CLASSIFICATION OF COMPLEX NETWORKS IN TERMS OF TOPOLOGICAL PROPERTIES

(TOPOLOJİK ÖZELLİKLERİNE BAĞLI OLARAK KARMAŞIK AĞ SINIFLANDIRMASI)

by

Burcu Kantarcı, B.Sc.

Thesis

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

Date of Submission : October 6, 2013

Date of Defence Examination : October 24, 2013

Supervisors: Dr. Vincent Labatut

Asst. Prof. Dr Murat Akın

Committee Members: Assoc. Prof. Y. Esra Albayrak

Assoc. Prof. Ebru Angün

Asst. Prof. B. Atay Özgövde

September 2013

ACKNOWLEDGEMENTS

First of all, I would like to present my deepest gratitude to Asst. Prof. Dr. Vincent Labatut for the patient guidance, useful comments, remarks and engagement through the learning process of my master thesis. He guided me with very valuable knowledge and skills in many areas.

I am also grateful for the useful support and guidance given by Günce Keziban Orman during the project.

I would also like to thank my family for the support they provided me through my entire life and in particular, I must acknowledge my friends, without whose encouragement and patience, I would not have finished this thesis.

In conclusion, I would like to thank all my teachers who guided me to improve myself with their great contributions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Complex networks are a powerful modeling tool, allowing the study of countless real-world systems. They have been used in very different domains such as computer science, biology, sociology, management, etc. Authors try to characterize them using various measures such as degree distribution, transitivity or average distance. Their goal is to detect certain properties such as the small-world or scale free properties. Previous works have shown some of these properties are present in many different systems, while others are characteristic of certain types of systems. However, each one of these studies generally focuses on a very small number of measures and networks. In this work, we aim at using a more systematic approach. We first constitute a corpus of 152 publicly available networks, spanning over 7 different domains. We then process 14 different topological measures to characterize them in the most possible complete way. We apply standard data mining tools to study correlation between the properties and identify which ones are discriminant or non-discriminant. An ANOVA completed by Tukey's test reveals two groups of domains can be distinguished in terms of average degree, modularity, transitivity and density. We apply cluster analysis tools to confirm these results, and find two more precisely defined clusters, in which the 7 domains are clearly separated (3 in a cluster, 4 in the other). An additional ANOVA confirms the previously mentioned measures are discriminant indeed, and additionally identifies diameter, average distance, closeness centrality, local transitivity and edgebetweenness centrality.

Keywords: Complex networks, clustering, ANOVA, Tukey's test, biomolecular networks, social networks, citation networks, computer networks, ecological networks, transportation networks.

RESUME

Les réseaux complexes sont un puissant outil de modélisation, permettant l'étude des innombrables systèmes du monde réel. Ils ont été utilisés dans des domaines très différents comme l'informatique, la biologie, la sociologie, la gestion, etc. Les auteurs tentent généralement de les caractériser en utilisant diverses mesures telles que la distribution de degré, la transitivité ou la distance moyenne. Leur but est de détecter certaines propriétés telles que les propriétés petit-monde ou sans-échelle. Des travaux antérieurs ont montré que certaines de ces propriétés sont présentes dans de nombreux systèmes différents, tandis que d'autres sont, au contraire, caractéristiques de certains types de systèmes. Cependant, chacune de ces études se concentre généralement sur un très petit nombre de mesures et de réseaux. Dans ce travail, nous utilisons une approche plus systématique. Nous constituons d'abord un corpus de 152 réseaux accessibles au public, s'étendant sur sept domaines différents. Nous traitons ensuite 14 mesures topologiques différentes, permettant de les caractériser de manière relativement complète, au regard des connaissances actuelles. Nous appliquons des outils d'extraction de données standard pour étudier la corrélation entre les propriétés et déterminer celles qui sont discriminant ou non discriminante. Une analyse de la variance complétée par le test de Tukey révèle deux groupes de domaines peuvent être distingués en termes de degré moyen, la modularité, la transitivité et la densité. Nous appliquons des outils d'analyse de cluster pour confirmer ces résultats, et nous trouvons deux plus précisément défini clusters, dans lequel les 7 domaines sont clairement séparées (3 dans un cluster, 4 dans l'autre). Une analyse de variance supplémentaire confirme les mesures mentionnées précédemment sont en effet discriminantes, et identifie en outre le diamètre, la distance moyenne, la centralité de proximité, transitivité locale et la centralité edgebetweenness.

Mots clés :

Réseaux complexes, le clustering, analyse de variance, Tukey, réseaux biomoléculaires, réseaux sociaux, réseaux de citations, des réseaux informatiques, des réseaux écologiques, réseaux de transport.

ÖZET

Sayısız gerçek sistemi, etkileşim halindeki düğümler ve aralarındaki ilişkileri kullanarak incelleyen karmaşık ağlar güçlü bir modelleme aracıdır. Bilgisayar bilimlerinden biyolojiye, sosyolojiden yönetim bilimlerine kadar pek çok farklı alanda kullanılmaktadır. Karmaşık ağ analizinde, uygulandığı sistemin ilk bakışta belirlenemeyecek özellikleri ortaya çıkarılmaya çalışılır. Bu kapsamda *derece dağılımı*, *geçişlilik* veya *ortalama uzaklık* gibi çeşitli ölçekler kullanarak ağ yapıları karakterize edilmeye çalışılmaktadır. *Küçük Dünya* etkisi ya da *Ölçeksiz ağ* gibi bazı karakteristik karmaşık ağ özellikleri bu şekilde tespit edilir. Önceki çalışmalar farklı alanlardaki karmaşık ağların benzer karakteristik özellikler taşıdığını göstermiştir. Ancak bunlar, genel olarak az sayıda ağ üzerinde ve az sayıda ölçekğe odaklanarak gerçekleştirimiştir. Bu çalışmada önceki çalışmalardan farklı olarak daha sistematik bir yaklaşım kullanılması hedeflenmiştir. Bu doğrultuda Bimoleküler, Sosyal, Ekolojik, Bilgisayar, Ulaşım, Alıntı ve İletişim olmak üzere 7 farklı alandan toplamda 152 tane yayınlanmış ağ toplanarak bir ağ havuz oluşturulmuştur. Havuzu oluşturan ağların, mümkün olan en detaylı şekilde karakterize edilebilmesi için 14 farklı topolojik ölçek kullanılmıştır. Ardından, standart veri madenciliği algoritmaları kullanılarak ölçümler arasındaki korelasyon belirlenmiştir. *Tukey* testi ile tamamlanan *Varyans Analizi* sonucunda havuzda bulunan ağ alanları iki gruba ayrılmıştır. Bu şekilde bir guruplaşmaya sebep olan ölçekler *ortalama uzaklık, geçişlilik, modülerlik* ve ıolarak belirlenmiştir. Bu sonucu doğrulamak için uygulanan kümeleme algoritmaları sonucunda da, önceki testi doğrular şekilde, 7 farklı alanın bir kümede 4(Biomoleküler, Ulaşım, Alıntı, Bilgisayar) ve diğerinde 3 (Sosyal, İletişim and Ekoloji) alan olacak şekilde 2 kümede gruplandığı gözlenmiştir. Kümeler ile yapılan ek bir Varyans Analizi de önceki analizin belirttiği ölçekleri doğrularken, bunu yanı sıra *ağ çapı, ortalama mesafe, yakınlık merkezi konumu*, *yerel geçişlilik* ve *bağlantı merkeziliği* ölçeklerini işaret etmiştir.

Anahtar Sözcükler: Karmaşık ağlar, kümeleme algoritmaları, Anova, Tukey testi, biyomoleküler ağlar, sosyal ağlar, alıntı ağları, bilgisayar ağları, ekolojik ağlar, ulaşım ağları.

1 INTRODUCTION

A complex system is a specific type of real-world system, i.e. a set of interacting elements relatively isolated from their environment, and possessing some emerging properties (Costa, Osvaldo et al. 2011). Such a property is not present at the level of a single element, but appears when considering the system as a whole. Its study consequently requires focusing on the interactions between the system elements. For this purpose, graphs are a very appropriate modeling tool, in which elements and their relations are represented by nodes and links, respectively. And indeed, they have been used as such in a number of domains such as computer science, physics, biology, sociology, etc.(Newman 2003). The graph representation of a complex system is called a complex network. Such a graph has non-trivial topological properties, due to the specific features of the complex system it represents. Concretely, this means complex networks differ from both regular and random graphs.

Graphs can be characterized by many different measures, each one reflecting some particular traits of the studied structure. One can cite the degree, the transitivity, the distance between nodes, the density, etc. Some of these measures have been used to detect certain properties, seemingly very widespread in complex systems. For instance, it is now well known that many complex networks are scale-free, meaning their degree is power-law distributed (Newman 2003). Many of them also possess the small-world property, i.e. the average distance between their nodes increases only logarithmically with the number of nodes (Orman 2010). Complex networks are also known to have a transitivity several order of magnitude larger than that of random graphs of the same size (Newman 2003). It is also very common for complex networks to display a hierarchical or a community structure (Newman 2003).

In the past, authors have focused on one or a few properties and studied them on networks representing a range of systems, with the purpose of showing their omnipresence. For

example, in (Watts and Strogatz 1998), Watts & Strogatz considered the transitivity and average distance in social, electrical and Biomolecular networks, and found out they all behave similarly. On the contrary, other studies tried to show some properties are characteristic only of a certain class of networks. For example, in (Lancichinetti and Fortunato 2009), Lancichinetti et al. observed different topological traits in community structures, depending on whether the considered data correspond to a biological, social, information, communication or computer network. These works highlight the importance of discovering regularities and discrepancies in complex networks topological properties. Indeed, these properties correspond to functional features. For example, a scale-free network is known to be sensitive to targeted attacks or failures, but resilient to random ones (Costa, Osvaldo et al. 2011). Topologically similar networks are therefore likely to represent systems with functional similarities, whereas network classes with specific topologic properties probably have unique functional features. However, existing works focused on a small number of networks and/or of properties. The network number limitation might be due to the difficulty of accessing data at this time. And for the focus on a few properties, this might be because those works were conducted to verify an a priori hypothesis. For example, one goal of Watts & Strogatz was to check if the small-world property was present also in non-social networks (Watts and Strogatz 1998).

In this work, we propose to adopt a systematic approach in the study and comparison of the topological properties of complex networks. First, it is now possible to retrieve many publicly available network datasets through the Web, which allows considering a number of different systems. Second, data mining techniques are able to consider a large number of properties simultaneously, and to automatically identify the relevant ones. By considering many of them at the same time, we can find how they relate, which is not possible when focusing only on a few of them. We first constituted a dataset of networks spanning several domains, which constitutes our first contribution. We processed the most widespread topological measures for these networks, and used them as feature vectors to characterize them. We then applied standard data mining tools to study them

depending on these features. Our second contribution is the analysis and interpretation of this outcome.

This study is organized as follows. In the second section, the notion of complex network and selected topological measures are presented. Section 0 describes the process we used to constitute our corpus, including data collection and their conversion to a unified format. In section 0, we examine the corpus, focusing on the considered domains, i.e. the type of real-world systems represented by the collected networks. We discuss them in terms of complex network representation and properties. In section 0, we describe our analysis methods, mainly preprocessing, clustering and post-processing tools. Section 0 is dedicated to the description and interpretation of our results. We conclude with a discussion of our work, its limitations and how these can be solved.

2 COMPLEX NETWORKS

In this section, we present some concept related to complex networks, which we use in the rest of the document. We first introduce some definitions regarding complex networks themselves, and then describe their main topological properties.

2.1 Definitions

Graph theory is a branch of discrete mathematics. However, graphs have been used to model all sort of real world systems, giving birth to the complex networks domain. In this section, we first give basic definitions of concepts related to graph theory, then define and illustrate the notion of complex networks.

2.1.1 Graph, Nodes and Links

A plain graph $G = (V, E)$ is constituted of a set of nodes V and a set of links E. When modeling systems, nodes generally represent the system elements, and links the relationships between them. The links are attached to the nodes, allowing to connect them. The number of nodes is $n = |V|$ and that of links is $m = |E|$. The neighborhood of a node u, noted $N(u)$ corresponds to the set of nodes directly connected to it. It is formally defined as:

$$N(u) = \{v \in V : e_{uv} \in E \vee e_{vu} \in E\} \tag{2.1}$$

Various properties can be added to the graph mathematical object (Newman 2003; Ghoshal 2009), as seen in Figure 2-1. Note these are not mutually exclusive, and can co-exist in the same graph.

Directions. Links are originally undirected, i.e. there is no difference between the two connected nodes (Figure 2-1-a, -b). However, it is possible to introduce such a distinction, and to consider one node is the source and the other is the target, resulting in a so-called directed link (Figure 2-1-c, -d, -f). Those can be used to represent asymmetric relationships, whether undirected links represent a symmetric relationship, or the absence of information regarding this aspect of the relationship.

Weights. Numerical weights can be associated to the links (Figure 2-1-b, -d). This allows representing relationships of different strength or intensity. Unweighted links represent the absence of such information, or the fact all relationships are similar (Figure 2-1-a, -c, -e, -f, -g, -h).

Multiple links. In plain graphs, there can be only 0 or 1 link between two given nodes. However, it is possible to relax this constraint and allow multiple links between two nodes, resulting in a so-called *multiplex* graph (Figure 2-1-f). This is convenient to model a system in which several independent types of relationships exist simultaneously.

Multipartite graphs. When the set of node can be partitioned in way such that links exist only between nodes of two different parts, then the graph is said to have a multipartite structure. In particular, if there are 2 or 3 parts, the graph is bipartite (Figure 2-1-g) or tripartite, respectively. By opposition, classic graphs are unipartite, i.e. there is only one part, with internal links (Figure 2-1-a, -b, -c, -d, -e, -f, -h). Multipartite graph are useful to model systems in which different kinds of nodes exist in the same network.

Dynamic graphs. It is possible to introduce a temporal dimension, by considering not a single graph, but rather a series of graphs corresponding to the different steps in the evolution of the dynamic graph. In terms of modeling, each one is a snapshot of the considered system. By opposition, classic graphs are called *static*.

Hypergraphs. Those are a generalization of the concept of graph, in which one link can connect more than two nodes (Figure 2-1-h) Such a link is therefore called a *hyperlink*. This kind of graph is useful to represent systems containing *n*-ary relationships between elements, by opposition to the classic binary relationships.

