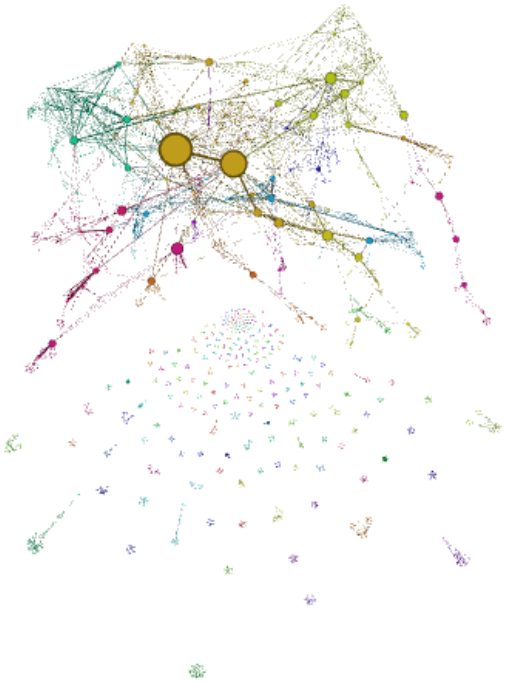
In 2008 we have:



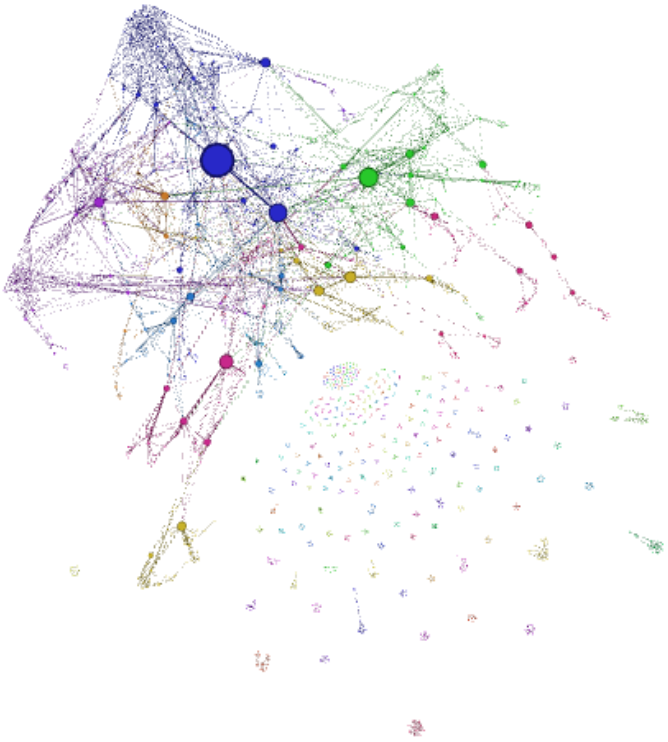
In 2009 we have:



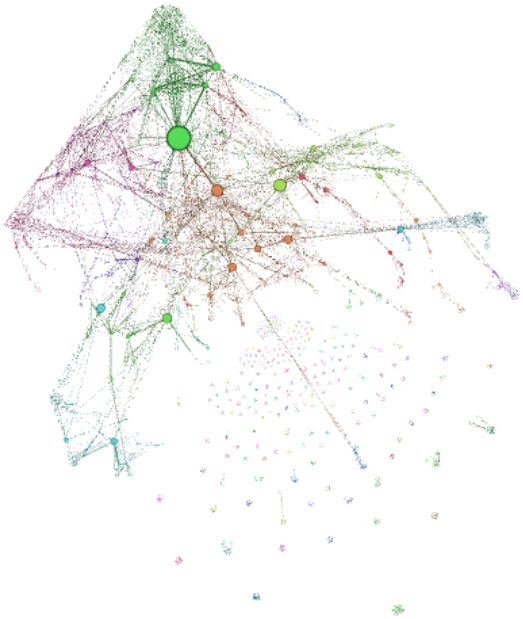
In 2010 we have:



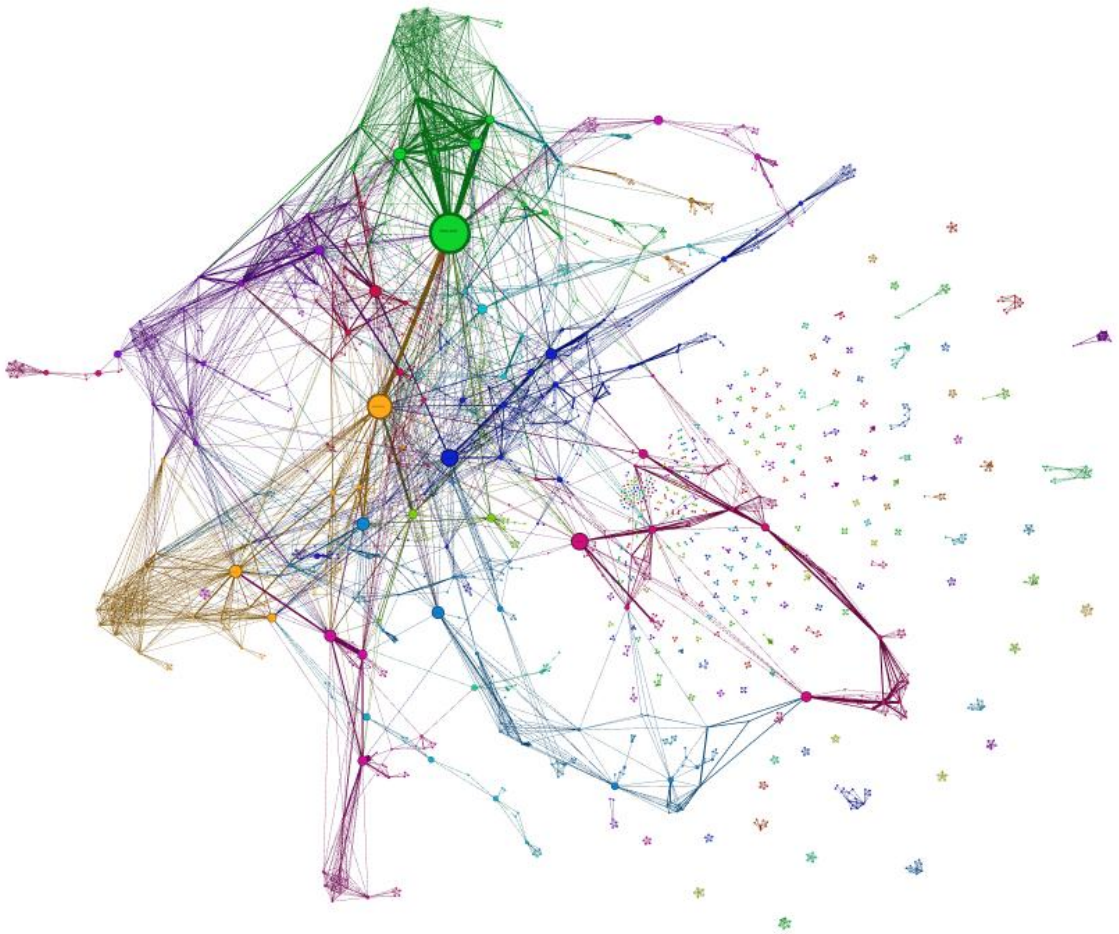
In 2011 we have:



In 2012 we have:

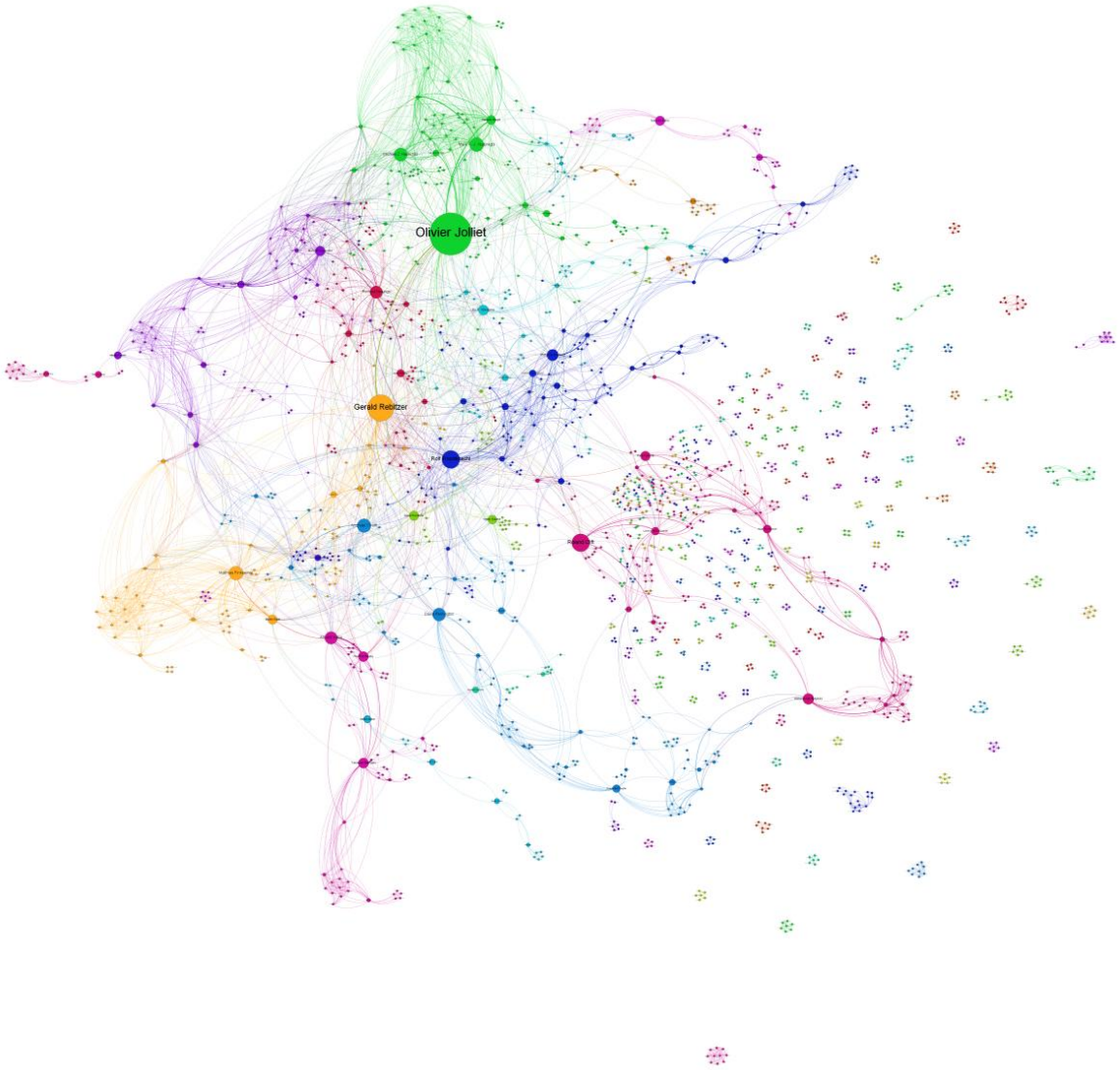


And finally the complete graph on 2013:



6.6. Preview and exportation

In the overview we select the algorithm "Label Adjust", we execute and so we avoid that author's names to overlap. Then go to the tab "Preview" and configure it to display the labels on each node, the font size and proportionality to the size of the node. The final graph is:



To explore the graph without quality loss, this can be exported in resizable vector graphic formats as SVG and PDF.