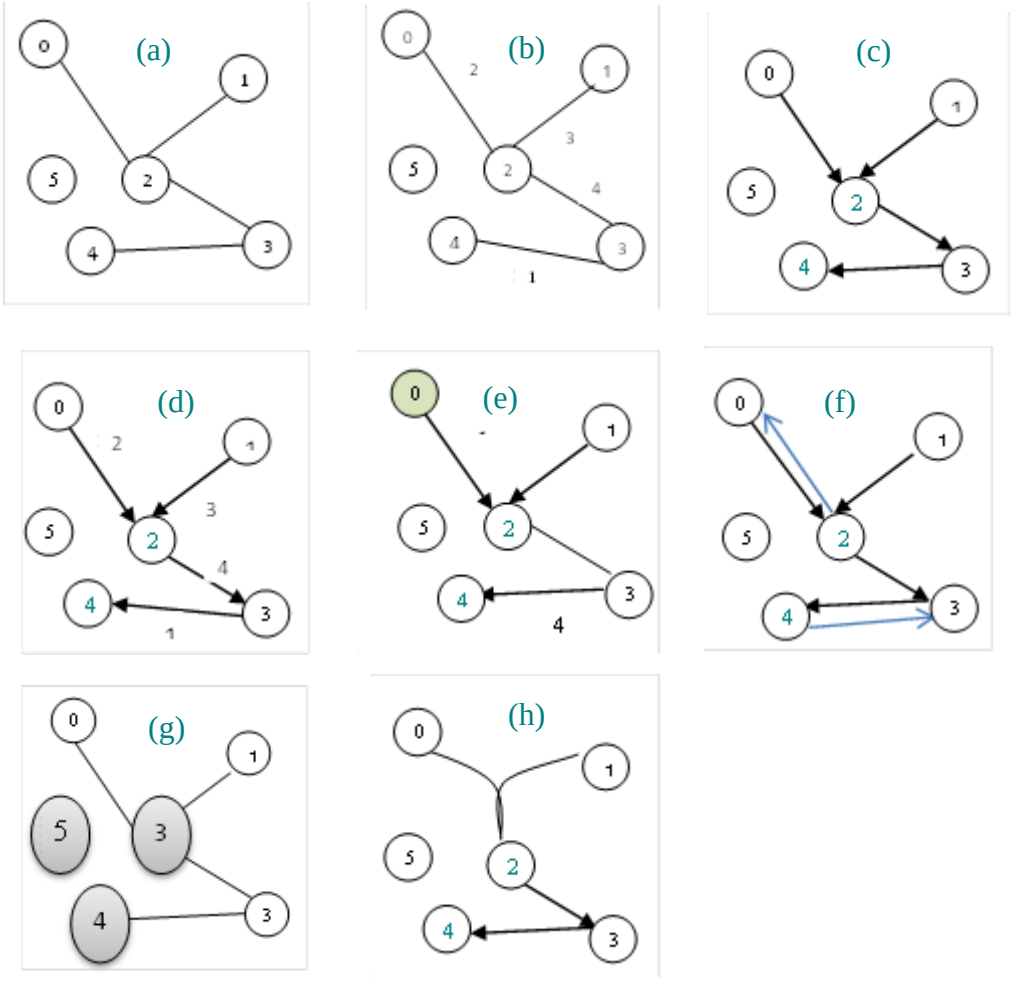

Figure 2-1 (a) Undirected network of 6 nodes; (b) Undirected weighted network of 6 nodes. (a) Directed network of 6 nodes; (b) Directed weighted network of 6 nodes. (d) Directed, weighted network of 6 nodes with different attributes. (f) Multiplex network. (g) Bipartite network of two types of nodes with link between different types. (h) Hypergraph.

2.1.2 **Adjacency matrix**

The structure of a graph can be represented by a so-called *adjacency matrix*. It indicates which vertices are adjacent to which nodes. A graph of n nodes is represented by a binary $n \times n$ matrix A, whose elements a_{ij} represent the link between nodes i and j: 0 if they are disconnected, and 1 if they are connected. Consequently, the diagonal is filled with zeros, provided the graph does not contain any loops (self-links). Figure 2-2 gives an example of such matrix.

For *undirected* networks, the matrix is symmetric, whereas it is asymmetric for directed ones. For *weighted* networks, the binary values are replaced by integers representing weights.

0	0	1	0	0	0
0	0	0	1	0	0
0	1	0	1	0	0
0	0	1	0	1	0
0	0	0	0	0	0
0	0	0	0	0	0

Figure 2-2 Adjacency matrix representing Figure 2-1

2.1.3 **Notion of Complex Network**

A *system* is a set of interacting elements, which are relatively isolated from their environment. A *complex system* is a system with *emerging* properties. An emerging property is something that one cannot observe at the level of the elements, but which is present at the level of the whole system (or possibly some of its subparts). In other terms: the system as a whole is more than the sum of its parts.

A certain number of properties, although not strictly needed, are generally observed in complex systems (Estrada, Fox et al. 2010):

- **Heterogeneity.** Interactions between objects are not distributed uniformly. In particular, not all objects interact together.
- **Locality.** Interactions are local; i.e. the system is generally not centralized.
- **Feedback.** Interactions constitute feedback loops, allowing a node to indirectly affect its own state at a later time, through a chain of interactions.

These properties, in turn, cause other properties to appear:

- **Segmentation.** Because of the heterogeneity and locality, groups of components appear, which interact more relatively to the rest of the system.

- **Hierarchy.** Each group can be in interaction with other groups, constituting groups of groups, and so on. This lead to several hierarchical levels in the system.
- **Fractal structure.** The objects can be complex systems themselves, and so on with their own components (e.g.: people in a social network).
- **Nonlinear behavior.** The presence of numerous feedback loops affects the dynamics of the system, and can lead to oscillations or other instable states.

Graphs representing complex systems are generally referred to as *complex networks*. They differ with *classic graphs* in the sense those are deterministic and regular (clique, lattices, etc.), and with *random graphs* in the sense their topology does not depend only on a random process. This relates to the properties displayed by complex systems concerning them including both chaotic and probabilistic mechanisms. The property of heterogeneity observed in the complex systems translates into what is called a *non-trivial topology* in the complex networks, i.e. neither regular nor completely random. Those properties are described in the following subsections.

2.2 Topological Properties

In this section, the topological properties used in this study are briefly described. We focus on the most popular ones in the complex network literature. Here, we distinguish *local* and *global* measures, i.e. those concerned with individual nodes or links, and those describing the network as a whole.

2.2.1 Local Properties

Degree. This nodal measure corresponds to the number of links attached to a node u, or in other words, to the size of its neighborhood $|N(u)|$. In real-world networks, it often follows a power law, leading to the so-called *scale-free property*. The degree $k(u)$ of a node u can also be formally defined using the adjacency matrix:

$$k(u) = \sum_{v \in V} a_{uv} \tag{2.2}$$

Distance. The *geodesic* distance $d(u, v)$ between two nodes u and v corresponds to the length of the shortest path between them. The distance distribution has been especially studied in the context of computer networks such as the Internet.

Eccentricity. This nodal measure $e(u)$ corresponds to the largest distance between a node u and any other node $v \in V \setminus u$ (Freeman 1978).

$$e(u) = max_{v \in V} d(u, v) \tag{2.3}$$

Betweenness centrality. This nodal measure $c_B(u)$ is the number of shortest paths going through a node u. In (2.4), σ_{vw} denotes the total number of shortest paths between nodes v and w, and $\sigma_{vw}(u)$ is the number of shortest paths between v and w going through u (Freeman 1978).

$$c_B(u) = \sum_{v < w \neq u} \frac{\sigma_{vw}(u)}{\sigma_{vw}} \tag{2.4}$$

Closeness centrality. This nodal measure $c_C(u)$ is the inverse of the sum of distances between the node of interest u and all the other nodes $v \in V \setminus u$. It quantifies how close a node is from the rest of the network, in average.

$$c_C(u) = \frac{1}{\sum_{v < u} d(u, v)} \tag{2.5}$$

Local transitivity. This nodal measure $C(u)$ corresponds to a ratio: the number of triangles including the node of interest u, to the number of possible triangles centered on this node. It can be interpreted as the probability for a link to exist between to randomly picked neighbors of the node of interest (Watts and Strogatz 1998).

$$C(u) = \frac{|e_{vw} \in E: v, w \in N(u)|}{k_i(k_i - 1)/2} \quad (2.6)$$

(Newman 2003)

Edgebetweenness. This measure $C_E(e)$ represents the number of shortest paths containing the link e. Links with high edge-betweenness centrality corresponds to bridge-like connectors between two parts of a network (Girvan and Newman 2001). In the following formula, $\sigma_{uv}(e)$ is the number of shortest paths between u and v containing the link e:

$$C_E(e) = \sum_{u<v} \frac{\sigma_{uv}(e)}{\sigma_{uv}} \quad (2.2)$$

2.2.2 Global Properties

Density. This global measure noted $\delta(G)$ corresponds to the ratio of existing to possible links in the network G. It ranges from 0 (no link at all) to 1 (all nodes are connected) (Orman 2010). Real-world networks are generally considered to be very sparse, with density close to 0.1.

$$\delta(G) = \frac{m}{n(n-1)/2} \quad (2.3)$$

Diameter & radius. The diameter $D(G)$ is the maximal distance between two nodes in G. It also corresponds to the maximum eccentricity over the network, i.e.:

$$D(G) = \max_{u \in G} e(u) \quad (2.4)$$

On the contrary, the radius $R(G)$ is the minimum eccentricity of the network (Dehmer 2011), i.e.:

$$R(G) = \min_{u \in G} e(u) \quad (2.5)$$

Transitivity. This measure, which is also called *clustering coefficient*, corresponds to the proportion of triangles in the network (Newman 2003). As such, it ranges from 0 (no triangles) to 1 (all possible triangles exist). It can be interpreted, when picking randomly a node, as the probability for two of its neighbors to be connected. According to the literature, the transitivity ranges from 0.1 to 0.8 in real-world network (Newman 2003). In (2.6) $\gamma(G)$ is the number of subgraphs with 3 links and 3 nodes (i.e. triangles) and $\tau(G)$ is the number of subgraphs with *at least* 2 links and 3 nodes (i.e. triangles and incomplete triangles).

$$T(G) = \frac{\gamma(G)}{\tau(G)} \tag{2.6}$$

Modularity. This measure assesses the quality of a community structure. It corresponds to the proportion of links located inside the communities, minus an estimation of the same quantity obtained for a null model. Consequently, its upper bound is 1 while 0 means the community structure is equivalent to a random one. Values observed in real-world networks possessing a community structure range from 0.3 to 0.7 . In the formulation of modularity, $\delta(u,v)$ is equal to 1 if nodes u abd v belong to the same community, and 0 otherwise (Newman 2006).

$$Q\,(G) = \frac{1}{2m}\sum_{u,v\in V}\left[A_{uv} - \frac{k(u)k(v)}{2m}\right]\delta(u,v) \tag{2.7}$$

Averages. Besides the mentioned properties, which are global by construction, we also consider as global properties the averages of the previously listed local properties: average distance, average local transitivity, etc.

.

3 CONSTITUTION OF THE CORPUS

This section describes the method used to constitute the corpus of complex networks which constitutes the basis of this work. Many different types of complex networks are publicly available, especially on the web, under very different forms and formats. Moreover, they represent very different systems, sometimes in different ways. Because of this high heterogeneity, the constitution of the corpus must follow a predetermined procedure in order to keep its quality high. Moreover, describing precisely this procedure will allow other persons to continue this work if necessary, therefore causing the corpus to grow while remaining consistent. Here, we first describe which form the corpus takes, and introduce various definitions used in the rest of this document. Then, we explain the procedure used to collect and format the suitable data. Finally, we review the different formats met to describe complex networks.

3.1 Properties of the Corpus

Our corpus is basically a database of networks. Its general description takes the form of a table (in an MS Excel file), and the networks themselves are stored in a normalized file structure. In this subsection, we first define the vocabulary we use when referring to the corpus. We then detail the information contained in the general description of the corpus, and then how the files used to store the network data are organized.

3.1.1 Corpus Related Concepts

In the context of this project, a few words have a very precise meaning. We define them here, in order to ease the understanding of the reader.

Source. A *source* is any means allowing retrieving a file representing a network. Most of the time, it is a web page. But the source can also be internal, when the data come

from some previous experiment of our research team, in which case we have a direct access to them. When the data are not publicly available through the web, but are nevertheless referred in some article, we might be able to get them from the article author, which constitutes a third type of source.

Dataset. The *dataset* corresponds to all the data originally retrieved from a source, for a given network. Each dataset has a specific and unique *id* in our corpus. It can be considered as the raw representation of the network. The dataset includes a data part and a meta-data part. The *data* is the information directly representing the network: network file, or other forms such as tables. The *meta-data* is all the information related to the network, but which is not the network itself: name, id, source, textual description, related bibliographic references, etc.

Network File. We call *network file* the normalized file representing a network and extracted from the data part of the dataset. Sometimes such a file is directly available from the dataset. Sometimes, the dataset contains some files representing the network under various other formats, in which case it must be converted to our normalized format. Finally, sometimes the data is not relational and the network must be extracted through a specific process. By extension, the id defined for the dataset also applies to the network extracted from this dataset.

Package. The *package* is a folder which contains both the dataset and the network file. In other words, it contains not only the original files retrieved from the source, but also the normalized network file. It also contains the R script used to extract the network file from the dataset (if such a script was necessary) and two files representing the topological properties of the network: one for global properties (transitivity, average shortest path, etc.) and one for local properties (degree distribution, node centrality, etc.). If the dataset is the object of an article, then the package also contains a PDF of this article. If its source is a web page, both the URL of the web page and a local record of it are placed in the package, in order to keep track in case of disappearance of the page. By extension, the id defined for the dataset also applies to the package containing this dataset.

Corpus. It refers to our whole database, i.e. the set of all packages. Each one is uniquely identified by the id of the dataset it contains.

3.1.2 General Description

We use a large table to summarize the datasets constituting the Corpus. Each dataset is characterized by its id, as mentioned before. A certain number of fields are associated to this key, in order to get a meaningful representation of each dataset. This table is meant to allow users to select datasets according to various criteria, and then apply an automatic processing on them. Certain of these fields describe the dataset in general; others are specific to the network extracted from the dataset. Table 3-1 gives a short description of all these fields.

Table 3-1 Fields of the table summarizing the corpus

Field	**Description**	***Type***
id	Unique identifier of the dataset	*Integer*
name	Name of the dataset	*Text*
n	Number of nodes in the network	*Integer*
m	Number of links in the network	*Integer*
directed	Whether the network contains directed links (Y), undirected links (N) or both kinds of links (B).	*Ternary*
weighted	Whether the network contains weighted links (Y), unweighted links (N) or both kinds of links (B).	*Ternary*
attribute	Whether nodal attributes are defined (Y), no attribute at all (N) or both (B).	*Ternary*
loop	Whether the network contains links	*Binary*

	between one node and itself.	
nodes	Nature of nodes	*Text*
links	Nature of links	*Text*
url	Web page of the dataset	*URL*
comments	Short description of the dataset	*Text*
main	Original format of the dataset	*Text*
date	Date the dataset was inserted into the corpus	*Text*
added by	Person who inserted the dataset into the corpus	*Text*
biblio	Bibliographic references associated to the dataset	*Text*

One of a representative example from Corpus is Oregon route-views datasets. Oregon route-views are published datasets in a web page related to a research. From id298 to id318 each dataset represents a network and each network differs from the others in terms of number of nodes and links. They are non-directional and non-weighted articles as well as there are not multiple links between nodes and neither loops in the network. Each dataset has inserted into the Corpus with the URL and the representation format which it is published.

Table 3-2 Example view of corpus

Id	**0298**	0324
name	Oregon route-views	high-quality biological processes maps PPIs
N	10729	537
M	22999	554
Dir	N	N
Weigh	N	N
Attr		N
Loop		Y
Nodes		Protein
Links		

url	http://topology.eecs.umich.edu/data.html	http://interactome.dfci.harvard.edu/C_elegans/index.php?page=download
comments	ASs according to the Oregon route-views	Protein-protein interactome network
Main	Edge list	Edge list
added	30.10.2011	07.06.2012
Added by	Burcu Kantarcı	BurcuKantarcı
references	(Chen, Chang et al. 2002)	

3.1.3 **Structure of the Packages**

As mentioned before, a package is a folder containing all the files related to a given network, including its dataset. Its name and structure are normalized, in order to allow subsequent automated processing.

The name is made up of the dataset unique id, followed by a dot and a string summarizing the network source. For instance in Table 3-2, for the dataset described in the first column, the corresponding package name would be `298.Oregon`. The content comprises two files:

- `_original.zip`: an archive containing the dataset and other related files;
- `network.graphml`: the network file itself.

The content of the archive is also normalized:

- `convert.R`: if a conversion script had to be applied in order to get the final network file, it is always named like this.
- `description.html`: if a web page describing the dataset is available, it is recorded locally using this exact name.
- `article.pdf`: if an article describing the dataset or its analysis is available, then it is recorded as a PDF file. If there are several articles, those are numbered chronologically.
- `...`: all dataset-files, whose names and number vary depending on the dataset. Note those files must contain the original version of the network,

in case the network file was obtained after from preprocessing or conversion. This allows going back to the original form in case of problem.

If there is only one source for the dataset, then all the above files are located directly in the root of the archive. If there are several sources, then each one is placed in an individual folder, which is itself located in the root of the archive.

3.2 **Procedure**

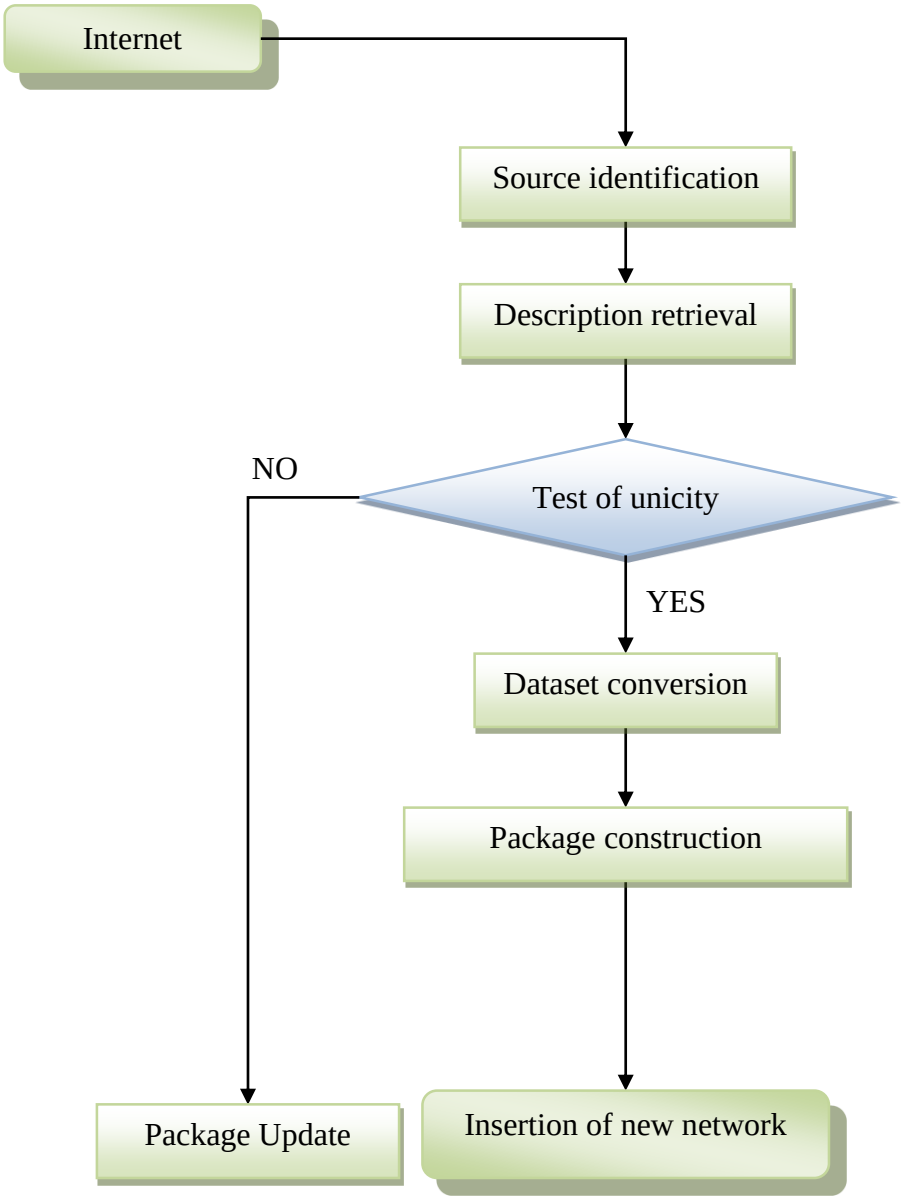

Figure 3-1 Flowchart of the corpus constitution

The procedure used to populate the corpus includes five steps, as described in Figure 3-1. The meta-data are retrieved, allowing determining if the dataset is already a part of the corpus. If it is not the case, then the data itself are retrieved and converted.

Source identification. The first step consists in identifying the source of the dataset. Depending on the type of source, the retrieval of the information will be slightly different. Indeed, the analysis of a full article will be more time consuming, whereas web pages are generally more synthetic.

Description retrieval. Thanks to the source, one can retrieve most of the meta-data, especially the meaning of the nodes and links, and more generally which kind of system the dataset represents. It is important to retrieve the meta-data first, because it is necessary to perform the test of unicity (cf. below).

Test of unicity. Each dataset present in the corpus must be unique. This step consists in using the description retrieved at the previous step and those already present in the corpus, in order to compare the new record and the older ones. The most important fields for that matter are the name of the dataset, the network sizes (numbers of nodes and links) and the type of system it represents. The bibliographic reference(s) associated to the dataset are also very relevant, because they allow characterizing the dataset in a unique way.

If the dataset is not already present in the corpus, then it is assigned a new id. It is possible to retrieve different networks extracted from the same raw data, but using different methods. In this case, both datasets are considered as different, but this relationship is indicated in the *comments* field. Some datasets correspond to the same system observed at different time steps. Then again, we consider these are different networks, but record their relationship.

Dataset conversion. As mentioned before; networks are represented in different file formats. The next step consists in transforming the original format into the normalized

format used in the corpus (when necessary). Sometimes, the dataset does not even contain a network file, in which case the network must be extracted from some raw data first. The various scripts used for these processes are included in the package for possible subsequent use.

Package construction. Once the network is available under the normalized format, it is possible to create the corresponding package. It contains all the files described in the previous section. Finally, the dataset and its description are introduced in the table summarizing the corpus.

3.3 Existing Network Formats

There are many file formats for the representation of complex networks. Each one has its own advantages and drawbacks. In this project, we favor several criteria, in order of decreasing importance:

- *Openness*: the format must not be linked to a specific proprietary tool;
- *Expressiveness*: the format must allow the representation of a wide variety of networks. The variety depends on the properties which a network possibly can carry. These properties are explained in the part 2.2.
- *Popularity*: There are many research made in complex network and it is important to use a format mostly used in this field.
- *Verbosity*: The format must clearly represent the network

In the rest of this section, we describe the main network file formats met while constituting the corpus and discuss them our criteria.

3.3.1 Edge List

This format focuses on links: each line in the file corresponds to a link of the network. The link is represented by citing both nodes, separated by some special character. The different kinds of separators met while constituting the corpus are presented in Table 3-3. One node can be represented by its number in the network (integer value) or by its name

(text). An example of the edge list format of a directed weighted network of 6 nodes is displayed in Figure 3-1 , which represents the network in Figure 2-1-b. As a neutral separator is used, the edge list can also be considered as an undirected network like in Figure 2-1-b. Then this information depends on the interpretation of the network. To summarize the only network which cannot be represented by edge list is the Figure 2-1 in which the nodes have some properties, between the example networks.

```
n0 n2 2
n1 n2 3
n2 n3 4
n3 n4 1
```

Figure 3-2 Edge list example

The content of an edge list file can be interpreted as directed or undirected links, but there is no way to encode this in the file itself, unless different separators are used, such as >, < and − are used. So it is necessary to know in advance if the considered network is directed or not. In the former case, each link is directed from the first node on the line towards the second one. Weights can also be encoded by adding a third (numerical) value. Alternatively, integer weights can be encoded by simply repeating the same link as many times as its weight.

The edge list format is simple and easy to format, so it is very widely spread. However, it suffers from some limitations. The first one is that the format allows mentioning only connected nodes: the isolates are absent of the list of links, since they are connected to none of them. The second one is its inability to let the user specifies link or node-related attributes. Finally, if it is possible to represent directions, it is not always explicitly indicated in the file itself.

Table 3-3 Types of separators met for the edge list format

Name	Character
Semi Column	;
Space	
Greater than	>
Smaller than	<
Hyphen	-

3.3.2 **Adjacency List**

In the adjacency list format, each line corresponds to one node of the graph and its direct neighborhood. The adjacency consists of all the nodes directly connected to the node of interest. Like for the edge list format, the nodes are represented by their index or name, separated by special characters (usually spaces). The central node is distinguished by its location (it is the first on the line) and sometimes it is additionally separated from the others nodes by a different character (e.g. a column `':'`).
The example below shows an example of the adjacency format corresponding to Figure 2-1-a. The network is simple, so the first nodes have only one node in their neighborhood. Only the last one has two nodes in its neighborhood.

```
n0: n2
n1: n2
n2: n3
n3: n2 n4
```

Figure 3-3 Neighborhood list example

The example below shows an example of the adjacency format corresponding to Figure 2-1-a. The network is simple, so the first nodes have only one node in their neighborhood. Only the last one has two nodes in its neighborhood.

3.3.3 **Graphml**

Graphml is an XML dialect designed to represent a large variety of networks (Brandes, Eiglsperger et al.). Besides the XML header, a Graphml file contains a root element `graphml`. It contains itself various references to the appropriate XML schema and namespaces, and two other types of elements use to define the network-related content. The first type allows defining node or link attributes, whereas the second type is dedicated to nodes and links themselves.

First, a sequence of `key` elements can be used to define several attributes. Of course, this is optional since not all networks possess attributes. In each `key` element, one has to specify an identifier, a name, a data type and a domain, using specific XML attributes. The identifier, declared through the `id` attribute, is used to refer to this attribute later in the file. The name, defined using the attribute `attr.name`, must be unique in the whole document. The data type is specified through the attribute `attr.type` and can take standard values: `boolean`, `int`, `float`, `long`, `double` and `string`. Finally, the domain is referenced by the `for` attribute. It represents the part of the network concerned by the attribute: `graph`, `node`, `edge` or `all`. The default value of the attribute can optionally be defined thanks to an additional `default` element.

A Graphml file can contain several networks, each one being represented by a `graph` element. Those are placed just after the `key` elements. Each one contains a series of `node` and `edge` elements, allowing describing the network topology. The network links can be directed or undirected. A default mode must be declared in the `graph` element, thanks to its `edgedefault` attribute. It takes two possible values: `directed` and `undirected`. The direction of each link can be specified individually, though. Like the `key` elements, `graph` elements can be identified by an `id` attribute, in case it is necessary to reference the network in a document containing several of them.

As mentioned before, the `graph` element contains `node` and `edge` elements. Those do not have to be placed in any specific order, but depending on the software accessing the file, it is generally recommended to define nodes first, and then links. Each `node` element represents a node. It has an `id` attribute which will be used later, when defining its incident links.

Each link is represented by an `edge` element. Like the other XML elements, it can be identified by an `id` attribute. It has to contain two compulsory attributes: `source` and `target`, which reference the identifiers of the two nodes corresponding to the outgoing and incoming nodes, respectively. If the network is not directed, then those attributes simply represent the two nodes indifferently connected by the link. The optional

`directed` attribute can take the values `true` or `false`. If the direction is not defined, then the default direction defined in the `graph` element is used.
The values of node and link attributes are defined though the `data` element, which is located inside the corresponding `node` or `edge` element. The `data` contains a `key` attribute, referring to an attribute declaration from the beginning of the document. The content of the `data` element must be consistent with this declaration.

The Figure 3-4 represents the Graphml source corresponding to the network from Figure 2-1-e. As it has combines all the properties of other networks, the rest of the networks in Figure 2-1-a-b, Figure 2-1-a-b can be also represented by Graphml. In this example a directed and weighted network of 6 nodes is defined. In the example, nodes have a string attribute called *color*, which is associated to the `key d0`, and whose value is *white* by default. The links have a double attribute called *weight*, associated to the `key d1` and whose default value is 0.0.

The first node of this example has the color *green*, specified by the help of *key d0* in its `data` element. All links are directed except the one between the second and third nodes. Indeed, in the declaration of the graph the default link type is set to directed, but for this specific link we assigned the value `false` to the *directed* attribute in the corresponding `edge` element. The links between nodes 0 and 2 and between nodes 3 and 4 are weighted and by the help of key `d1` in their respective `data` elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
   mlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
   http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
   <key id='d0' for='node' attr.name='colour' attr.type='string'/>
      <default> white </default>
   </key>
   <key id='d1' for='edge' attr.name='weight' attr.type='double'/>
      <default> 0.0 </default>
   </key>
   <graph id='G' edgedefault='directed'>
      <node id='n0'/>
         <data key='d0'>green</data>
      </node>
      <node id='n1'/>
      <node id='n2'/>
      <node id='n3'/>
```

```
      <node id='n4'/>
      <node id='n5'/>
      <edge source='n0' target='n2'/>
         <data key='d1'>1.0</data>
      </edge>
      <edge id =" e2" directed ="true" source='n1' target='n2'/>
      <edge directed="false" source='n2' target='n3'/>
      <edge source='n3' target='n4'/>
         <data key='d1'>4.0</data>
      </edge>
   </graph>
</graphml>
```

Figure 3-4 Graphml file representing the Figure 2-1-e

Additionally, other kinds of graph properties can be represented with Graphml, which supports hyperlinks, nested graphs and node ports declarations. As there is no such type of networks in the corpus, this kind of properties are not relevant for us.
Because XML syntax is very verbose, the size of a Graphml file can be much larger than for most other formats. However, the same syntax can be expanded by new attributes, and offers a very expressive format. Moreover, this language is independent from any tool. These arguments are the main reasons why Graphml is used during this project.

3.3.4 **Gexf**

GEXF (Graph Exchange XML Format) is an XML-based language used for describing networks (Heymann 2009). It lets the user specify nodes and links, as well as user-defined attributes such as node weights or link directions. It was defined to be used as an interchange format between graphing applications (Heymann 2009). Generally it shows the same properties as Graphml for XML elements although there are still some differences in their grammars.
Besides the classic XML header, the first GEXF element is `meta`, which allow indicating some metadata concerning the network, such as the date of last modification of the document.

Each network in a GEXF file is then represented by a `graph` element. Its attribute `defaultedge` allows setting the default link direction: either `directed`,

`indirected` or `mutual`. This element contains `attributes`, `nodes` and or `edges` elements, which contain the corresponding parts of the network.

Node and link attributes are defined separately using two distinct `attributes` elements. The attribute `class` of `attributes` allows indicating which type of component is concerned: `node` or `edge`. Each attribute is defined using an `attribute` in `attributes`. It is characterized by a unique identifier (amongst attributes) defined though the attribute `id`. The attribute `title` is used to specify the name of the attribute, and `type` its data type, which must be XSD-compliant. A default value can be specified by including a `default` element in attribute.

As mentioned before, each node is declared in the `nodes` element of `graph`. A `node` element defines a node, using an attribute `id` to specify a unique identifier (amongst nodes). The node description is given through the `label` attribute. Attributes values are defined by including an element `attvalues` in `node`. This element contains as many `attvalue` elements as there are attributes. Each `attvalue` elements has a `for` attribute, whose value correspond to the concerned attribute `id`, and a `value` attribute, whose value is the attribute value.

Each link is represented by an `edge` element defined in the `edges` element of `graph`. Like for nodes, each link must have a unique identifier (amongst links), defined through the attribute `id`. The attributes `source` and `target` define the two nodes connected to the link, using the nodes identifiers. Self-links (a node connected to itself) are allowed. The attribute `type`, similar to `defaultedge` in `graph`, allows defining the link direction. An optional `weight` attribute allows defining a real-valued weight for the link. Attributes values are defined exactly like for nodes, using `attvalues` and `attvalue` elements.

The example below is the Gexf representation of Figure 2-1-e which summarizes the GEFX properties. In the example a directed and weighted network of 6 nodes is defined. The nodes can take color attributes while the links can have weights. Additionally a life

time is given to the network with the dynamic attribute therefore the life time of the graph is specified.