7. Studying the graph

7.1. Previous concepts

A graph *G* is an ordered pair *G = (V, E)* where *V* is a set of nodes and *E* is a set of edges that connect these nodes. Typically *V* is usually finite. Many important results on graphs are not applicable for infinite graphs. In graph theory and network analysis there are several measures that allow us to characterize the network, both nodes individually and overall network level. The most used are the following.

7.2. Centrality

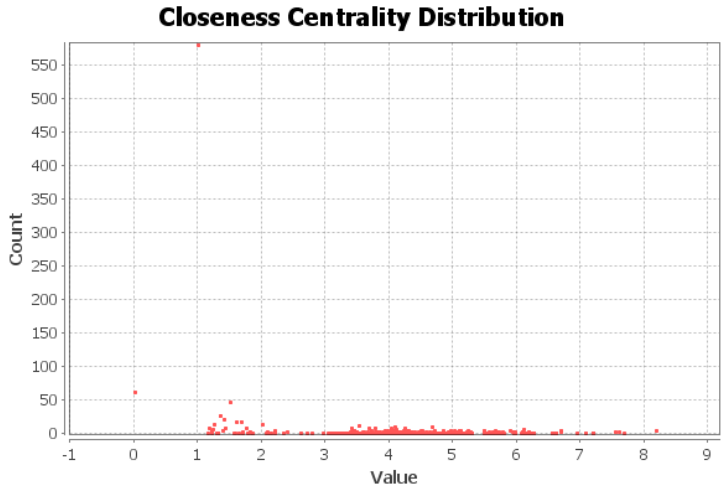
The centrality measures according to a criterion the contribution or importance of a node within the network according to its location. There are two types of centrality, radial measures and medial measures.
The radials take as their reference point a given node that initiates or terminates routes on the network and medials take as their reference the paths that pass through a given node.
The radial measures can be classified in volume and length. Volume measures, measure the number of routes limited to a predetermined length. Length measures measure the length of routes needed to reach a predefined number of routes.

7.2.1. Closeness centrality

It is a measure of radial length and it is calculated as the sum or average of the shortest distances from one node to all the others. Therefore the most central node is the one with a lower closeness centrality. It can be interpreted as a measure of the speed with which the information is transmitted from one node to all the others. The definition of closeness centrality that allows analyzing disconnected graphs mathematically is calculated as:

$$C_{CLO}(i) = \sum_{j=1}^{n} 2^{-(S)_{ij}}$$

Where $(S)_{ij}$ is the matrix of distances of the network, that is to say, that matrix whose elements (i, j) correspond to the shortest distance from the node i until the node j.

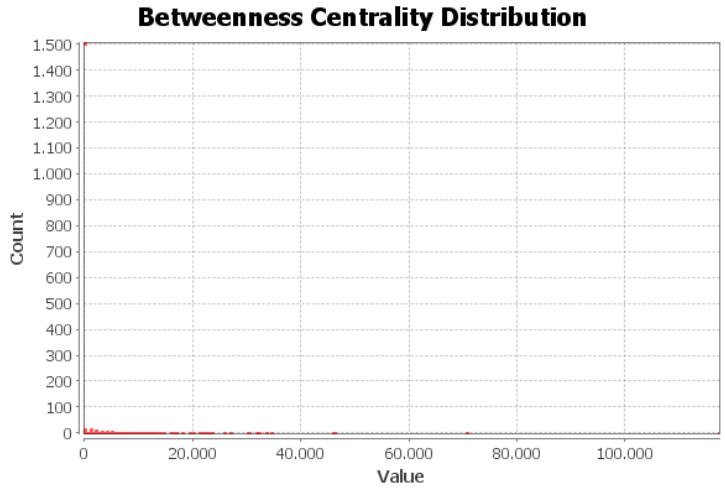


7.2.2. Betweenness centrality

It is a medial measure that quantifies the frequency or the number of times that a node acts as a bridge along the shortest path between another pair of nodes. It is defined as:

$$C_{BET}(i) = \sum_{j,k} \frac{b_{jik}}{b_{jk}}$$

Where b_{jk} is the number of shortest paths from the node j until the node k and b_{jik} is the number of shortest paths from j until the node k which pass through the node i. Nodes with high betweenness centrality have an important role in the structure of the network because they communicate distant communities. These nodes are somehow controllers or regulators of the information flow and they can act spreading or cutting the flow to other communities.



7.2.3. Degree centrality

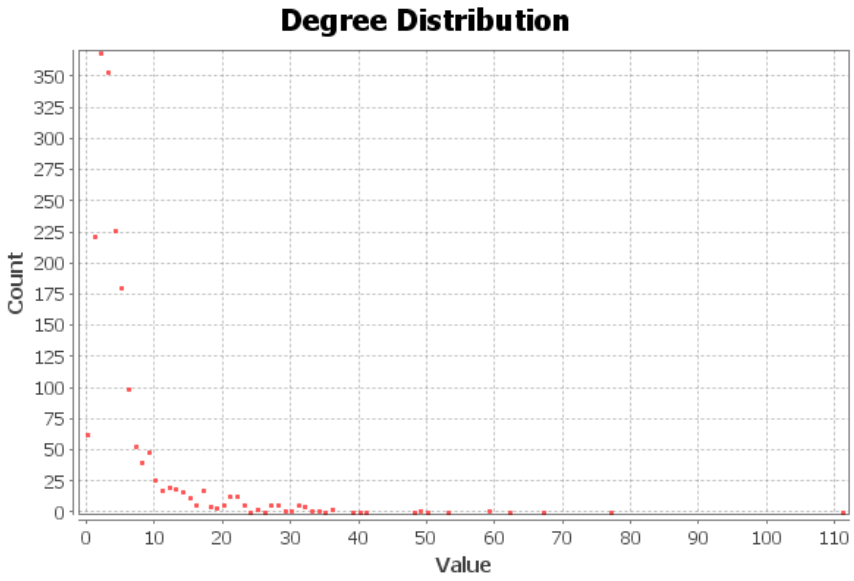
The degree centrality is a volume radial measure and corresponds to the number of links that one node has with others. Mathematically given a graph *G=(V,E)* where *V* is the set of nodes and *E* is the set of edges, for each node $v \in V$ its degree centrality is:

$$C_{DEG}(v) = degree(v)$$

and is calculated as:

$$C_{DEG}(j) = \sum_{i} a_{ij}$$

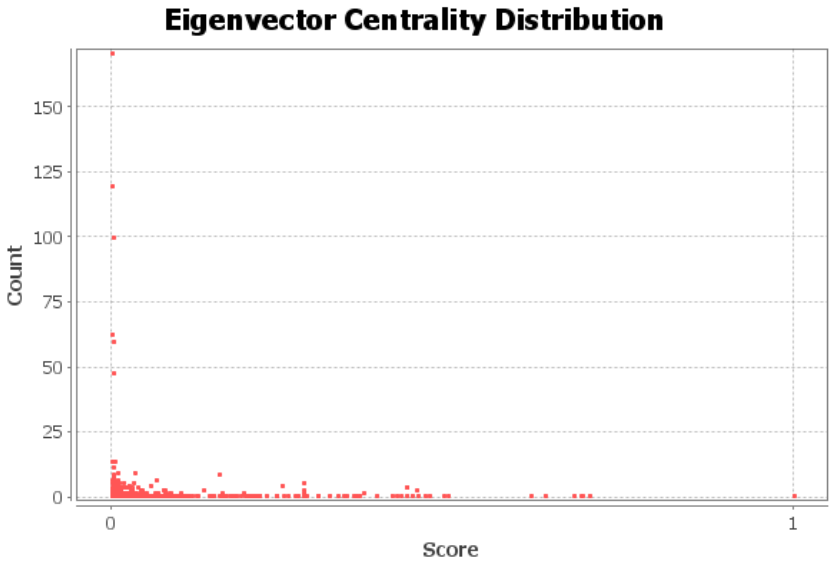
Where a_{ij} are the positions of the adjacency matrix of the graph.
Because our graph is undirected the adjacency matrix is symmetric. For directed graphs, two different measures of degree centrality can be defined, depending on the degree of input or output. This measure indicates the number of authors who have collaborated with an author, that is, quantifies the connectivity or popularity in the network.



7.2.4. Eigenvector centrality

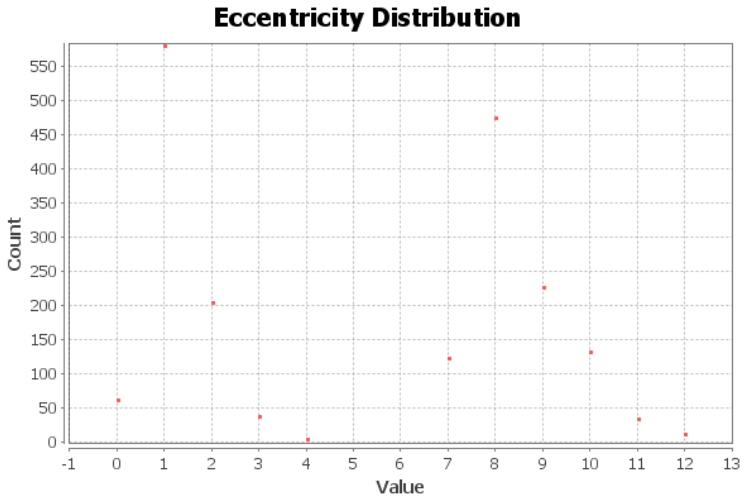
The eigenvector centrality is a radial measure of volume that measures the influence of a node in a network. It assigns relative scores to the nodes of the network based on the following: the nodes

communicated with high score nodes have higher scores. It was proposed by Phillip Bonacich in 1972 and corresponds to the principal eigenvector of the adjacency matrix of the analyzed graph. An example is the Google PageRank calculation, used to measure the relevance of web pages on the Internet.