```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.2draft"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.gexf.net/1.2draft
http://www.gexf.net/1.2draft/gexf.xsd" version="1.2">

<meta lastmodifieddate="2012-10-29">

<graph mode="dynamic" defaultedgetype="undirected"
timeformat="date">
    <attributes class="node" mode="dynamic">
        <attribute id="0" title="color" type="string"/>
             <default>white</default>
        </attribute>
     </attributes>
    <attributes class="edge" mode="dynamic">
         <attribute id="1" title="indegree" type="float"/>
             <default>0.0</default>
    </attributes>
    <nodes>
        <node id="0" label="Gephi" start="2009-03-01">
            <attvalues>
                <attvalue for="0" value="green"/>
            </attvalues>
        </node>
        <node id="1" label="Gephi">
        <node id="2" label="Gephi" start="2009-03-01">
        <node id="3" label="Gephi" start="2009-03-01">
        <node id="4" label="Gephi" start="2009-03-01">
        <node id="5" label="Gephi" start="2009-03-01">
     </nodes>
     <edges>
       <edge id="0" source="0" target="1" start="2009-03-01"/>
            <attvalues>
                <attvalue for="1" value="0.1"/>
            </attvalues>
       </edge>
       <edge id="1" source="0" target="2" start="2009-03-01"
end="2009-03-10"/>
       <edge id="2" source="1" target="0" start="2009-03-01"/>
       <edge id="3" source="2" target="1" end="2009-03-10"/>
       <edge id="4" source="0" target="3" start="2009-03-01"/>
           <attvalues>
                <attvalue for="1" value="0.4"/>
            </attvalues>
       </edge>
      </edges>
</graph>
```

Figure 3-5 Gefx representation of Figure 2-1-e

Gexf also allows representing dynamic networks, community structures and hierarchical structures. However, this kind of information is not provided with the networks we are studying in this work. Community structures will be estimated for each network, but will

be stored separately form the network file itself, in order to be easily evaluated and analyzed. For these reasons, we will not describe further these two Gexf features.

3.3.5 **Pajek**

Pajek is a free software program used for analyzing networks. The file format pajek contains nodes definitions as vertices, directed link definitions as arcs and undirected link definitions as edges.

Representation of the graph has three regions. The regions *`Vertices` represent a list of nodes with their attributes as name color, lent etc. After the declaration of *Vertices the number of nodes are given. As the definition of nodes ends the *`Arcs` definition starts which expresses the directed link list. Additionally to the links there can be link related attributes mentioned at the same time. Finally the list of undirected links are given under the `*Edges` part, with the link related attributes as well.

Figure 3-6 represents the graph at Figure 2-1. The nodes are listed at the beginning of the document under the `*Vertices` title with their attributes. As there are `6` nodes at the graph it is indicated after the tag. The node 0 has an attribute as Green and it is added to its declaration part. The `*Arcs` part represents the directed links at the graph and the first link has an attribute as "1" so that it is indicated at the end of the link representation. The `*Edges` are the list of undirected links. Obviously it is possible to represent both directed/undirected and weighted/non-weighted graph by using Pajek file format. Both Figure 2-1-a and Figure 2-1-c could be represented by using this format.