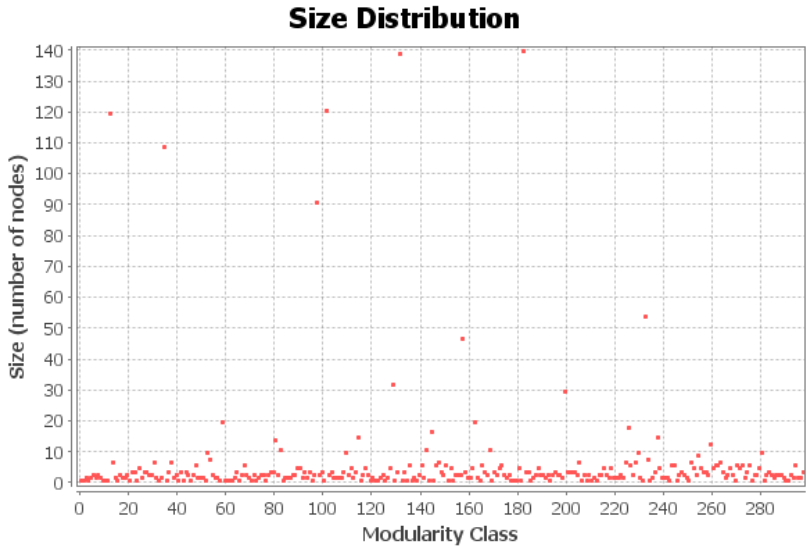
7.1. Eccentricity

The eccentricity of a node is the largest geodesic distance between that node and other one. The geodesic distance is the shortest path between a pair of nodes. Thus, the eccentricity can be considered as the distance from a node to the farthest node.



7.2. Modularity

Modularity is a measure of the network or graphs structure. It was designed to measure the strength of the division of a network into modules (also called groups, clusters or communities). Networks with high modularity have solid connections between nodes of the same community, but just a few connections between nodes from different communities. Modularity is often used in optimization methods to detect communities in networks.



7.3. Clustering coefficient

The clustering coefficient of a node quantifies how a node is grouped (or interconnected) to its neighbors. It is calculated as the number of edges connecting node's neighbors divided by the total number of possible edges between its neighbors. If the node is grouped forming a complete graph, its value is the maximum, while a low value indicates a little clustered node in the network. The network clustering coefficient is calculated using the Watts and Strogatz as the average of the coefficients of clustering of all network nodes. A graph is considered as a small world if the clustering coefficient of the network is significantly higher than another that can provide a random graph constructed with the same set of nodes, and if at the same time has a small average distance. So, we will generate a random graph with the same number of nodes and the same probability of interconnection.

Firstly we calculate the quantity of edges that would have our graph if it were a full graph:

$$\frac{n(n-1)}{2}$$

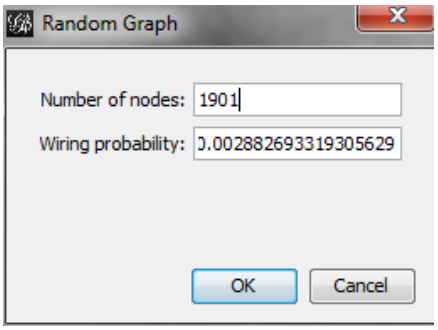
Where *n* is the number of nodes, so we have:

$$\frac{1901(1901-1)}{2} = 1805950 \text{ possible edges}$$

The interconnection probability of the current graph is calculated as:

$$\frac{\text{number of edges}}{\text{total number of possible edges}} = \frac{5206}{1805950} = 0.00288269$$

We generate the random graph:



We calculate and compare the new coefficients and we note that the two conditions are fulfilled, so our graph is a small world.

	Clustering coefficient	Average path length
Graph of study	0.874	4.492
Random graph	0.003	4.623

7.4. Network diameter

In a graph the distance between two nodes is the least number of edges of the possible paths between them. The diameter is the largest distance between all the pairs of nodes of the graph. We can also consider the average path's length as the graph average distance between all pairs of nodes.
In the study network we have:

- Diameter: 12
- Average path length: 4.49

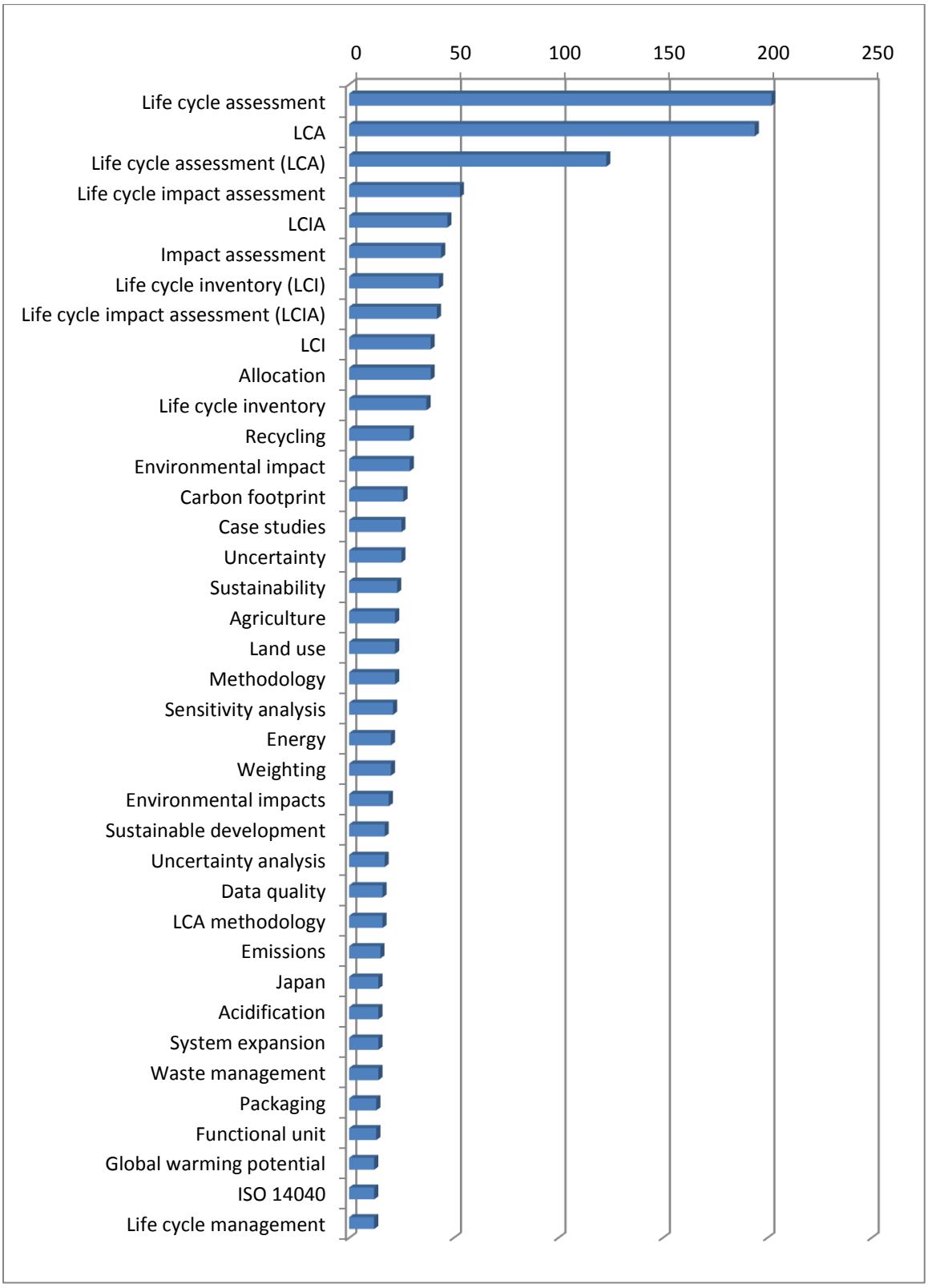
8. Statistics

8.1. Frequency of keywords

To get the number of occurrences of each keyword we do the following query:

```
select
k.id,k.nombre,count(pk.id_keyword) as
total
from publicaciones as p
join publicaciones_keywords as pk
on p.id=pk.id_publicacion
join keywords as k
on k.id=pk.id_keyword
group by k.nombre
order by total desc;
```

And we obtain:

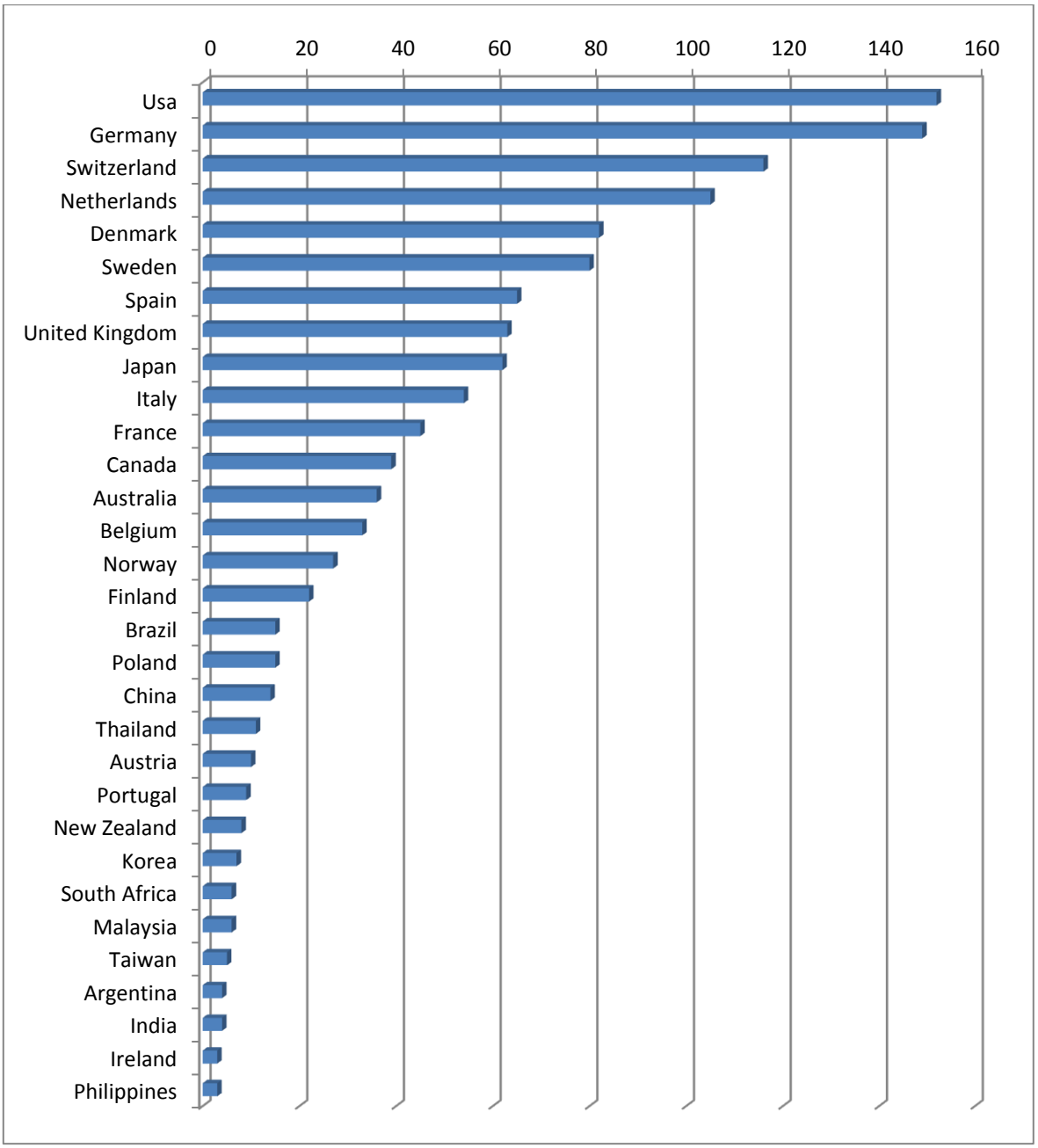

0
50
100
150
200
250
Life cycle assessment
LCA
Life cycle assessment (LCA)
Life cycle impact assessment
LCIA
Impact assessment
Life cycle inventory (LCI)
Life cycle impact assessment (LCIA)
LCI
Allocation
Life cycle inventory
Recycling
Environmental impact
Carbon footprint
Case studies
Uncertainty
Sustainability
Agriculture
Land use
Methodology
Sensitivity analysis
Energy
Weighting
Environmental impacts
Sustainable development
Uncertainty analysis
Data quality
LCA methodology
Emissions
Japan
Acidification
System expansion
Waste management
Packaging
Functional unit
Global warming potential
ISO 14040
Life cycle management

8.2. Frequency of countries

To obtain the frequency of the nationalities of the authors we do the following query:

```
select
p.id,p.nombre,count(pup.id_pais) as
total
from publicaciones as pu
join publicaciones_paises as pup
on pu.id=pup.id_publicacion
join paises as p
on p.id=pup.id_pais
group by p.nombre
order by total desc;
```

And we obtain:

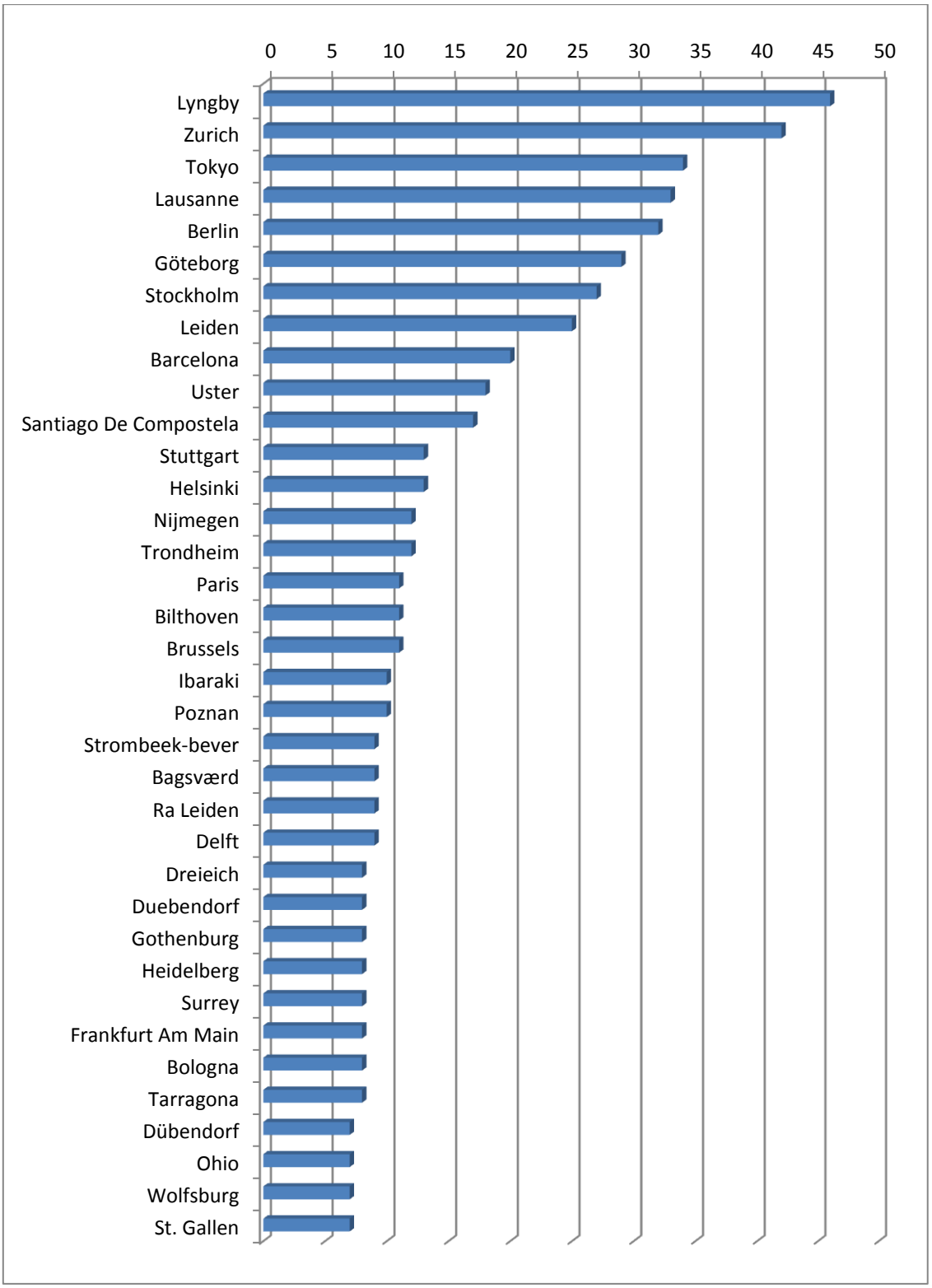

0
20
40
60
80
100
120
140
160
Usa
Germany
Switzerland
Netherlands
Denmark
Sweden
Spain
United Kingdom
Japan
Italy
France
Canada
Australia
Belgium
Norway
Finland
Brazil
Poland
China
Thailand
Austria
Portugal
New Zealand
Korea
South Africa
Malaysia
Taiwan
Argentina
India
Ireland
Philippines

8.3. Frequency of cities

To obtain the frequency of the cities of origin of the authors we do the following query:

```
select
c.id,c.nombre,count(puc.id_ciudad) as
total
from publicaciones as pu
join publicaciones_ciudades as puc
on pu.id=puc.id_publicacion
join ciudades as c
on c.id=puc.id_ciudad
group by c.nombre
order by total desc;
```

And we obtain:

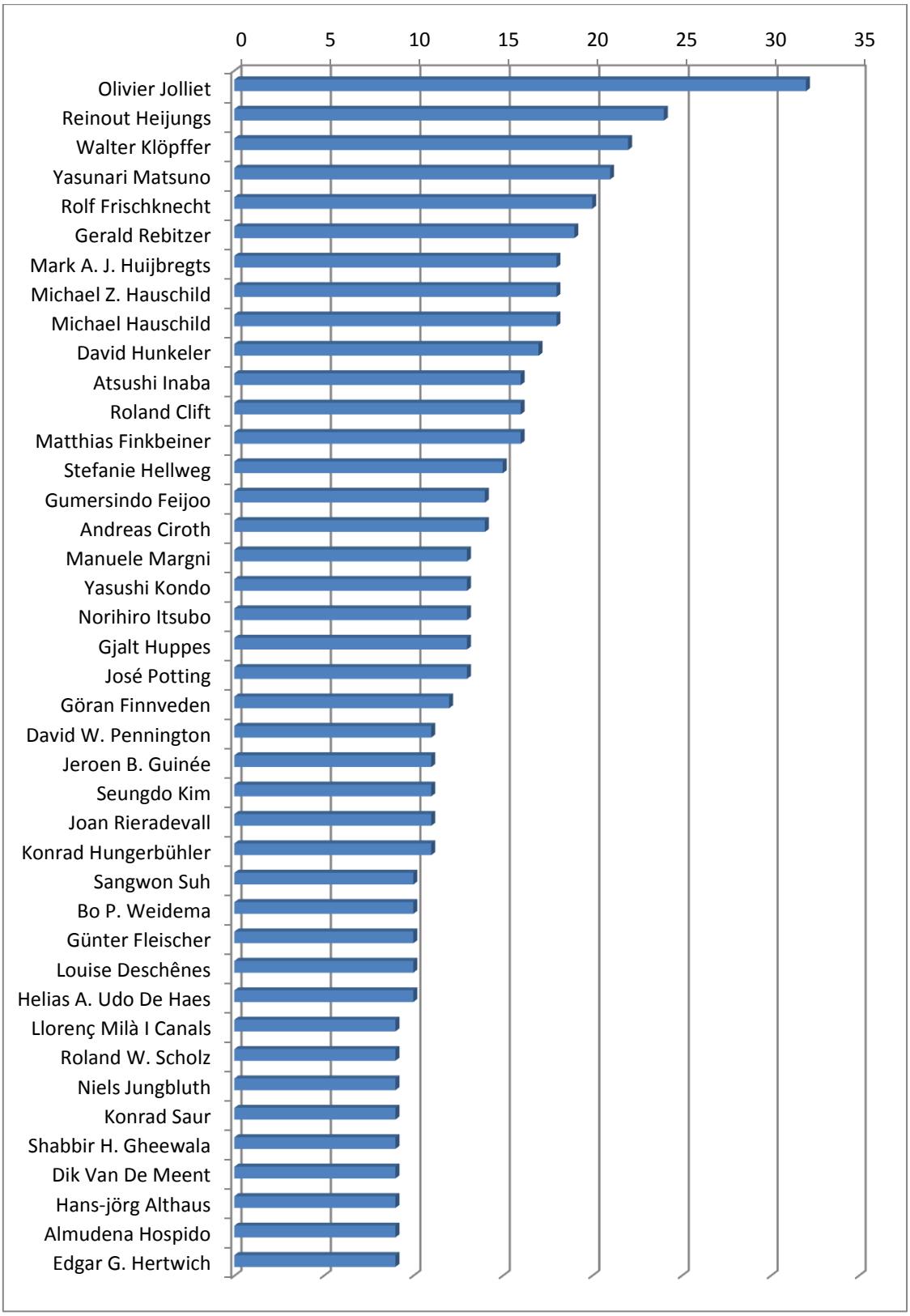

0
5
10
15
20
25
30
35
40
45
50
Lyngby
Zurich
Tokyo
Lausanne
Berlin
Göteborg
Stockholm
Leiden
Barcelona
Uster
Santiago De Compostela
Stuttgart
Helsinki
Nijmegen
Trondheim
Paris
Bilthoven
Brussels
Ibaraki
Poznan
Strombeek-bever
Bagsværd
Ra Leiden
Delft
Dreieich
Duebendorf
Gothenburg
Heidelberg
Surrey
Frankfurt Am Main
Bologna
Tarragona
Dübendorf
Ohio
Wolfsburg
St. Gallen