```
*Vertices 6
1 "0" Green
1 "1"
2 "2"
3 "3"
4 "4"
5 "5"
*Arcs
1 2 "1"
2 3
3 4
*Edges
2 3
```

Figure 3-6 Pajek file format example of Figure 2-1-a

It is not possible to represent dynamic or hierarchical networks by using Pajek but as there is not such kind of networks in this project corpus it is not an important point for the choice of the format.

3.3.6 **Matrix Market**

Matrix market is a file format which saves the data in binary format.

3.4 **Corpus Format and Conversion**

As explained in section 3.3, we defined four criteria the file format of the network should provide. Firstly, the format must not be linked to a specific tool (*openness*). Secondly it should ensure that a large variety of networks can be represented (*expressiveness*). Third, the format should be widely used in this field, so that many tools can take advantage of our corpus (*popularity*). Fourthly, the way networks are represented must be compact, so that the corpus does not take too much storage space (*verbosity*).

Edge list is one of the most popular network file formats and it is not related to a specific tool. Beside the fact that it is not also a verbose file format, it is not possible to represent some kind of networks by using edge list. This means that edge list provides the openness, popularity and verbosity criteria but expressiveness is still missing.

Adjacency list is the other file format which is independent from any specific tool. This format is also very far from verbosity. However It is not enough expressive and it is not a quite popular format. As results it provides just two of the criteria which are openness and verbosity but it rests expressiveness and popularity.

GEFX is a XML file format and it is possible to represent a huge variety of networks by using GEFX. GEFX is mostly linked to Grephi, a Java program, and is not used other

tools. It provides the criteria expressiveness and popularity but it is not popular and as it is an XML file it is verbose.

Pajek is a file format depending on free software called Pajek. It is quite popular and expressive but it is not independent from software. Moreover, several types of file coexist depending on the type of data to be represented, which makes it difficult to use.

Matlab matrix is a file format supportad by Matlab and Matrix Market. Therefore it is not independent from software. It is not expressive enough and not easy to treat from other softwares. As a result it is not open, nor expressive and it is also not readable without any conversion.

Graphml is open, as it is an XML language. Beside the verbosity, it is expressive which allows the representation of much type of networks. Another important point is that Graphml is a popular format used in many graph analysis packages and tools. As a result, we chose Graphml as the main file format in this project.

In each package, the network file is compliant with the Graphml format. Because of this choice, any other format must be converted to Graphml, in order to produce a normalized package. This is not a trivial task, because there are many different formats to represent networks. In the rest of the section while the reason incapability of other formats are discussed the conversion techniques will be explained.

In many of the conversion techniques, we use the R language and its package iGraph. R is a free open source platform and a language for statistical studies, whereas iGraph is a free open source package for R and Python, programmed in C, and allowing handling graphs. iGraph can load and record graphs using various formats, and is also able to perform various processes and transformations on them. In particular, it is able to generated files respecting the Graphml format.

Table 3 Comparison between formats.

	Edge list	Adjacency list	Graphml	Gexf	Pajek	Matlab Matrix

Openness	Yes	Yes	Yes	No	No	No
Expressiveness	No	No	Yes	Yes	Yes	No
Verbosity	Yes	Yes	No	No	No	No
Popularity	Yes	No	Yes	No	Yes	No

3.4.1 **From Edge list**

We use the iGraph library to load data at the edge list format. We get a network object which we then record at the Graphml format using the `read.graph` function.
However, not all variants of the edge list format are supported. For the non-supported variants, we manually load the content of the file using R basic functions, which are more flexible. The resulting table or matrix can then be turned into a graph object thanks to the method `graph.edgelist` of iGraph, and finally be recorded as a Grahpml file.

3.4.2 **From Adjacency list**

It is possible to load its content in order to build an adjacency list by using iGraph. Then, the iGraph method `graph.adjlist` can be used to obtain a graph object. This object can then directly be recorded as Graphml. However the latest version of iGraph allows to directly loading adjacency lists, which is even faster.

3.4.3 **Gexf**

Gexf refers man additional advantages like these files can be read and written using a Java library called gexf4j.

3.4.4 **Pajek**

By using iGraph, it is possible to load and convert the main Pajek format into Graphml file format. By using the `read.graph` function of iGraph, giving `pajek` as the `format` parameter, the graph is loaded. By using the `write.graph`, function the graph is saved using the Graphml format.

3.4.5 **Matrix Market**

iGraph is not able to process this format. We first use Scilab and MatrixMarket package on Matlab, the file in matrix market format can be read. Scilab provides a conversion function to convert sparse matrix into adjacency matrix. By using sp2adj function a sparse matrix is converted into adjacency matrix and by using MatrixMarket write function, the adjacency matrix is written into a file.
After taking the adjacency matrix it is possible to convert it into Graphml by using R, igraph functions.

4 DESCRIPTION OF THE CORPUS

Any discrete system can be represented virtually by complex networks owing to their natural suitability. By the way the field of complex networks fits to deal with the nonlinear phenomena and by using graph theory, large and complex structures can be analyzed by statistical and mathematical methods so that the results represents a property of a whole system.

During the researches of this project many different networks from different domains are processed. In each domain of study networks carry some common points and each scientific area has their characteristic features which affect the nature of the corpus and the results of the classification processes which run on corpus.

4.1 Biological Networks

Biological Networks are studied in four groups: Biological, Medicine, Ecology, and Neuroscience. As there are Biomolecular and Ecological Networks in the corpus processed in this project in this section this two network domains are examined.

4.1.1 Biomolecular or Interactome Networks

An interactome network represents all or some of the molecular interaction taking place inside a biological cell. In the literature, they are often represented under the form of complex networks. One can distinguish three distinct scales. First, the *genetic regulatory network*, which represents how proteins are generated as the expression of genes. Second, at a higher level, the *protein-protein interaction* network, which models how the generated proteins interact with each other. Third, the *metabolic reaction network* and the *signal transduction network*, which represent how these interactions are

chained to allow generating and destroying molecules for the former, and inter-cell communication for the latter.

Living organism are dynamic systems, so in these networks, the nodes and links evolve with time. For instance, in a protein-protein interaction network, several proteins can interact to form a new one, which did not exist before and might not exist later because of the absence of certain factors needed for its production. However, in many studies, the network is considered at a given time step, because the available databases remain incomplete.

4.1.1.1 **Protein-Protein Interaction Networks**

A *protein* is a molecule whose role is essential in the functioning of biological cells. It can be located inside or outside the cell. It is defined as a connection of one or several *amino acid* chains. The sequence of amino acids constituting the chain directly depends on the DNA of the considered organism: this sequence is encoded in a specific *gene*, under the form of a sequence of nucleotides. *Nucleotides* are the molecules combined to build larger molecules such as DNA and RNA (adenine, guanine, thymine, cytosine, etc.).

What makes proteins important is their ability to *bind* with other molecules. This ability depends on the spatial configuration of the protein, which makes specific *binding sites* appear. These are sorts of pockets with a specific shape, into which parts of other molecules can fit, therefore constituting a physical (by opposition to chemical) connection. The specific shapes of the binding sites and bound molecules allow targeting specific molecules. Proteins can bind with other proteins or smaller molecules. In the former case, they form molecular machines called *protein complexes*, which is a way of achieving a particular biological function. The main *biological functions* are:

- *Catalytic*: the complex is an enzyme which accelerates some chemical reaction.

- *Signaling*: the complex is either an extracellular molecule acting as a message sent to another cell, or a receptor, i.e. a membrane protein able to react to a signal and induce a chemical reaction inside the cell.
- *Transporting*: the complex is able to transport a smaller molecule (called ligand).
- *Structural*: give rigidity to the cell.

A protein can belong to several complexes, and the function implemented by the complex depends strongly of its context: location in the cell (space), stage of the cell lifecycle (time), chemical state of the cell, etc. This makes it difficult to determine exactly if two proteins interact or not, and the available data are generally stochastic (i.e. under some conditions, both molecule interact with a certain confidence). Many different experiments and statistical operations are required to identify the protein-protein interaction network for a certain organism.

In a *protein-protein interaction network*, nodes represent proteins and links represent possible bindings between them. As the binding is reversible, the links are not directed. Moreover, bindings cannot be distinguished in terms of importance, so links are generally unweighted. In some studies, the confidence one has on the binding is introduced under the form of links, though (Costa, Osvaldo et al. 2011).

Protein-protein interaction networks have the small-word property, with a small average shortest distance and large average transitivity (Costa, Osvaldo et al. 2011). They mostly have a hierarchical structure and a heavy-tailed degree distribution. The main structure is designed like a hierarchical composition of densely connected large groups, connected via generally hub proteins as seen in the Figure 4-1. The hub proteins are believed to be the oldest proteins in the network that the disconnection of this heavily connected node cause the breaking up of network in disconnected groups.

Figure 4-1 Picture representing a typical protein network (Gursoy, Keskin et al. 2008)

The extraction and study of protein-protein interaction networks is a part of a field called *proteomics*. The goal is to model complete networks, in order to understand how living organism functions, the protein interactions being the basis of this functioning. One of the most important topic in this field is the identification of protein complexes and their roles. This relies heavily on the task of community detection: complexes correspond to densely interconnected subgraphs in the protein-protein interaction network.

In the corpus, the networks whose id is between 318 and 322 are several representative examples of protein-protein networks used in this project.

4.1.1.2 **Metabolic Networks**

Metabolism is the set of chemical reactions which allows living cells to function by breaking down (*catabolism*) and building up (*anabolism*) biomolecules. A *metabolic pathway* is the specific sequence of chemical reactions occurring inside a cell and allowing to transform a certain initial molecule, called *substrate,* into something different called *product*. The term *metabolite* refers to both substrates and products. The

mentioned chemical reactions are caused by enzymes, i.e. protein complexes acting as catalysts. This highlights the relation with protein-protein interaction networks.

A *metabolic network* gathers one or several metabolic pathways. A number of forms can be found in the literature. In the most informative case, metabolites, enzymes and reactions are represented as three different types of nodes in a 3-mode tripartite network (Costa, Osvaldo et al. 2011). The links are directed and weighted. They stand for mass flows from reactants to reactions, and for catalytic reactions from enzymes to reactions. Some authors prefer to use 2-mode bipartite networks (Zhao, Yu et al. 2006), in which the two types of nodes represent metabolite and enzymes, respectively. The links connect the substrates to the enzymes able of processing them, and enzymes to the products they generate. Alternatively, some authors prefer to represent reactions instead of enzymes (Costa, Osvaldo et al. 2011).

This 2-mode network can be projected on either one of both dimensions, leading to two other kinds of networks. In the first, the nodes correspond to metabolites and the links to enzymes (or reactions) as represented in the Figure 4-2 (Zhao, Yu et al. 2006; Fortelny 2010) . The directed links connect one substrate to the corresponding product, through the appropriate enzyme (or reaction).

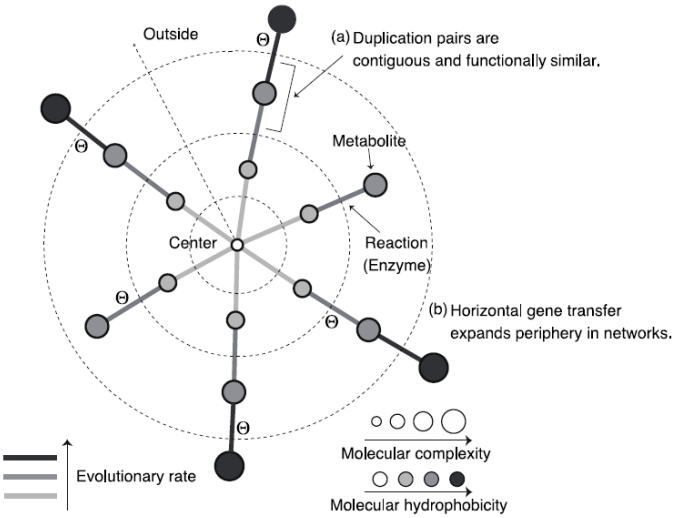

Figure 4-2 Representation of metabolic network of a reaction where the nodes are the metabolites and enzymes are the links (Takemoto 2012)

In the second type of projection, the nodes are enzymes (or reactions) and the links are metabolites (Zhao, Yu et al. 2006; Fortelny 2010). Two reactions are connected if they share a metabolite: it is the product of the source node and the substrate of the target node (Costa, Osvaldo et al. 2011).

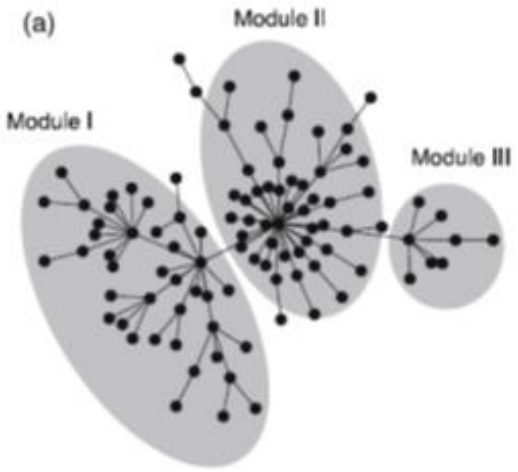

Figure 4-3 Power law representation of metabolic networks (Takemoto 2012)

Research indicates that whatever the considered form, metabolic networks show power law degree distribution (Costa, Osvaldo et al. 2011; Takemoto 2012). The diameter increases logarithmically when new nodes are added, approaching a constant value. This addition corresponds to an increase in the complexity of the considered metabolism. There can be hub nodes, whose removal ends with the separation of the network in several components. Moreover, research shows that the clustering coefficient of metabolic networks evolves independently from the network size, following a low of the form $c(k) \sim k^{-1}$. Finally, these networks are known to have a hierarchical and modular organization structure

Network id 324 is an example from corpus which represents a metabolic network.

4.1.1.3 **Signal Transduction Networks**

As metabolic networks focus on catalytic protein complexes, signal transduction networks are concerned only on the signaling function of these molecules. The goal here is to study how information flows through cells.

A cell receives information through a specific protein called *receptor*. It can be located either on the cell surface (transmembrane receptor) or inside the cell (intracellular receptor). This protein is activated when a compatible molecule appears and can bind with it: hormone, neurotransmitter, etc. The result of this activation is either a direct effect on the cell behavior, or an indirect effect mediated by intermediary protein interactions. The sequence of reactions allowing the communication between one cell and another is called a *signaling pathway*. A signal transduction network is a set of possibly overlapping signaling pathways.

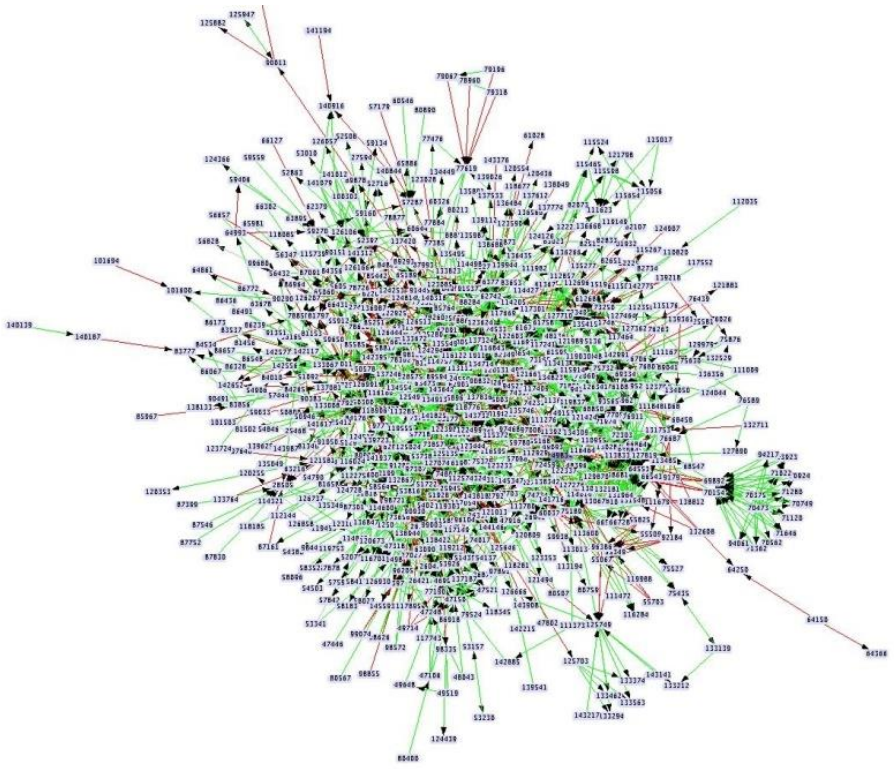

Figure 4-4 SIMPathway Signal Transduction Network

4.1.1.4 **Gene-Regulatory Networks**

The production of a protein is a complex process. As mentioned before, the sequence of amino acids composing a given protein is encoded in a specific gene. The *transcription* of a gene consists in generating an mRNA molecule, which can be considered as a mobile copy of the information contained in the gene. This *messenger RNA* is then passed to the ribosome, located in another part of the cell. The *ribosome* is a molecular machine, in charge of performing the *translation* process, i.e. interpreting the mRNA in order to generate the corresponding protein.

The transcription step necessitates various proteins or protein complexes able to bind with the DNA, called *transcription factors*. They allow selecting which genes should be or should not be transcripted at a given moment. In other words, they perform a *regulation* of gene transcription, by acting as *repressor* or *activator* during this this process. Therefore, it is possible that a newly generated protein will activate the transcription of a gene, leading to the production of a new protein, itself able to trigger another gene, and so one. The result is a so called *regulatory cascade*. Moreover, the regulation process can include several cells thanks to communication mechanisms of larger scale, such as signal transduction.

In a *gene-regulatory network*, also called transcriptional regulatory network, nodes and links represent genes and transcription factors, respectively (Schlitt and Brazma 2007; Costa, Osvaldo et al. 2011). The links are directed from a gene encoding some protein, to a gene for which this same protein acts as a transcription factor (i.e. it triggers its expression).

As not all genes are active at a given time, the structure of transcriptional regulatory networks changes with time or environmental conditions. The structure can undergo three different changes: (i) duplication of the transcription factor, which results in both copies regulating the same gene, (ii) duplication of the target gene with its regulatory region, where the target gene will be regulated by the same transcription factor, and (iii) duplication of both the factor and the target. This results in the presence of motifs, and specific connectivity distribution in the network.

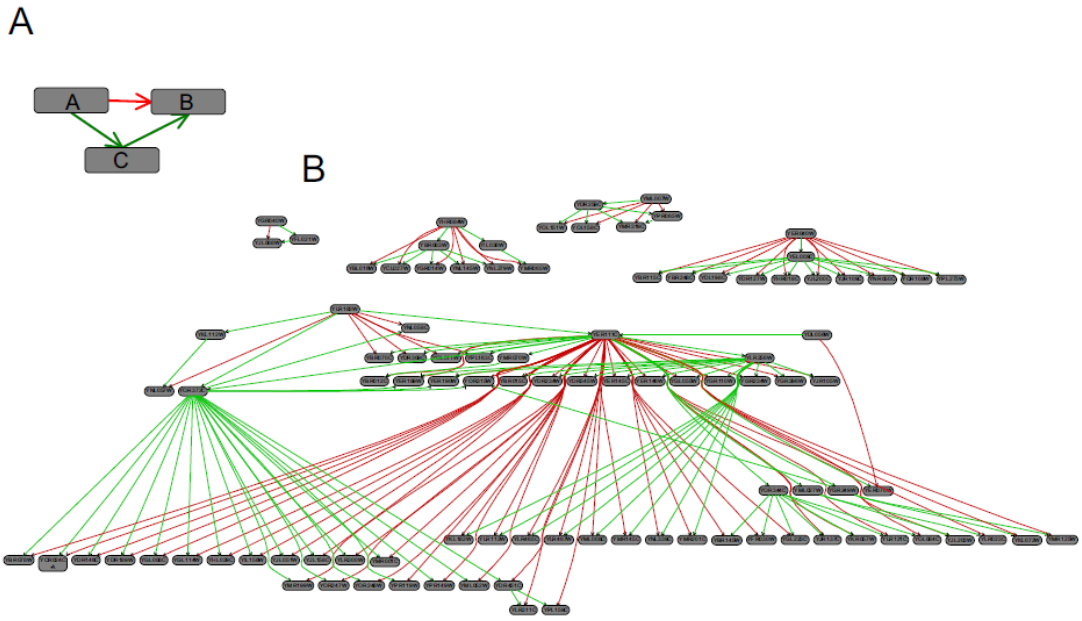

Figure 4-5 Representation of several genetic reaction while constructing B gene (Schlitt and Brazma 2007)

The distribution of connectivity of transcriptional regulatory networks shows two main properties: (i) the incoming degree distribution decreases exponentially, and (ii) the outgoing degree distribution follows a power law. The exponential character of the incoming degree indicates that most TGs are regulated by a similar number of factors while the scale-free distribution of the outgoing degree points to a few TFs participating in regulation of a large number of TGs.

Networks whose id's are between 329 and 331 are the representatives of genetic networks.

4.1.2 **Ecological Networks**

Ecology is a branch of science which studies the relation between living organisms their environment. Ecological networks consist of populations, natural areas, food web, population growth, energy consumption and biotic diversity. The most popular subject of ecologist is Food webs in recent years and there are large studies made on food webs (Odum and Barrett 1953; Costa, Osvaldo et al. 2011).

Therefore in terms of food webs, in ecological networks species represents the nodes and interaction between nodes is the links. There can be different type of interactions which mainly contains competition, parasitism and mutualism. The linkage between nodes is construed if a species I eat the other species J in a food chain. Therefore ecological networks are directed in general (Albert and Barabasi 2002).

Food webs can be studied in terms of modeling and type of linkage. Firstly food webs can be divided in three levels which are static models, dynamic models and species assembly and evolutionary models. Secondly as it is said before there can be a relation mutualism, competition relation, parasitism or predator-prey relation. It is known that mutualism relation networks are more nested and more connected than the other type of ecological networks (Costa, Osvaldo et al. 2011).

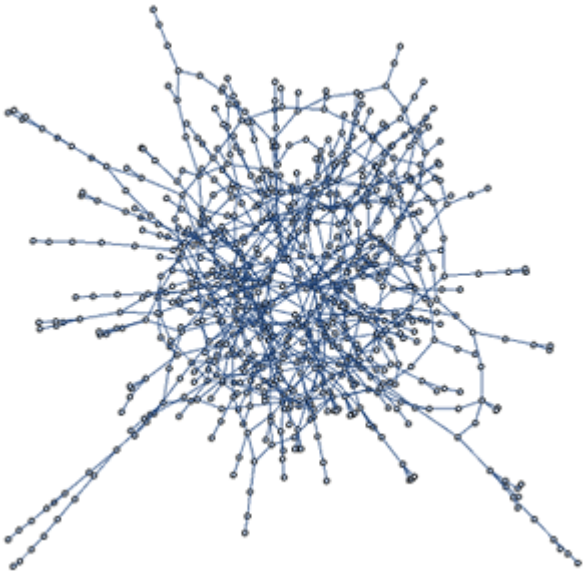

Figure 4-6 A random network of mutualistic networks with 1000 nodes and 1000 links(Zhang, Hui et al. 2011)

Although the food webs differ from each other in terms of number of nodes and average degree it is known that they have small size and small world property. It is also observed that they respect a power law of a small degree (Costa, Osvaldo et al. 2011). Moreover it is widely observed that the average degree is mostly three of fever. The existence of

key species in environment points the existence of hub nodes in food webs (Albert and Barabasi 2002).

4.2 Social Networks

Social networks refer to social relations which contains a grand number of complex network variations. There are many sub-network domains related to social networks like Personal Relations, Sport, and Movie Actors etc. ın this project the type of network which is used is Personal Relations which is presented in this section.

4.2.1 Personal Relations

Personal relations are almost the oldest network type in terms of sociology. People may contact to others in different ways which divide personal relation networks in several classes as acquaintances, trust, sexual, email, professional etc. The corpus used in this project contains acquaintances, trust and sexual relation networks which are studied in this section.

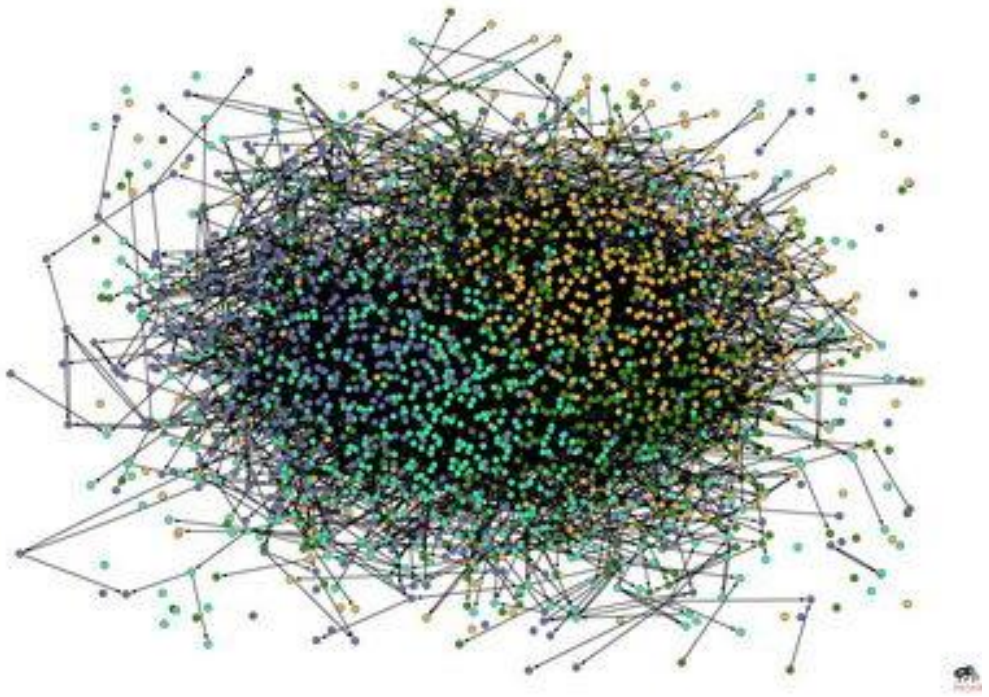

Figure 4-7 A high school's empirical friendship network(Gonzalez, Lind et al. 2006)

4.2.1.1 **Acquaintances**

Acquaintances networks are based on the Milgram's famous social experiment, "six-degree of separation" concept. According to the experiment which depends on the separation by six intermediate individuals on average of two random chosen people. Acquaintances networks then consist of nodes as people and their relationships in-between as links. The research shots that the degree distribution respects a high power law coefficient and this kind of networks has a high level of clustering.

When the structure is observed by node removing, the removal of most connected node has local effects within the communities inversely the removal of weekly connected nodes causes' loss of communication between communities.

The clustering coefficient increases over the time and the size of cluster decreases while the relation become stronger between people.

Another behavior of acquaintances network depends on the nodal attributes which points a homophile in terms of these attributes like gender, age or nationality etc.(Costa, Osvaldo et al. 2011)

4.2.1.2 **Trust**

Trust networks can be seen as a sub-network of acquaintances network in which there exist stronger connections between persons. This kind of network can be generated by using Pretty Good Privacy algorithm which let one person certify another by sharing his public encryption key.

Each person represents the nodes and links are the connection between. There exist two types of degree in this type of networks. In-degree is the number of person which shares key with the person and the out-degree is the number of person which this person shares

his public key with. The analysis shows that there exists a power law degree distribution in both directions.

The clustering coefficient is generally independent of the component size as there exists strong connections(Costa, Osvaldo et al. 2011).

4.2.1.3 **Sexual Relations**

Sexual relation networks are also sub-networks of acquaintances networks which consist of woman-men relations.

In sexual relation networks woman and men are represented by nodes and the relations in-between are the links. The average degree of man is larger than woman. On the other hand the degree distribution of both type of nodes respect a power law with a quite similar value.

Nodes may have attributes as personal properties and skills and analysis shows that the degree distribution is under influence of node properties. This also leads a clustered structure in the network(Costa, Osvaldo et al. 2011).

4.3 **Citation Networks**

Citation is a reference to a published or unpublished source made from a source. Scientific papers, research made in many different area points other previous papers, researches or resources and this generates a growing network.

The published papers are represented by nodes and the reference running from a paper to another is called as a directed link. There exist two types of degree in citation networks which are in-degree and out-degrees. In-degree represents the reference from other papers which points the importance of paper as it refers to the number of appearance the paper in other papers. Despite the fact that in-degree is value which can increase by the time, out-degree represents the reference made in the paper therefore once the paper is

published the value is fixed(Costa, Osvaldo et al. 2011) . While the in-degree respects a power law with a high coefficient, the out-degree distribution has an exponential tail(Albert and Barabasi 2002).

The analysis over universal citation databases indicates that this fact causes un increment on average number of citation while the average number of citation of published decrease over time (Redner 1998; Vazquez 2001).

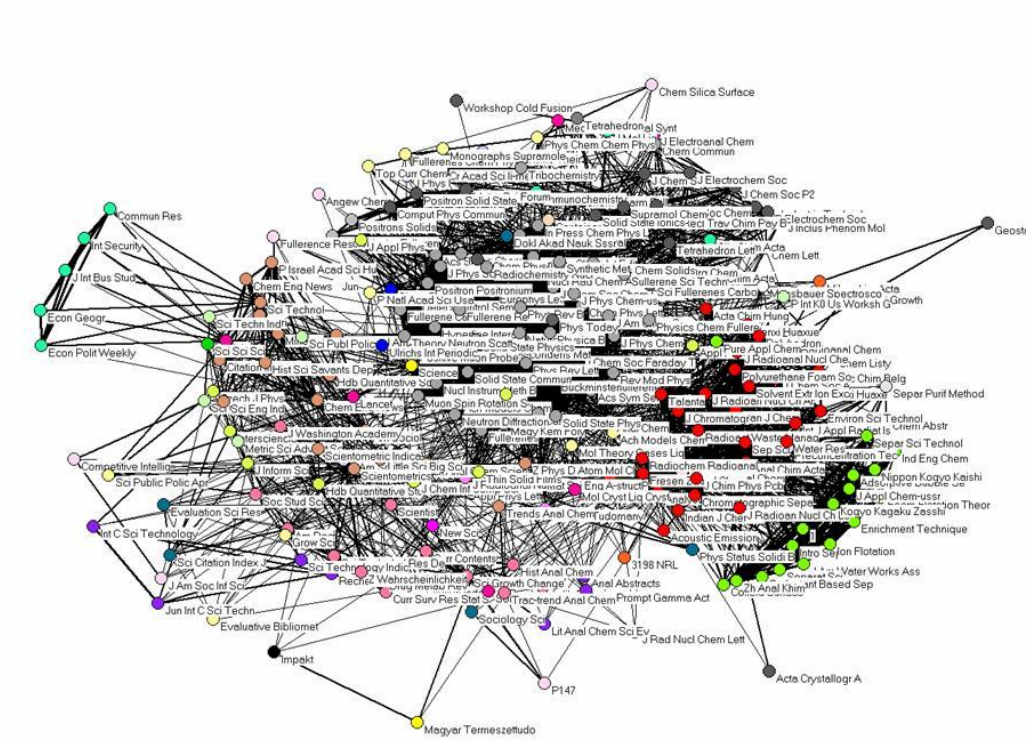

Figure 4-8 Citation network of 236 node and 2221 link in Braun's oeuvre (Leydesdorff 2007)

4.4 **Communication Networks**

The usage of internet and mobile phones causes a huge amount of data consisting on social systems. The following two sections are the most popular communication areas.

4.4.1 **E-mail Networks**

Electronic mail is one of the most popular way of communication and it provides a large data of social relationships (Costa, Osvaldo et al. 2011). There are two type of construction in email networks. In the first approach, the nodes represent email addresses and there is a directed link between two nodes only if an email was sent from the source address to the target address. In the second approach, the nodes are the email addresses as well, but a directed link appears if the target address is contained in the address book of the user of the source address. In some cases, the link directions are ignored, leading to undirected networks.

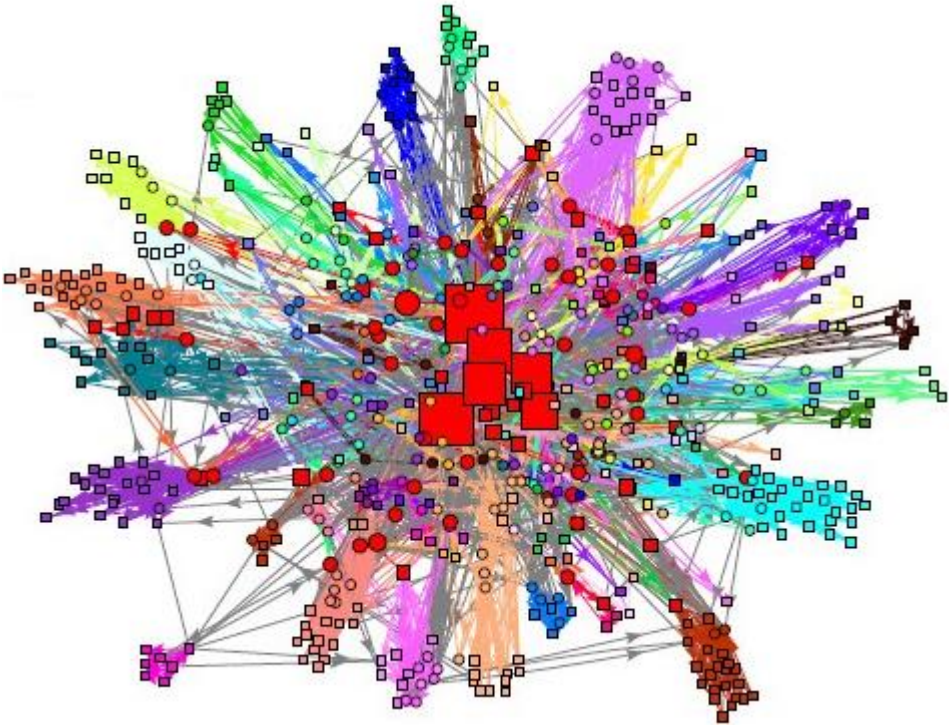

Figure 4-9 Mail network example of University of Kiel

Some studies showed that email networks are composed of communities. The degree distribution and betweenness changes in this kind of networks, the hub nodes are changing dramatically which insist of reinterpretation of the hubs in dynamic networks.

4.4.2 Telephone Networks

Telephone call networks are constructed from telephone calls made during a period of time. The result is a directed network whose nodes are the telephone numbers and directed links correspond to calls from one number to another (Albert and Barabasi 2002). In some cases the duration of the call or the number of calls between two nodes can be used to define link weights..

The researches shows that the incoming and outgoing degree distributions of the telephone networks follow a power law (Albert and Barabasi 2002). Another point is that probability of two nodes to be connected is inversely proportional to the square of their geographical distance (Costa, Osvaldo et al. 2011) that is also observed that there exist a degree correlation in this kind of networks. . The structure of the network contains communities in which the removal of the strong node can cause a local effect within the communities. When the weak connection is removed, the communities can be divided in smaller ones The degree distribution and betweenness change dramatically in this kind of networks, but the mobility behavior of the mobile call graphs are not random but it shows some regularities (Costa, Osvaldo et al. 2011).

4.5 Computer Networks

4.5.1 Internet

Internet is the originally a military networking system founded by ARPANET. To keep the system secure and rebuts the explicit approach of study is a graph to represent internet. The internet is a network of physically linked computers and other communication devices. It has been studied at two hierarchical levels. At the *router* level, the nodes represent routers and the link represent the physical connections between them. As the mapping of internet changes constantly and is not administrated centrally, an interdomain level can also be alternatively considered. In this case the nodes represent autonomous systems (AS), i.e. sub-networks which are administrated separately, and the links are the physical connections between them (Albert and Barabasi 2002; Costa, Osvaldo et al. 2011). There are many approaches to model the internet in order to

understand its growth, assure its security and improve its performance (Albert and Barabasi 2002).

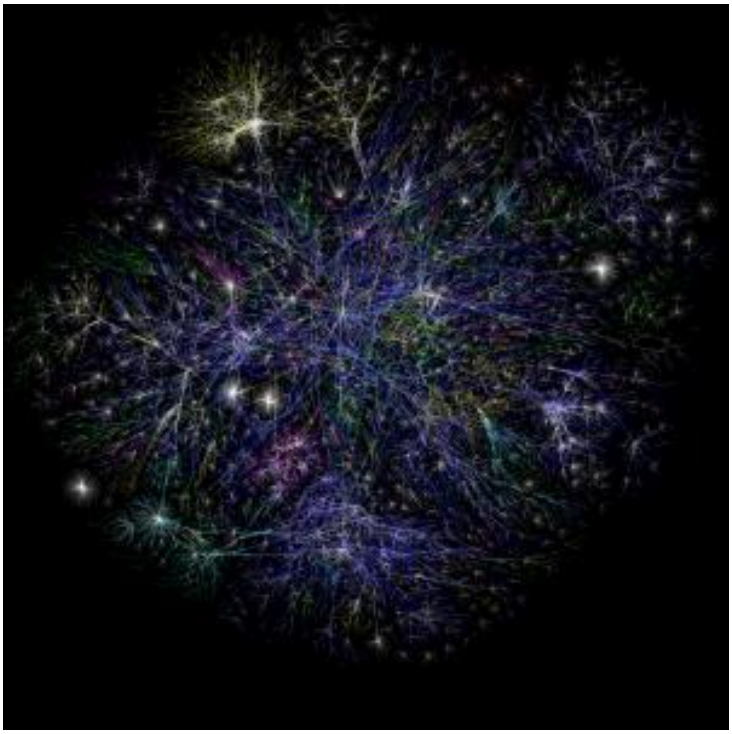

Figure 4-10 Internet as a Complex Network

At both levels, the studies show that the degree distribution follows a power law. Both networks also have the small world property, with high clustering coefficient and small average distance.

4.5.2 **World Wide Web**

World Wide Web (WWW) or just Web is a large software network based on the Internet. WWW is system based on interlinked hypertext documents. It relies on the Hypertext Transfer Protocol for communication. Via a web browser running on the client side, people can access the web pages hosted on Web servers distributed over the internet. Each web page contains text, images, videos etc. And may contain hyperlinks to other pages available on WWW, which give WWW a networked structure (Costa, Osvaldo et al. 2011).

When modeled as a complex network, the nodes of the network correspond to web pages, whereas the directed links are the hyperlinks connecting them.. The WWW is a huge network of over billions of nodes and has an uncontrolled growth (Albert and Barabasi 2002). Since individuals and organizations publish their own interconnected pages, nodes and links are constantly created, modified or destroyed, which makes the system very dynamic.

The map of WWW is usually generated by a computer program called *crawler*. It browses through pages, storing hyperlinks, source and target pages. As some pages are dynamic, require authorization or servers are unreachable, the resulting map cannot contain all the pages. Sometimes the WWW is not navigated at the page level, but rather at the site level. In this case, a node correspond to a site, and a directed link to the set of hyperlinks between the collections of pages forming two sites (Costa, Osvaldo et al. 2011).

4.6 **Transportation Networks**

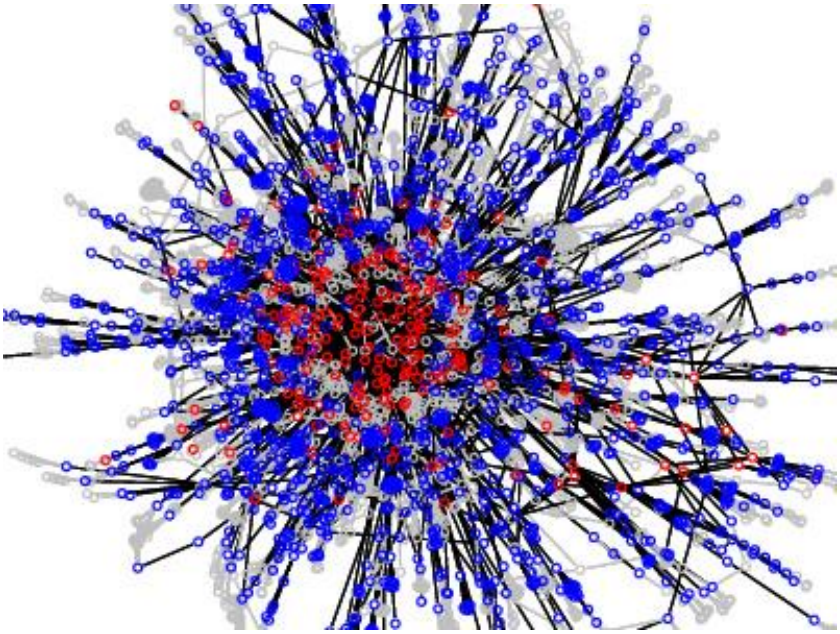

Figure 4-11 Network of web data

The resulting map of crawler consists of a network which represents the WWW, as in Figure 4-11. As the network of the web is directed, there are two degree distributions: incoming and outgoing. The studies shows that both outgoing and incoming degrees have a power law distribution (Albert and Barabasi 2002). Even though some researches prove that a path between randomly chosen pairs of nodes exists in 24 % of times, which supports the absence of a small-world effect, despite the huge size of the network, the WWW displays the small world property in many studies (Costa, Osvaldo et al. 2011). Furthermore, because of the directed nature of the network, to be able to calculate the clustering coefficient the networks are made undirected by making each link bidirectional. Therefore the efficiency of the studies depends on the web data which must be as complete as possible, including page interconnectivity and Meta data of the pages.

4.7 Summary

Table 4-1 represents the distribution of networks over domains. In this section a summary of topological property generalities which are explained at the previous sections are represented as table to facilitate comparison. The free rows are the properties which are not specific for the domain.

Table 4-1 Number of network in each domain

	Number of Networks
Social	25
Citation	20
Communication	28
Ecology	20
Biomolecular	32
Computer	21
Transportation	5

Table 4-2 General property comparison between different network fields

	Degree distribution	Transitivity	Modularity	hierarchical	Hub nodes
Biomolecular	Power law	-	modular	no	yes
Ecology	Power law	-	modular	no	yes
Personal	Power law	Transitive	modular	no	yes
Computer	Power law	-	-	no	yes
Transport	Power law	-	-	no	no
Citation	Power law	-	-	no	no

5 METHODS

5.1 General Method

The clustering algorithms cannot be applied directly to the networks forming the corpus: some preprocessing must be applied first. Figure 5-1 describes the general method we used for this purpose. The first step is to calculate the properties of interest for all the networks in the corpus. The second step is to normalize them, because certain clustering algorithms are sensitive to difference in orders of magnitude in the parameter values. Third, from these normalized properties, it is possible to process distances between each pair of networks. We get a *partial* distance for each property. Fourth, we aggregate these distances to obtain the *total* distance between each pair of networks.

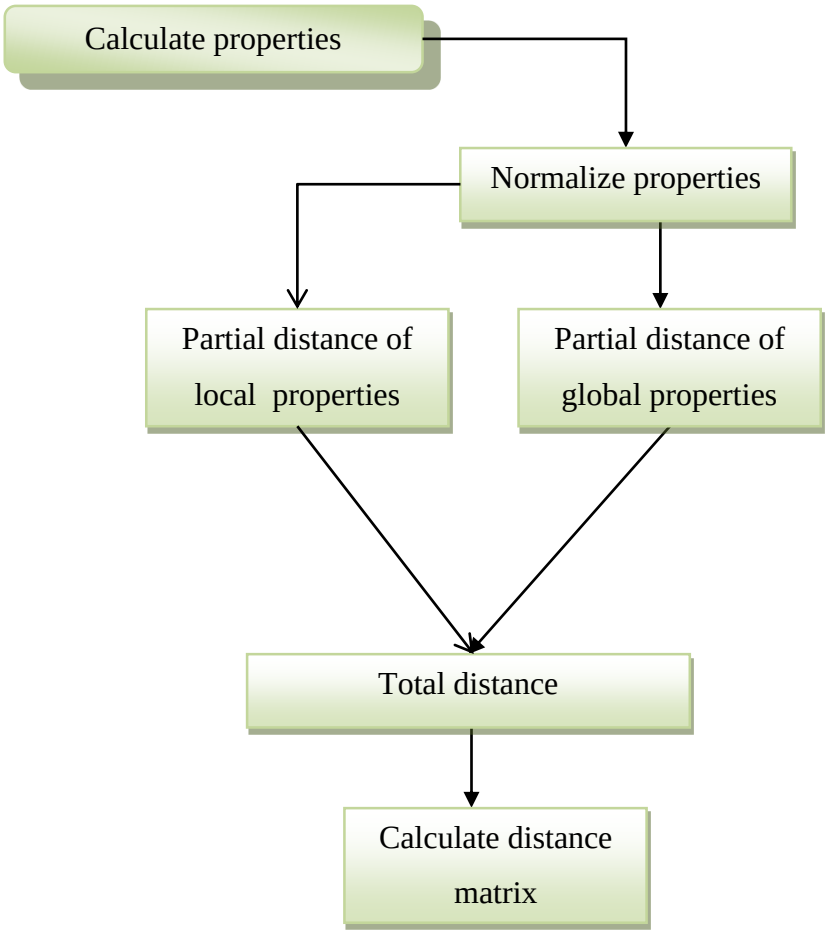

Figure 5-1 Preprocessing of corpus

Fifth, using these total distances, we build the distance matrix of the corpus. Almost all the clustering algorithms can take a dissimilarity or distance matrix as input data. In such a square matrix D, each cell d_{ij} represents the distance between networks number i and j. Figure 5-2 gives an example of distance matrix. Sixth, we apply the clustering algorithms to this distance matrix to get clusters of networks.