8.4. Articles by author

To obtain the number of articles written by each author we do the following query:

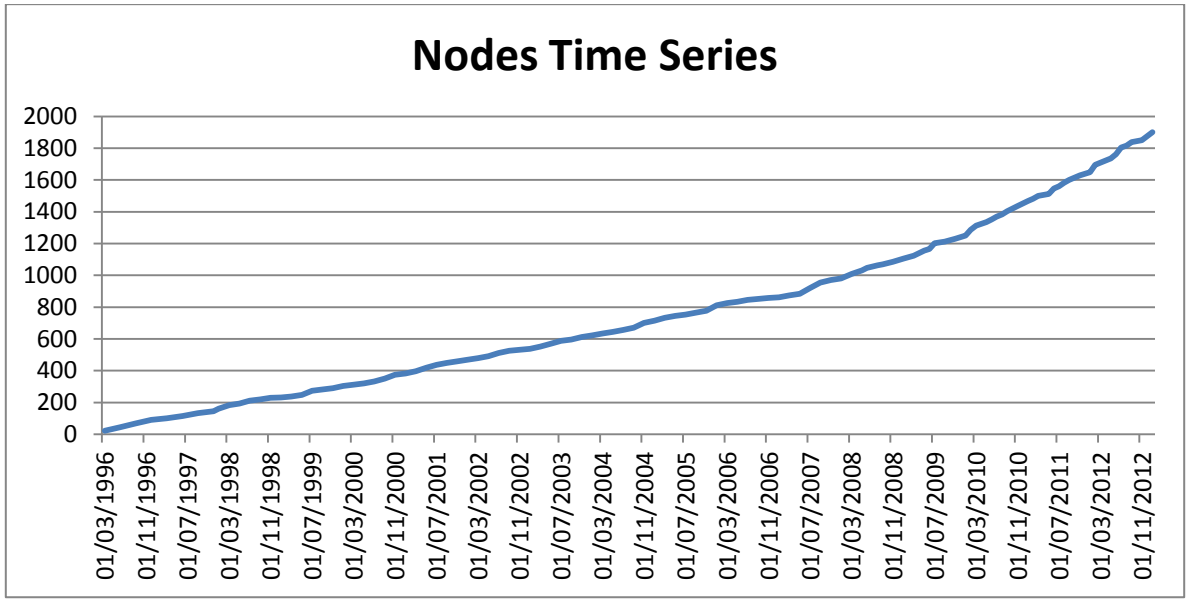
```
select
a.id,a.nombre,count(pua.id_autor) as
total
from publicaciones as pu
join publicaciones_autores as pua
on pu.id=pua.id_publicacion
join autores as a
on a.id=pua.id_autor
group by a.nombre
order by total desc;
```

And we obtain:

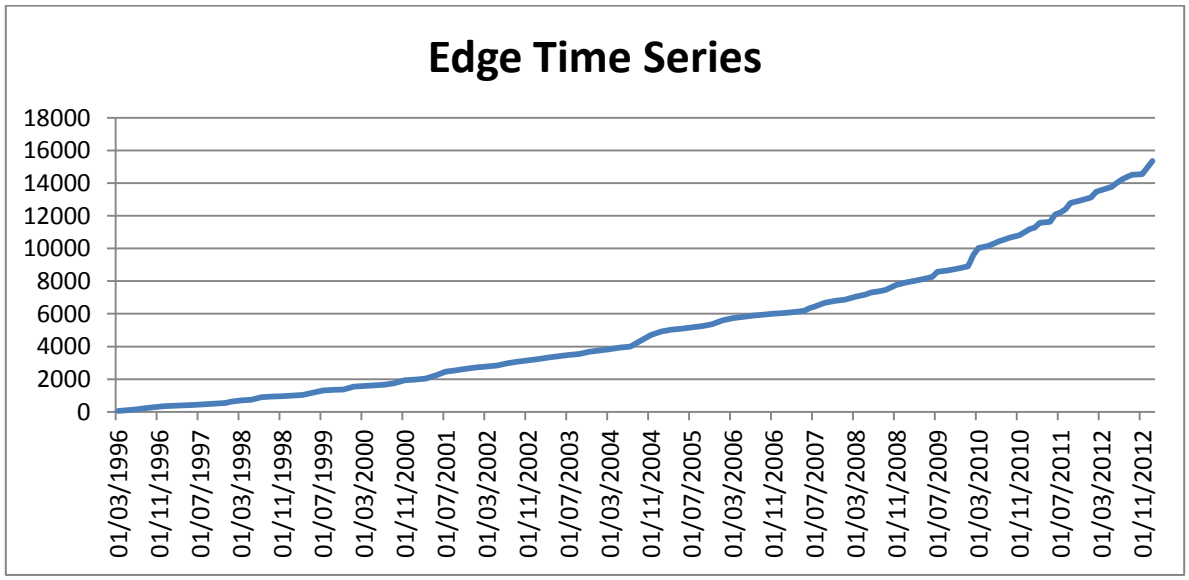

0
5
10
15
20
25
30
35
Olivier Jolliet
Reinout Heijungs
Walter Klöpffer
Yasunari Matsuno
Rolf Frischknecht
Gerald Rebitzer
Mark A. J. Huijbregts
Michael Z. Hauschild
Michael Hauschild
David Hunkeler
Atsushi Inaba
Roland Clift
Matthias Finkbeiner
Stefanie Hellweg
Gumersindo Feijoo
Andreas Ciroth
Manuele Margni
Yasushi Kondo
Norihiro Itsubo
Gjalt Huppes
José Potting
Göran Finnveden
David W. Pennington
Jeroen B. Guinée
Seungdo Kim
Joan Rieradevall
Konrad Hungerbühler
Sangwon Suh
Bo P. Weidema
Günter Fleischer
Louise Deschênes
Helias A. Udo De Haes
Llorenç Milà I Canals
Roland W. Scholz
Niels Jungbluth
Konrad Saur
Shabbir H. Gheewala
Dik Van De Meent
Hans-jörg Althaus
Almudena Hospido
Edgar G. Hertwich

8.5. Growth of the network

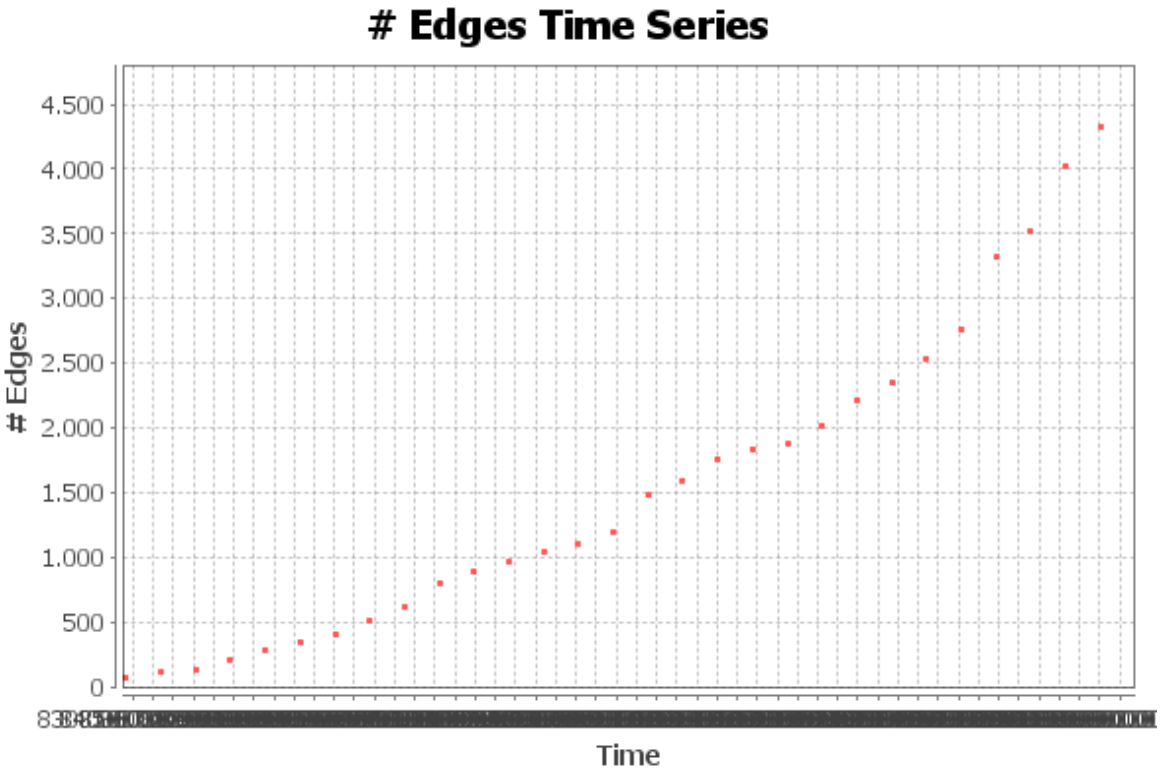
Here we see the evolution of the number of nodes over time.



Here we see the new relations (edges) that appear each year, including all , that means, if an author already had a relationship with another author, and the following year they work again together, this collaboration is counted as a new edge. Thus the graph reaches the 18000 edges while actually in the graph there are 5206 unique edges but with weights.



Here we see the evolution of the number of edges, only taking into consideration unique edges.