```
          [,1]       [,2]       [,3]       [,4]       [,5]
[1,] 0.05237702 0.13776124 0.08411008 0.08223725 0.08247006
[2,] 0.06580911 0.20216412 0.11024494 0.11022834 0.07242202
[3,] 0.22337402 0.22751947 0.15651494 0.15747670 0.19137089
[4,] 0.09595271 0.22120486 0.06989735 0.09004819 0.10649451
[5,] 0.14825731 0.21560811 0.08168898 0.08545655 0.10431461
```

Figure 5-2 Distance matrix view of a 5 network dataset

5.1.1 Representation of the Networks Properties

The first part of our processing is the extraction and normalization of networks properties. As described in section 2.2, those can be characterized through both global and local measures. With the former, we obtain a single value which summarizes the property for the whole network. With the latter, we get a value for each node or link in the network, resulting in a series of values charactering the property. These two kinds of measures need to be handled differently.

For each network in our corpus, we generated two files by calculating the network topological properties. The first one contains the global measures as a single vector; each raw representing a specific measure and it is saved under the name `globals.txt`. The other file contains the local properties as a vector of vectors. Each internal vector also corresponds to a distinct local measure and the whole is saved as `locals.txt`. Each value in an internal vector represents the value measured for an object (node, link, etc.).

5.1.2 Normalization of the Measures

Not all measures are defined on the same range. This can be a problem when performing cluster analysis, because certain algorithms are sensitive to large differences in attribute values, and might give more importance to those with the largest magnitude (de Souto, Araujo et al. 2008). To avoid this, we apply the same *Min-Max* normalization to all measures. Let $X = (x_1, \dots, x_k)$ be a vector of values we want to normalize, where k is here the number of values. Then, the Min-Max normalization consists in processing $X' = (x_1', \dots, x_k')$ such as:

$$x_i' = \frac{x_i - \min\limits_{j \in [1;k]} x_j}{\max\limits_{j \in [1;k]} x_j - \min\limits_{j \in [1;k]} x_j} \tag{5.1}$$

After such normalization, all the values are defined in $[0; 1]$.

However, local and global measures are treated differently. For the global ones, each network is described by a single value: we consequently normalize a series of values over the corpus. This way, networks can be compared consistently, knowing the minimal and maximal values a network can reach in the dataset are 0 and 1, respectively. For instance, for the diameter, suppose we have a maximum of 12729 and a minimum of 1, for all networks. If the diameter of a network is 24, then the normalized value is calculated by dividing $(24-1)$ by $(12729-1)$ which gives 0.001 as a result.

For the local measures, we have a *series* of values for each network, instead of a single one. Each value corresponds to a node, a link or a pair of nodes. these series are normalized separately, so that 0 and 1 now correspond to the minimal and maximal value one can find *for a given network* (and not for the whole dataset, like before). Indeed, the goal here is to compare distributions, so relative differences are more important than absolute ones. After this normalization, the distribution is processed under the form of histograms, containing 20 bins with a step of 0.05.

For instance a network whose degree distribution is $(10,\ 2, 14, 14, 10, 10,\ 6,\ 8, 2,\ 8,\ 4,\ 8)$ becomes $(0.66, 0.10, 1.00, 1.00, 1.00, 1.00, 0.74, 1.00, 0.24, 1.00, 0.49, 1.00)$ after normalization. The corresponding histogram has the following values: $(0, 0.08, 0.00, 0, 0.083, 0, 0, 0, 0, 0.08, 0, 0, 0, 0, 0.08, 0, 0, 0, 0, 0.58)$

5.1.3 **Distance Processing**

Once the measures have been normalized, it is possible to process the distances. First, we individually process a partial distance, for each selected topological measure. Then all of them are combined to get the overall distance between two networks. Again, we treat global and local measures differently.

For each global measure, two networks are compared by simply considering the *Manhattan* distance d_M, i.e. the absolute value of the difference between the measures. If we note x and y the quantities to be compared, we have:

$$d_M(x, y) = |x - y| \tag{5.2}$$

Thus, distances on global measures range from 0 to 1, thanks to the previously performed normalization. For instance, the distance between two networks whose diameters are 0.001 and 0.003 is 0.002.

The result is necessarily a value ranging from 0 to 1 for the local measures, we process the *Earth Mover* (EM) distance (Rubner, Tomasi et al. 1998) between both histograms. Both histograms are normalized and contain the same number k of bins, which ease the computation of the measure. Let us consider two series of bins $X = (x_1, \dots, x_k)$ and $Y = (y_1, \dots, y_k)$. The EM distance is based on a partial distance d'_{EM} processed recursively over each bin, using the following formula:

$$d'_{EM}(x_i, y_i) = \begin{cases} 0 \text{ if } i = 0 \\ |x_i + d'_{EM}(x_{i-1}, y_{i-1}) - y_i| \text{if i} > 0 \end{cases} \tag{5.3}$$

Then, the Earth Mover's distance is defined as the sum of the partial distances over all bins:

$$d_{EM}(X, Y) = \sum_{1 \le i \le k} d'_{EM}(x_i, y_i) \tag{5.4}$$

Because the EM distance is applied to normalized histograms, it also ranges from 0 to 1. The distance between two distributions is calculated by passing two normalized histograms to the function `emd` which is implemented in R under the package `emdist`.

The overall distance is processed for each pair of networks, in order to build the distance matrix required by the clustering algorithms. Let us note n the number of global properties, and m the number of local properties. Each normalized global property is noted M_i $(1 \le i \le n)$ and each normalized local property is noted N_j $(1 \le j \le m)$. Equation (2.6) shows how we combine all property distances to obtain the overall distance $d_O(G, H)$ between two networks G and H.

$$d_O(G,H) = \frac{1}{n}\sum_{1\le i\le n} d_M\big(M_i(G), M_i(H)\big) + \frac{1}{m}\sum_{1\le j\le m} d_{EM}\left(N_j(G), N_j(H)\right) \qquad (5.5)$$

The fact the distances for global and local properties are all taking values in $[0; 1]$ allows us to average them without any scale problem. The resulting average is also ranging from 0 to 1.

5.2 Cluster Analysis Methods

There exist many methods to perform cluster analysis. In this section, we present a selection of 4 representative tools we used to analyze our dataset.

5.2.1 Definition of Cluster Analysis

Cluster analysis consists in empirically forming groups of objects, called clusters, with high intra-cluster similarity and low inter-cluster similarity. One can distinguish various general approaches: partitional, hierarchical and density-based algorithms.

Partitional approaches. They first split the dataset in several mutually exclusive clusters, and then maximize/minimize the intra/inter-cluster similarity by moving objects from one cluster to another (Reynolds, Richards et al. 1992).

Hierarchical approaches. They build a hierarchy of clusters, called *dendrogram*. Two different methods exist for this matter: bottom-up and top-down. In the former, each object is initially considered as a cluster, and those are iteratively merged until only one cluster containing all objects remains. In the latter, on the contrary, all the objects are in the same unique cluster, which is then repeatedly divided until obtaining only singleton clusters. The choice of the final clusters is made by selecting a level, called cut, in the dendrogram, according to some criteria of interest (Kaufman and Rousseeuw 1990).

Density-based approaches. They iteratively build clusters by aggregating close objects around initial objects called seeds.

In this project, we decided to apply several algorithms, each one representative of one of these approaches, in order to be able to check the reliability of our results. In the next subsections, we present briefly the select tools: Agnes (hierarchical bottom-up) (Kaufman and Rousseeuw 1990), Diana (hierarchical top-down) (Kaufman and Rousseeuw 1990), Pam (partitional) (Reynolds, Richards et al. 1992) and DBscan (density-based) (Ester, Kriegel et al. 1996).

5.2.2 **Agnes**

Agnes (Agglomerative Nesting) corresponds to a bottom-up hierarchical approach (Kaufman and Rousseeuw 1990). It is implemented in the R package named `cluster`, under the form of the function `agnes`. Here is its header:

```
agnes(x, diss = inherits(x, "dist"), metric = "euclidean",
stand = FALSE, method = "average", par.method,
keep.diss = n < 100, keep.data = !diss)
```

The parameters are as follows:

- `x` can be a distance matrix or the raw data matrix.
- `diss` is a Boolean flag which must be set to `TRUE` if `x` is a dissimilarity matrix.
- `metric` must be specified when x is a data matrix, in order to indicate how the dissimilarity matrix should be processed.
- `stand` is a Boolean flag which must be set to TRUE if one wants to normalize the raw data before processing the distance matrix, in the case where `x` is not already a distance matrix.
- `method` defines the method used to process inter-cluster distance when selecting the clusters to be merged:
    - `"average"`: (average linkage) average of inter-object distances
    - `"single"`: (single linkage) minimal inter-object distance
    - `"complete"`: (complete linkage) maximal inter-object distance
    - `"ward"`: Ward's method
    - `"weighted"`: (weighted average linkage) and its generalization

    - o "flexible": generalization of the previous one. It uses the thepar.method parameter.
- keep.diss, keep.data: Boolean flags indicating if the distance and input matrices should be kept in the returned values, respectively.

In this project, the Agnes function is used by passing distance matrix as parameter x. As we use distance matrix diss parameter takes the value FALSE and metric is not used. The stand parameter is also used as FALSE as the matrix is already normalized.

5.2.3 **Diana**

Diana (Divisive Analysis) is a top-down hierarchical method (Kaufman and Rousseeuw 1990). It is also implemented in the R package cluster, under the form of the function named Diana. Here is its header:

```
diana(x, diss = inherits(x, "dist"), metric = "euclidean",
stand = FALSE, method = "average", par.method,
keep.diss = n < 100, keep.data = !diss)
```

Its parameters are similar to those of the function in 5.2.2 Agnes.

5.2.4 **Pam**

Pam (Partitioning Around Medoids) (Reynolds, Richards et al. 1992) is a partitional approach which can be considered as a generalization of the *k*-means method to the case where it is not possible to perform average operation on the data. Since the centers of the clusters cannot be processed exactly, they are approximated by using centroids, i.e. the instance which is the closest to the actual center. It is implemented in the R package cluster, under the form of the pam function.