9. Conclusions and future applications

In this document we have studied the network of co-authorship of the research community of life cycle analysis. We have used the programming as an essential tool to get the data. We have also presented the different algorithms to process the graph and the various measures for its study.
Our network of co-authorship shows a large amount of collaborations across institutional boundaries. The 53% of the authors are part of a large community in which everyone is connected, through other authors (connex graph). In the other 47% there is a 35% that makes small independent communities and the remainder 12% are authors who publish individually.
We can say that the collaboration between authors in this area is good, but there are still many authors who only work in their small communities or individually. Observing the trend of the graph over the time it appears that as the years advance these authors will be integrated in the core network.
The five authors with highest betweenness centrality are Olivier Jolliet, Gerald Rebitzer, Rolf Frischknecht, Roland Clift and Mark A. J. Huijbregts. Stands above the rest Olivier Jolliet . These authors are consequently controllers or regulators of the flow information, thus, they can act spreading or cutting off the flow to other communities.
The authors with higher degree and therefore better connected with the rest are Olivier Jolliet, Gerald Rebitzer, Michael Z. Hauschild, Mark A. J. Huijbregts, Matthias Finkbeiner and Manuele Margni.
The graph satisfies the small-world conditions, has a diameter of 12 and an average path length of 4.493.
The ranking by nationality is lead by the United States, followed by Germany, Switzerland, Holland, Denmark, Sweden, Spain and the UK.
Network models that have been presented have several applications. The betweenness centrality or PageRank could be used as indicators to assess the impacts of research, or to evaluate quantitatively the prestige of the authors. Being a weighted graph we can filter the links according to the degree which allows emphasizing in important links and removing trivial links.
For future studies, it would be interesting to obtain more data from journals in that area and add them to the existing graph. Thus we could assess the network of experts in this field as a whole.

10. Acknowledgements

Firstly thank to my family, especially to my mother and grandparents, without them I would not be here today.
Secondly to my project tutor, Junbeum Kim, for his help and understanding shown during the work. Also to Charles Perez for his advice on programming.
Finally, to my friends, old friends and new friends that I've made in Troyes and that I will never forget.

11. Bibliography

[1] Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications.* Cambridge University Press.

[2] Junbeum Kim. (2008). *Sustainability Network Theory and Analysis: Focused on Economic, Energy and Environmental Flow Network*, Arizona State University

[3] Watts, D. (2001). *Small worlds: The dynamics of networks between order and randomness.* Princeton University Press.

[4] Newman, M. E. J. (2004). *Analysis of weighted networks.* Physical Review E, 70, 056131.

[5] Newman, M. E. J. (2001). *Scientific collaboration networks: II. Shortest paths, weighted networks, and centrality.* Physical Review E, 64, 016132.

[6] Barabási, A.-L. (2002). *Linked—the new science of networks.* Cambridge, MA: Perseus Publishing.

[7] Xiaoming Liu, Johan Bollen, Michael L. Nelson, Herbert Van de Sompel. (2005). *Co-authorship networks in the digital library research community.* Available online at www.sciencedirect.com.

[8] Theresa Velden, Asif-ul Haque, and Carl Lagoze . (2010). *A New Approach to Analyzing Patterns of Collaboration in Co-authorship Networks - Mesoscopic Analysis and Interpretation.* Computer and Information Science, Cornell University, Ithaca, U.S.A.

[9] James Moody. *The Structure of a Social Science Collaboration Network: Disciplinary Cohesion from 1963 to 1999*. The Ohio State University.

[10] Chaomei Chen, Ray J. Paul. *Visualizing a Knowledge Domain's Intellectual Structure*. Brunel University.

[11] Weimao Ke, Katy Börner and Lalitha Viswanath. *Major Information Visualization Authors, Papers and Topics in the ACM Library.* Indiana University, School of Library and Information Science & School of Informatics.

[12] Fuyuki Yoshikane, Takayuki Nozawa and Keita Tsuji. *Comparative Analysis of Co-authorship Networks Considering Authors' Roles in Collaboration: Differences between the Theoretical and Application Areas.* Faculty of University Evaluation and Research, Tokyo.

[13] M. E. J. Newman. *Coauthorship networks and patterns of scientific collaboration.* Center for the Study of Complex Systems and Department of Physics, University of Michigan.

[14] Katy Börner, Luca Dall'Asta, Weimao Ke, Alessandro Vespignani. *Studying the Emerging Global Brain: Analyzing and Visualizing the Impact of Co-Authorship Teams.*

[15] Tze-Haw Huang and Mao Lin Huang. *Analysis and Visualization of Co-authorship Networks for Understanding Academic Collaboration and Knowledge Domain of Individual Researchers.* University of Technology, Sydney, Australia.

[16] José Federico Medrano, Joé Luis Alonso Berrocal y Carlos G. Figuerola. *Visualización de Grafos Web*. Universidad de Salamanca.

[17] Jesse D. Lecy, PhD Student, Maxwell School. *Citation Network Analysis, A software tool for structured literature reviews.*

[18] Molina, J. L. *El análisis de redes sociales. Una introducción.* Barcelona: Editorial Bellaterra.

12. Annexes

12.1. Code for creating the database

```
-- --------------------------------------------------------
CREATE TABLE IF NOT EXISTS `autores` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=2059 ;
-- --------------------------------------------------------
CREATE TABLE IF NOT EXISTS `autores_universidades` (
  `id_autor` int(10) unsigned NOT NULL,
  `id_universidad` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id_autor`,`id_universidad`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
-- --------------------------------------------------------
CREATE TABLE IF NOT EXISTS `ciudades` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=848 ;
-- --------------------------------------------------------
CREATE TABLE IF NOT EXISTS `keywords` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `nombre` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=3950 ;
-- --------------------------------------------------------
CREATE TABLE IF NOT EXISTS `paises` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=316 ;
-- --------------------------------------------------------
CREATE TABLE IF NOT EXISTS `publicaciones` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `titulo` varchar(220) COLLATE utf8_unicode_ci NOT NULL,
  `fecha` date NOT NULL,
  `volumen` tinyint(3) unsigned NOT NULL,
  `issue` tinyint(3) unsigned NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1061 ;
-- --------------------------------------------------------
CREATE TABLE IF NOT EXISTS `publicaciones_autores` (
```

```
  `id_publicacion` int(10) unsigned NOT NULL,
  `id_autor` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id_publicacion`,`id_autor`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
-----------------------------------------------------------
CREATE TABLE IF NOT EXISTS `publicaciones_ciudades` (
  `id_publicacion` int(10) unsigned NOT NULL,
  `id_ciudad` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id_publicacion`,`id_ciudad`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
-----------------------------------------------------------
CREATE TABLE IF NOT EXISTS `publicaciones_keywords` (
  `id_publicacion` int(10) unsigned NOT NULL,
  `id_keyword` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id_publicacion`,`id_keyword`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
-----------------------------------------------------------
CREATE TABLE IF NOT EXISTS `publicaciones_paises` (
  `id_publicacion` int(10) unsigned NOT NULL,
  `id_pais` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id_publicacion`,`id_pais`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
-----------------------------------------------------------
CREATE TABLE IF NOT EXISTS `universidades` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `nombre` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=2179 ;
```

12.2. Program code of extraction and processing of data

```
#!/usr/local/php/bin/php -q
<?php
//============================ DIRECTIVAS ================================//
set_time_limit(0);
ob_implicit_flush(1);
echo str_pad('',4096);
//=========== VARIABLES DE PLANTILLA
=========================================================//
$file="backup.csv";
$fileAutores="autores.csv";
fclose(fopen($file,"w"));
fclose(fopen($fileAutores,"w"));
$GLOBALS['__opc_TiempoPeticionesMin']=3;
$GLOBALS['__opc_TiempoPeticionesMax']=5;
//=========== INICIALIZACION DE VARIABLES
=========================================//
```

```
$oOpen=new url_open();       // Objecto para realizar peticiones a las
webs, retorna file.html
$vRefRep=array();
$noIntegrados=0;
$volumen=$issue=$fecha=$titulo='';
//====================== PARSEO
=============================================================//
##-------Recorro VOLUMES(años)----##
echo "Empieza el parseo: ";
echo date("Y-m-d H:i:s")."\n";
echo "Empieza el parseo: ";
echo date("Y-m-d H:i:s")."\n";

$uIni = "http://link.springer.com/journal/volumesAndIssues/11367";
$tall='<a class="title" href="';

if(!($fa=fopen($oOpen->run($uIni,'file1.html'),"r")))    echo "Error
abriendo la url: ".$uIni."\n";
else{
    while(!feof($fa)){
        $buff=trim(fgets($fa));
        if(strstr($buff,$tall)){

$url='http://link.springer.com'.getStringBetween2($tall,'">',$buff);
            echo "1- URL a escanear: [$url]\n";
            Prods($url);
            sleep(rand(3,5));
        }
    }
}

echo "Acaba el parseo: ";
echo date("Y-m-d H:i:s")."\n";
die("finnnnnnnnnnnnn");

//====================== FUNCION PRODS
=====================================================//
function Prods($ul){
    global $oOpen;
    $tallUrl2='h3 class="title">';//para articulos con Issue
    $tallPagSiguiente='<a class="next" href="';

    if(!($fd=fopen($oOpen->run($ul,"www2.html"),"r")))    echo "Error
abriendo la url: $ul\n";
    else{
        $finalHoja=0;
        while(!feof($fd)){
            $buff=trim(fgets($fd));

            if(strstr($buff,$tallPagSiguiente)){
                $finalHoja++;
            }
```

```
            if(strstr($buff,$tallUrl2)){
                $buff=trim(fgets($fd));
                $url=getStringBetween2('href="','"',$buff);
                $url="http://link.springer.com".$url;
                $url=str_replace('fulltext.html','',$url);
                echo "2- URL a escanear: [$url]\n";
                Parsear($url);
                sleep(rand(3,5));
            }
            if($finalHoja==2){

$url='http://link.springer.com'.getStringBetween2($tallPagSiguiente,'"',$
buff);
                $finalHoja=0;
                echo "2.2 - Siguiendo paginado URL a escanear: [$url]\n";
                Prods($url);
                sleep(rand(3,5));
            }
        }
        fclose($fd);
    }
}
//====================== FUNCION PARSEAR
=====================================================//
function Parsear($ul){
    global $oOpen,$vRefRep,$noIntegrados;
    $tallArticuloPublico='<span class="action icon-view">';
    $tallTitulo='<dd id="abstract-about-title">';
    $tallAutores='<li itemprop="author" itemscope="itemscope"
itemtype="http://schema.org/Person">';
    $tallKw='<ul class="abstract-keywords">';
    $tallFinKw='</ul>';
    $vAutores=array();
    $vPaises=array();
    $vCiudades=array();
    $vKW=array();
    $tallFecha='<span id="date" itemprop="datePublished">';
    $tallVolumen='<span id="volume-range">Volume';
    $tallIssue='>Issue ';
    $tallPais='<span class="affiliation">';
    $tallFecha2='dd id="abstract-about-cover-date">';
    $espublico=0;

    if(!($fd=fopen($oOpen->run($ul,"www3.html"),"r")))    echo "Error
abriendo la url: $ul\n";
    else{
        while(!feof($fd)){
            $buff=trim(fgets($fd));
            if(strstr($buff,$tallTitulo)){
                while(!stristr($buff,'</dd>')){
                    $buff.=trim(fgets($fd));
                }
```

```
                $titulo=getStringBetween2($tallTitulo,'</dd>',$buff);
                $titulo=strip_tags($titulo);
                codificacion($titulo);
                $titulo=trim($titulo);
                echo $titulo;
                echo "\n";
                }
                if(stristr($buff,$tallArticuloPublico)){
                    $espublico=1;
                    echo "es publico";
                    echo "\n";
                }
                if(stristr($buff,$tallAutores)){
                    while(!stristr($buff,'</ul>')){
                        $buff.=trim(fgets($fd));
                    }
                    $vLiAutor=explode($tallAutores,$buff);
                    array_shift($vLiAutor);
                    foreach($vLiAutor as $indice => $valor){

$tempAutor=getStringBetween2('itemprop="name">','</',$valor);
                        codificacion($tempAutor);
                        $tempAutor=trim($tempAutor);
                        $tempAutor=mb_strtolower($tempAutor,"UTF-8");
                        $tempAutor=ucwords($tempAutor);
                        $vAutores[]=$tempAutor;
                        unset($tempAutor);

                        $tempUniversidad=getStringBetween2('<sup
title="','">(',$valor);
                        codificacion($tempUniversidad);
                        $tempUniversidad=trim($tempUniversidad);
                        $vUniversidades[]=$tempUniversidad;
                        unset($tempUniversidad);


$tempIdUniversidad=getStringBetween2('">(',')</',$valor);
                        $tempIdUniversidad=trim($tempIdUniversidad);
                        $vIdUniversidad[]=$tempIdUniversidad;
                        unset($tempIdUniversidad);
                    }
                }

                if(stristr($buff,$tallFecha2)){
                    $tempFecha=getStringBetween2($tallFecha2,'</dd>',$buff);
                    $fecha=trim($tempFecha);
                    unset($tempFecha);
                }

                if(stristr($buff,$tallPais)){
                    $buff=trim(fgets($fd));
                    $vTemp=explode(',',$buff);
```

```
                    $tempPais=array_pop($vTemp);
                    codificacion($tempPais);
                    $tempPais=trim($tempPais);
                    $tempPais=mb_strtolower($tempPais);
                    $tempPais=ucwords($tempPais);
                    $vPaises[]=$tempPais;

                    $tempCiudades=array_pop($vTemp);
                    codificacion($tempCiudades);
                    $tempCiudades=trim($tempCiudades);
                    $tempCiudades=mb_strtolower($tempCiudades,"UTF-8");
                    $tempCiudades=ucwords($tempCiudades);
                    $vCiudades[]=$tempCiudades;
                    unset($tempPais);
                    unset($tempCiudades);
                }
                if(strstr($buff,$tallKw)){
                    $buff=trim(fgets($fd));
                    while(!stristr($buff,$tallFinKw)){
                        while(!stristr($buff,'</li>')){
                            $buff.=trim(fgets($fd));
                        }
                        $tempvKW=getStringBetween2('<li>','</li>',$buff);
                        codificacion($tempvKW);
                        $vKW[]=trim($tempvKW);
                        $buff=trim(fgets($fd));
                    }
                }

                if(strstr($buff,$tallFecha)){
                    if(!isset($fecha)){
                        $fecha=getStringBetween2($tallFecha,'</',$buff);//se
usa esta fecha si no está la otra
                        if(strstr($fecha,'-')){
                            $fecha='01-'.$fecha;
                        }
                        $fecha=date("Ymd",strtotime($fecha));
                    }
                    $volumen=getStringBetween2($tallVolumen,'</',$buff);
                    $volumen=trim($volumen);
                    $volumen=preg_replace('#[^0-9]#','',$volumen);
                    $issue=getStringBetween2($tallIssue,'</',$buff);
                    $issue=trim($issue);
                    $issue=preg_replace('#[^0-9]#','',$issue);
                }

    }
    fclose($fd);

    if($espublico){

if(!integrable($volumen,$issue,$fecha,$titulo,$vAutores,$vKW,$ul,$vPaises
```

```
,$vCiudades,$vUniversidades)){
            $noIntegrados++;
            echo "\n**********faltan datos [$noIntegrados]\n";
        }
        else{
            if(!isset($vRefRep[md5($ul)])){ //Comprueba que no está
repetido
            $vRefRep[md5($ul)]=1;
            echo "URL: [$ul]\n";
            echo "FECHA: [$fecha]\n";
            echo "TITULO: [$titulo]\n";
            echo "AUTORES: ";
            print_r($vAutores);
            echo "\n";
            echo "UNIVERSIDADES: ";
            print_r($vUniversidades);
            echo "\n";
            echo "KEYWORDS: ";
            print_r($vKW);
            echo "\n";
            echo "CIUDADES: ";
            print_r($vCiudades);
            echo "\n";
            echo "PAISES: ";
            print_r($vPaises);
            echo "\n";

Add2File($volumen,$issue,$fecha,$titulo,$vAutores,$vKW,$vCiudades,$vPaise
s);
            Add2FileAutores($vAutores);

AddDB($volumen,$issue,$fecha,$titulo,$vAutores,$vKW,$vCiudades,$vPaises,$
vUniversidades);
            $volumen=$issue=$fecha=$titulo='';
            unset ($volumen);
            unset ($issue);
            unset ($fecha);
            unset ($titulo);
            unset ($vAutores);
            unset ($vKW);
            unset ($vCiudades);
            unset ($vPaises);
            unset ($vUniversidades);
            }
        }
    }
    else{
        echo "URL no publica: [$ul]\n";
        $espublico=0;
        $volumen=$issue=$fecha=$titulo='';
        unset ($volumen);
        unset ($issue);
```

```
        unset ($fecha);
        unset ($titulo);
        unset ($vAutores);
        unset ($vKW);
        unset ($vCiudades);
        unset ($vPaises);
        unset ($vUniversidades);
    }
    }
}
//=============================================================================
==================//
function
AddDB($volumen,$issue,$fecha,$titulo,$vAutores,$vKW,$vCiudades,$vPaises,$
vUniversidades){
//================Datos conexion a la BD================//
$ip_mysql_server='localhost';
$usuario='root';
$password='*******';//password servidor mysql
$nombre_BD='pfc';
//================Tratamiento datos de entrada================//
$fecha=str_replace('-','',$fecha);
//================Conexion a BD================//
$con=mysqli_connect($ip_mysql_server,$usuario,$password,$nombre_BD);
mysqli_set_charset($con, "utf8");
if (!$con){
    die('Error de Conexión ('. mysqli_connect_errno() .') '.
mysqli_connect_error());
}else{
    echo 'Conexion establecida... ' . mysqli_get_host_info($con) . "\n";
}
//================Inserto publicacion================//
$titulo = mysqli_real_escape_string($con,$titulo);
$query="insert into publicaciones (titulo,fecha,volumen,issue)
                values('$titulo',$fecha,$volumen,$issue)";
$resultado=mysqli_query($con,$query);
$id_ultima_pub=mysqli_insert_id($con);
unset($resultado);

//================Compruebo si los datos ya estan en la
BD================//
introduceToBD($vAutores,$con,$id_ultima_pub,"autor");
introduceToBD($vKW,$con,$id_ultima_pub,"kw");
introduceToBD($vCiudades,$con,$id_ultima_pub,"ciudad");
introduceToBD($vPaises,$con,$id_ultima_pub,"pais");

//================Inserto universidad y relacion con
autores================//
foreach($vUniversidades as $indice => $universidad){
$universidad = mysqli_real_escape_string($con,$universidad);
$query="select id,nombre from universidades where nombre='$universidad'";
$resultado=mysqli_query($con,$query);
```

```
    //Si no está, lo inserto y obtengo id_universidad
    if($fila=mysqli_fetch_array($resultado,MYSQLI_ASSOC)){
        echo "Si está********!!";
        print_r($fila);
        $id_universidad=$fila["id"];
        unset($resultado);
    }
    else{
        echo "No está*****!!";
        //Inserto universidad
        $query="insert into universidades (nombre)
                values('$universidad')";
        $resultado=mysqli_query($con,$query);
        $id_universidad=mysqli_insert_id($con);
        unset($resultado);
    }
    //obtengo id_autor
    $query="select id from autores where nombre='$vAutores[$indice]'";
    $resultado=mysqli_query($con,$query);
    $fila=mysqli_fetch_array($resultado,MYSQLI_ASSOC);
    $id_autor=$fila["id"];
    unset($resultado);
    //Inserto la relacion universidad autor
    $query="insert into autores_universidades ()
                values($id_autor,$id_universidad)";
    $resultado=mysqli_query($con,$query);
    unset($resultado);
}
mysqli_close($con);
}
//================FUNCIONES================//
function introduceToBD($vDatos_input,$con,$id_ultima_pub,$autorOkw){
if($autorOkw=="autor"){
    $tabla1='autores';
    $tabla2='publicaciones_autores';
}
elseif($autorOkw=="kw"){
    $tabla1='keywords';
    $tabla2='publicaciones_keywords';
}
elseif($autorOkw=="ciudad"){
    $tabla1='ciudades';
    $tabla2='publicaciones_ciudades';
}
elseif($autorOkw=="pais"){
    $tabla1='paises';
    $tabla2='publicaciones_paises';
}
elseif($autorOkw=="universidad"){
    $tabla1='universidades';
    $tabla2='publicaciones_universidades';
}
```

```
else{die("Último parametro mal escrito");}

foreach($vDatos_input as $dato){
    $dato = mysqli_real_escape_string($con,$dato);
    $query="select id,nombre from $tabla1 where nombre='$dato'";
    $resultado=mysqli_query($con,$query);
    //Si no está, lo inserto y obtengo id_autor
    if($fila=mysqli_fetch_array($resultado,MYSQLI_ASSOC)){
        echo "Si está!!";
        print_r($fila);
        $id_autor=$fila["id"];
        //Inserto id_ultima_pub y id_autor
        $query="insert into $tabla2 () values($id_ultima_pub,$id_autor)";
        $resultado=mysqli_query($con,$query);
        unset($resultado);
    }
    else{
        echo "No está!!";
        //Inserto autor en autores
        $query="insert into $tabla1 (nombre) values('$dato')";
        $resultado=mysqli_query($con,$query);
        unset($resultado);
        $id_autor=mysqli_insert_id($con);
        //Inserto id_ultima_pub y id_autor
        $query="insert into $tabla2 () values($id_ultima_pub,$id_autor)";
        $resultado=mysqli_query($con,$query);
        unset($resultado);
    }
}
}


function getStringBetween2($startString, $stopString, $buffer){
    if($stopString==''){
        $buffer=$buffer.'##';
        $stopString='##';
    }
    if($startString==''){
        $buffer='##'.$buffer;
        $startString='##';
    }
 if($startString != '' && $stopString != ''){
    if (($start=strpos($buffer, $startString)) !== false) {
        if (($stop=strpos($buffer, $stopString,
$start+strlen($startString))) !== false) {
            $from =$start+strlen($startString);
            $length=strlen($buffer)-$from-(strlen($buffer)-$stop);
            $res = substr($buffer, $from, $length);
            return ($res);
        }
    }
 }
```

```
 return(false);
}

function
Add2File($volumen,$issue,$fecha,$titulo,$vAutores,$vKW,$vCiudades,$vPaise
s){
global $file;//crea backup.csv
$vSeparador=array("");
if(!($fd=fopen($file,"a"))) echo "\nERROR al crear ".$file;
    else{

$vNuevaLinea=array_merge((array)$fecha,(array)$volumen,(array)$issue,(arr
ay)$titulo,$vSeparador,$vAutores,$vSeparador,$vKW,$vSeparador,$vCiudades,
$vSeparador,$vPaises);
        $nuevaLinea=implode (';',$vNuevaLinea);
        $nuevaLinea=$nuevaLinea.chr(13).chr(10);
        fwrite($fd,$nuevaLinea);
        fclose($fd);
    }
}
function Add2FileAutores($vAutores){
global $fileAutores;
$vSeparador=array("");
if(!($fd=fopen($fileAutores,"a"))) echo "\nERROR al crear ".$fileAutores;
    else{
        if(count($vAutores)==1){
            $elemento=array_shift($vAutores);
            $nuevaLinea='"'.$elemento.'"'.chr(13).chr(10);
            fwrite($fd,$nuevaLinea);
        }
        else{
            foreach($vAutores as $indice1 => $valor1){
                $elemento=array_shift($vAutores);
                foreach($vAutores as $indice => $valor){

$nuevaLinea='"'.$elemento.'","'.$valor.'"'.chr(13).chr(10);
                    fwrite($fd,$nuevaLinea);
                }
            }
        }
    }
    fclose($fd);
}


function
integrable($volumen,$issue,$fecha,$titulo,$vAutores,$vKW,$ul,$vPaises,$vC
iudades){
    $noSeIntegra=0;
    if(trim($fecha)==''|| !isset($fecha)){
        echo "\n\n";
        echo __FUNCTION__."() - INFO: Falta la fecha de [$ul], no se
```

```
integrara\n\n";
        $noSeIntegra=1;
    }
    if(trim($titulo)==''|| !isset($titulo)){
        echo "\n\n";
        echo __FUNCTION__."() - INFO: Falta el titulo de [$ul], no se
integrara\n\n";
        $noSeIntegra=1;
    }
    if(count($vAutores)==0|| !isset($vAutores)){
        echo "\n\n";
        echo __FUNCTION__."() - INFO: Faltan los autores de [$ul], no se
integrara\n\n";
        $noSeIntegra=1;
    }
    if(count($vPaises)==0|| !isset($vPaises)){
        echo "\n\n";
        echo __FUNCTION__."() - INFO: Faltan los paises de [$ul], no se
integrara\n\n";
        $noSeIntegra=1;
    }
    if(count($vCiudades)==0|| !isset($vCiudades)){
        echo "\n\n";
        echo __FUNCTION__."() - INFO: Faltan las ciudades de [$ul], no se
integrara\n\n";
        $noSeIntegra=1;
    }
    if($noSeIntegra){
        return false;
    }
    return true;
}


class url_open {
    var $url='';
    var $fileHtml= "file.html";                    //Fichero de salida
    var $fileError= "error.txt";                   //Fichero de errores
    var $fileHead= "head.txt";                     //Cabeceras recibidas
    var $aParams=array(
        CURLOPT_FAILONERROR=>1,                        //Si el codigo de
cabecera es mayor de 300 no devuelve resultado
        CURLOPT_HEADER=>0,                                //Incluir la
cabecera en el resultado
        CURLOPT_CONNECTTIMEOUT=>60,                //Tiempo máximo para
establecer una conexion
        CURLOPT_TIMEOUT=>120,                             //Tiempo máximo
de ejecución del curl, conexión + descarga
        CURLOPT_USERAGENT=>"Mozilla/5.0 (Windows NT 5.1; rv:2.0)
Gecko/20100101 Firefox/4.0",   //User agent utilizado
        //CURLOPT_USERPWD=>"",                        //Pasar un usuario y
contraseña, formato: user:passw
```

```
        CURLOPT_FOLLOWLOCATION=>1,                //Permite redirecciones
        CURLOPT_MAXREDIRS=>5,                           //Permite un
Máximo de 5 Redirecciones
        CURLOPT_POST=>NULL,                             //Método POST
desactivado, para utilizarlo this->run_post
        CURLOPT_POSTFIELDS=>NULL,                  //Datos a enviar por
POST
        CURLOPT_COOKIEFILE=>"cook.txt",      //Cookie File
        CURLOPT_COOKIEJAR=>"cook.txt",       //Cookie File
        CURLOPT_VERBOSE=>0,                             //Modo verbose de
la petición curl
        CURLOPT_REFERER=>''                             //Utilizar el
siguiente referer
        );
    var $iTimePeticiones=3;                        //Por defecto se leen
desde base de datos
    var $iTimePeticionesMax=5;                 //Por defecto se leen
desde base de datos
    var $bVerbose=0;                                        //Modo
Verbose
    var $aInfoCurl=array();                        //Guarda la
información del último curl
    var $autoReferer=1;                             //Se gestiona
automáticamente el referer

    function url_open($url='',$sFile=''){
        $this->url=$url;
        $this-
>fileHtml=($sFile!='')?getcwd()."/".$sFile:getcwd()."/".$this->fileHtml;
        $this->fileError=getcwd()."/".$this->fileError;
        $this->fileHead=getcwd()."/".$this->fileHead;
        $this->iTimePeticiones=$GLOBALS['__opc_TiempoPeticionesMin'];
        $this->iTimePeticionesMax=$GLOBALS['__opc_TiempoPeticionesMax'];
        $this->set_params(CURLOPT_COOKIEFILE,getcwd()."/".$this-
>get_param(CURLOPT_COOKIEFILE));
        $this->set_params(CURLOPT_COOKIEJAR,getcwd()."/".$this-
>get_param(CURLOPT_COOKIEJAR));
    }

    function set_params($k,$v){
        $this->aParams[$k]=$v;
    }
    function get_param($k){
        return (isset($this->aParams[$k]))?$this->aParams[$k]:false;
    }
    function run($url='',$sFile=''){

        $this->url=($url!='')?$url:$this->url;
        $this->fileHtml=($sFile!='')?getcwd()."/".$sFile:$this->fileHtml;
        if($this->autoReferer)  $this-
>set_params(CURLOPT_REFERER,(($this->get_info_curl('url'))?$this-
>get_info_curl('url'):''));  //Gestión automática del referer
```

```
        $ch=@curl_init($this->url);
    if(! ( $__fg=@fopen($this->fileHtml,"w") ) ) return NULL;
    $__fe=@fopen($this->fileError,"w");
    $__fh=@fopen($this->fileHead,"w");
      curl_setopt($ch,CURLOPT_FILE,$__fg);         //Fichero de resultados
      curl_setopt($ch,CURLOPT_STDERR,$__fe);       //Fichero de errores
      curl_setopt($ch,CURLOPT_WRITEHEADER,$__fh); //Fichero de cabeceras

      foreach($this->aParams as $k=>$v)
        if($v!==NULL)
            @curl_setopt($ch,$k,$v);

      @curl_exec($ch);

      $this->aInfoCurl=@curl_getinfo($ch);
      if($this->bVerbose) echo "[CURL_TOTAL_TIME: ".$this-
>aInfoCurl['total_time']."]\n";
      @curl_close($ch);
      @fclose($__fg);
      @fclose($__fe);
      @fclose($__fh);

      $this->auto_sleep();
      return $this->fileHtml;
    }
    function run_post($url='',$sFile='',$aPost=array()){
        $this->set_params(CURLOPT_POST,1);
        $this-
>set_params(CURLOPT_POSTFIELDS,(is_array($aPost)?implode("&",$aPost):$aPo
st) );
        return $this->run($url,$sFile);
    }
    function auto_sleep(){
        if($this->iTimePeticiones && $this->iTimePeticionesMax ){
        $iSleep=rand($this->iTimePeticiones,$this->iTimePeticionesMax);
        if($this->bVerbose) echo "[WAITING RAND: $iSleep seg]\n";
        sleep($iSleep);
      }elseif($this->iTimePeticiones){
        if($this->bVerbose) echo "[WAITING: ".$this->iTimePeticiones."
seg]\n";
        sleep($this->iTimePeticiones);
      }
    }
    function get_info_curl($param){
        return (isset($this->aInfoCurl[$param]))?$this-
>aInfoCurl[$param]:false;

    }
}

function pstr_decode($str){
    return (stristr($str,'Ã'))?utf8_decode($str):$str;
```

```
}

function codificacion(&$frase){
    $frase=str_replace("â€œ","¿",$frase);
    $frase=str_replace("â","Â",$frase);
    $frase=str_replace("ã","Ã",$frase);
    $frase=str_replace('Â€"','-",$frase);
    $frase=str_replace("Â€™","'",$frase);
    $frase=str_replace("Â€˜","'",$frase);
    $frase=str_replace("Ã±","ñ",$frase);
    $frase=str_replace("Ã¡","á",$frase);
    $frase=str_replace("Ã©","é",$frase);
    $frase=str_replace("Ã¨","è",$frase);
    $frase=str_replace("Ã","í",$frase);
    $frase=str_replace("Ã³","ó",$frase);
    $frase=str_replace("Ã°","ú",$frase);
    $frase=str_replace("Â¿","¿",$frase);
    $frase=str_replace("Â","",$frase);
    $frase=str_replace("Ã","à",$frase);
    $frase=iconv("UTF-8","UTF-8", $frase);
    $frase =pstr_decode($frase);
}
?>
```