```
pam(x,k, diss = inherits(x, "dist"), metric = "euclidean",
    medoids = NULL, stand = FALSE, cluster.only = FALSE,
    do.swap = TRUE,
    keep.diss = !diss && !cluster.only && n < 100)
```

In terms of the parameters,

- `x:` is a raw or distance matrix.
- `k:` is the number of cluster.
- `diss:` is the logical flag representing the x type. TRUE in case of distance matrix and FALSE otherwise.
- `metric:` is character representing the metric to be used for calculating the distance matrix .
- `medoids:` `NULL` (default) or length-`k` vector of integer indices (in `1:n`) specifying initial medoids instead of using the '*build*' algorithm.
- `Stand:` logical; if true, the measurements in `x` are standardized, false otherwise.
- `cluster.only:` logical; if true, only the clustering will be computed and returned
- `keep.diss:` logicals indicating if the dissimilarities and/or input data `x` should be kept

In this project a normalized distance matrix is used as data. As we use distance matrix `diss` parameter takes the value `FALSE` and metric is not used. The `stand` parameter is also used as `FALSE` as the matrix is already normalized.

5.2.5 **DBscan**

DBscan uses the density-based approach (Ester, Kriegel et al. 1996). The R package `fpc` implements it, under the form of the `DBscan` function. Here is its header:

```
dbscan(data, eps, MinPts = 5, scale = FALSE, method =
c("hybrid", "raw",
    "dist"), seeds = TRUE, showplot = FALSE, countmode =
NULL)
```

- `data:` can be a data matrix or dissimilarity matrix.
- `eps:` is the reachability distance and `MinPts` is minimum number of reachable point.

- `scale`: is used to scale the data.
- `method`: represents the `treatment` method of matrix. The value "`dist`" treats data as distance matrix, "`raw`" treats data as raw data and avoids calculating a distance matrix, "`hybrid`" expects also raw data, but calculates partial distance matrices.
- `seeds`: is a flag which is FALSE to not include the isseed-`vector` in the dbscan-object.
- `showplot`: takes values 0 for not to plot and 1 for plot per treatment and 2 for plot per sub iterations.
- `countmode`: is used to specify the vectors of points at which the progress will be reported. X and object are objects of `dbscan`. `Predict.max` is batch size for predictions.

In this project a distance matrix is used as data. `eps` and `scale` are evaluated to find out the best fitting clustering structure.

5.3 Clusters Comparison and Evaluation

Two problems are the direct consequence of cluster analysis. First, each tool can produce different clusters, depending on the parameter values used. The question is then to know how to select the best clusters. Second, since we have several tools, how can we compare their best clusters, in order to assess their agreement? Several approaches exist to solve both problems. In this section, we focus only on the most widespread ones. We first present the Silhouette measure, used to assess the quality of a partition (cluster set) and then the Adjusted Rand Index, which can be used to compare two the clusters of two distinct partitions, and therefore quantify the agreement between the two methods having produced the clusters.

5.3.1 Silhouette

As mentioned before, the Silhouette measure was designed to assess the quality of a partition, i.e. of a set of clusters. It relies on a distance matrix comparing each pair of instances in the considered dataset. The first step consists in calculating the *average distance* between an instance of interest i and the rest of the instances located in the *same cluster*: this value is noted $a(i)$. We then do the same thing between the instance and the instances located in *another cluster* C, and we note the result $b(i, C)$. We note $b(i)$ the closest cluster, i.e. the one for which the distance to i is the smallest: $b(i) = \min_{C} b(i, C)$.

Intuitively, in a good clustering, $a(i)$ should be low (strong cohesion inside the cluster) and $b(i)$ should be high (strong separation between the clusters). The Silhouette value of an instance $s(i)$ is then defined as (Aranganayagi and Thangavel 2007):

$$s(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)} \tag{5.6}$$

The range of $s(i)$ is $[-1; 1\,]$, and greater value means that i clearly belongs to its current cluster. By averaging $s(i)$ over all clustered instances, we get an overall performance measure also defined on $[-1; 1\,]$.

5.3.2 **Adjusted Rand Index**

The Rand Index (Rand 1971) is a classical measure used to compare partitions, and therefore mutually exclusive clusters like ours. Let us define the following quantities:

- a: number of pairs of instances put in the same cluster in both partitions.
- b: the number of pairs of instances put in different clusters in both partitions.
- c: number of pairs of instances put in the same cluster in the first partition, but in different clusters in the second one.
- d: number of pairs of instances put in different clusters in the first partition, but in the same cluster in the second one.

Then, the Rand Index is defined as:

$$RI = \frac{a+b}{a+b+c+d} \tag{5.7}$$

The *Adjusted rand index* is a corrected for chance version of the classic Rand index (Hubert and Arabie 1985). The general formula for chance correcting some index I is the following:

$$I_C = \frac{I - E(I)}{I_{max} - E(I)} \tag{5.8}$$

Where I is the non-corrected index, I_{max} is its upper bound, $E(I)$ is its expected value on the considered data, and I_C is the corrected index. For the Rand index, let us note n_{ij} the number of instances belonging to cluster i in the first partition, and to cluster j in the second one. We can then note n_i the number of instances belonging to cluster i in the first partition, whatever their cluster in the second partition is, and n_j the symmetric quantity: number of instances in cluster j, independently from their cluster in the first partition. The Adjusted Rand Index is then defined as:

$$ARI = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}}{\left(\frac{1}{2}\right)\left[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}} \tag{5.9}$$

6 RESULTS AND DISCUSSION

The dataset used in this project consists of a collection of 152 networks publicly available on the Internet. As mentioned at the beginning, one of our goals was to study how the type of systems represented by the network can affect its topology. For this reason, the dataset is split into 7 different domains, as described in section 0:

- **Social interactions**: acquaintances, sexual and trust networks;
- **Scientific citations**: bibliographic references;
- **Communication networks**: email and phone networks;
- **Ecological systems**: taxa and their predator-prey relationships;
- **Biomolecular interactions**: protein, metabolic and genetic interaction networks;
- **Computer networks**: various representations of the Internet and the Web;
- **Transportation systems**: airport interconnections and road systems.

In this section, we present the analysis of these data, using the methods described in section 5.1. We first discuss the topological properties of the networks. Then, we compare them in terms of correlation. We also identify the properties allowing to discriminate the domains. Finally, we look for clusters in our dataset, and comment the identified groups.

6.1 Topological Properties

Let us describe our datasets in terms of the topological properties presented in section 2.2.

Size. For all domains, the size of the smallest networks is of the same order of magnitude: a few tens of nodes. However, this is not the case for the largest ones. The largest Ecological and Transportation networks contain a few hundred nodes. For Social, Communication and Biomolecular networks, it is several thousand nodes. And Citation and Computer Science networks reach several tens of thousands of nodes. This highlights the fact real-world network sizes are very heterogeneous, spanning 3 orders of magnitude. This is confirmed by the generally very large standard deviations.

Density. Similarly to what can be observed in the literature, most of our networks are very sparse, as seen in the average density and standard deviation of all domains. For some of them, the density is even as low as 10^{-4}. However, the average density of Social and Transportation networks is clearly higher (roughly the double of the others). Moreover, some networks are remarkably dense in the Social, Communication and Biomolecular domains, as highlighted by their upper bounds.

Table 6-1 Overview of the main topological properties of networks in terms of domains

	Size	**Density**	**Diameter**	**Radius**
Social	[11,1882]	[0.0004, 0.38]	[10, 10406]	[1,16]
	Mean:143.88	Mean: 0,32	Mean: 619.17	Mean: 7
	S: 448.52	SD: 0,26	SD: 25.01	SD:6.84
Citation	[35,27779]	[0.0004,0.26]	[3,37]	[0,49]
	Mean:3424.53	Mean: 0.07	Mean: 13.93	Mean: 8.29
	SD: 7547.97	SD: 0.09	SD: 0.26	SD: 13.67
Communication	[11,1882]	[0.0004, 0.26]	[3, 24]	[1,22]
	Mean:143.88	Mean: 0,07	Mean: 11,58	Mean: 15.30
	S: 448.52	SD: 0,09	SD: 7.43	SD: 15.50
Ecological	[24,128]	[0.08, 0.23]	[8, 12729]	[2, 11]
	Mean: 65.38	Mean: 0.15	Mean: 1417	Mean: 3
	SD: 35.00	SD: 0.03	SD: 369.3	SD: 2.16
Biomolecular	[23,3839]	[0.001, 0.34]	[3, 30]	[1, 63]
	Mean 1099.44	Mean: 0.03	Mean: 12.74	Mean: 10.21
	SD:889.27	SD: 0.08	SD: 4.98	SD: 25.29
Computer Science	[18,10680]	[0.0002, 0.05]	[4, 46]	[1,352]
	Mean: 158.28	Mean: 0.04	Mean: 14.5	Mean: 38.13
	SD:2973.78	SD: 0.11	SD: 10.95	SD: 86.11
Transportation	[75,332]	[0.03, 0.24]	[1, 19]	[0, 16]
	Mean:174.40	Mean: 0.22	Mean: 6.94	Mean: 4.28
	SD: 107.60	SD: 0.26	SD: 6.27	SD: 5.67

	Transit.	**Modularity**	**Avg. distance**	**Avg. degree**
Social	[0, 0004,0.86]	[0, 0.89]	[1.16, 11.65]	[1.84,33.34]
	Mean: 0.40	Mean: 0.26	Mean: 2.71	Mean: 8.22
	SD:0.25	SD: 0.28	SD: 2.40	SD: 7.88
Citation	[0.03, 0.69]	[0.14, 0.93]	[1.76,8.46]	[3.24,516.80]
	Mean: 0.23	Mean: 0.41	Mean: 3.88	Mean: 39.81
	SD: 0.17	SD: 0.20	SD: 1.55	SD: 104.77
Communication	[0.03, 0.59]	[0.14, 0.76]	[1.75, 5.34]	[3.42,156.80]
	Mean: 0.27	Mean: 0.42	Mean: 3.19	Mean: 63
	SD:0.16	SD: 0.19	SD: 1.13	SD: 16.18
Ecological	[0.28, 0.49]	[-0.003, 0.51]	[1.6, 3.36]	[5.12, 33.90]
	Mean: 0.38	Mean: 0.06	Mean: 1.81	Mean: 18.15
	SD: 0.07	SD: 0.15	SD: 0.39	SD: 10.11
Biomolecular	[0.02,0.54]	[0.04, 0.80]	[1.80, 7.65]	[2.41, 14.48]
	Mean: 0.07	Mean: 0.52	Mean: 4.66	Mean: 5.37
	SD: 0.14	SD: 0.14	SD: 1.16	SD: 2.15
Computer Science	[0,001, 0.50]	[0, 0.88]	[1.49, 18.98]	[2.54, 39.1]
	Mean: 0.13	Mean: 0.43	Mean: 4.31	Mean: 6.95
	SD: 0.14	SD: 0.26	SD: 3.48	SD: 8.67
Transportation	[0,003, 0.84]	[0, 0.44]	[1.21, 3.48]	[4, 194.64]
	Mean: 0.32	Mean: 0.15	Mean: 2.37	Mean: 37.90
	SD: 0.26	SD: 0.16	SD: 0.70	SD: 69.61

Degree. According to the Kormogorov-Simirnov test, all the studied networks have a power law distributed degree, a prominent feature in complex networks literature. For most domains, degree bounds have the same order of magnitude: a few units for the lower bound, several tens for the upper bound. The exceptions are Transportation, Communication and Citation networks, whose upper bounds reach several hundreds. For the Citation domain, this can be explained by the fact the networks are larger (in terms of nodes), compared to other domains, while they are as dense. For the Transportation and Communication domains, the networks are small but very dense, which can explain these high upper bounds.

Transitivity. The literature highlights the fact real-world networks generally have a high transitivity. It does not seem to be the case so much when looking at the average values obtained on our dataset, which range from 0.07 to 0.40. A look at the bounds shows us the smallest values are almost zero, and the highest ones are not so large (around $0.5 - 0.6$), with the exception of Social and Transportation networks (0.86 and 0.84, respectively). The relatively large standard deviations highlight the heterogeneity of the networks in terms of transitivity. When However, when comparing with values expected for ER networks with the same size and density, it turns out the networks of our dataset are more transitive, while following very closely the evolution observed in ER networks, as seen in Figure 6-1.

Distance. The order of magnitude of the average distance and both distance bounds are roughly the same for all domains: the lower bounds are close to 1, the upper bounds are close to 10, and the average distances lie in between. All networks consequently have a very small average distance, when compared to their size in terms of nodes. Larger networks have a longer distance, but the increase is marginal. The observed average distances are higher than those expected for ER random networks of same size and density, as illustrated by Figure 6-1. This means the observed values alone are not sufficient to decide if the networks are small-world.

Eccentricity. Distance distribution of networks mostly follows a similar curve. It is observed that there is concentration in the low and high eccentricity which follows a

sharp decraease from the low values to the average and a smooth increase towards high values. Except Ecological networks which have a more smooth eccentricity distribution. The order of magnitude of the diameter is the same for most domains, independently from the network size: it ranges from a few hops to a few tens. However, this is not true for the Social and Ecological networks, since the upper bound is tens of thousands of hops for them. This means that, even if the average distance is of the same order of magnitude than in other domains, it is possible for nodes to be much far from the network center in Social and Ecological networks. Interestingly, the same observation does not hold for the radius, which is roughly similar for most domains. Computer networks stand out, with a radius of hundreds of hops, instead of tens for the other domains.

Centrality. It is observed that betweenness centrality follows a normal distribution in all domains whereas closeness centrality has an opposite behavior which supports the number of nearest and farthest nodes are not neglectable. In terms of edgebetweenness we have two picks in the curve of distribution one between the minimum and the mean, the second one is near the upper-bound. This indicates there are mostly and equally used links in networks.

Modularity. The modularity of the dataset changes between -0.02 and 0.92 and it is seen that a large number of networks have positive modularity value. The most modular networks belong to Citation networks. Except Transportation and Ecological networks modularity of dataset seems very high. Modularity in ER networks are accepted as0. When compared to ER networks it is seen that all domains have significantly modular networks.

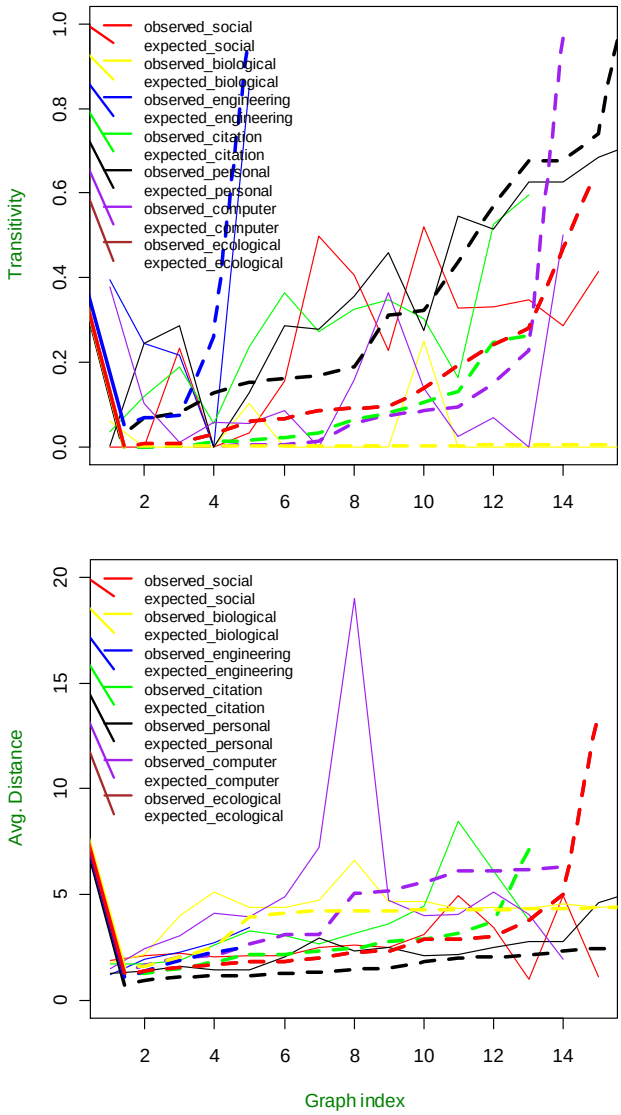

Figure 6-1 Comparison between transitivity and average degree of networks and same size of Erdős–Rényi networks in terms of domains.

6.2 **Correlation Study**

We now examine the correlations between the topological properties studied in the previous subsection. As mentioned before, we distinguish two types of properties: *global* and *local* ones. To ease the interpretation of our results, we split the correlation study in three parts: global vs. global, local vs. local and global vs. local.

Global vs. global. Table 6-2 shows the correlation between global properties only. Most of the values are close to zero, indicating no linear relationships between the properties. However, a few strong positive and negative correlations are also observed. The highest (0.76) one is measured between the density and transitivity which can be explained by the fact that when a network becomes denser the possibility to find triangles increases, too. The average distance and radius are also highly correlated (0.59). This is certainly due to the fact both measures are based on the notion of distance, and reflect how compact the network is. . Density and transitivity are both negatively correlated to average distance (-0.45 and -0.43, respectively). When the network becomes denser, the average distance automatically decreases: because of the additional links, the shortest paths become even shorter. When the average distance is large, the probability for direct connections decreases, impacting the number of triangles. Modularity is positively correlated with average distance (0.60), and like this measure, it is negatively correlated with both density and transitivity (-0.71 and -0.51 respectively). Indeed, the presence of a community structure requires links to be concentrated in communities. Do, the network must be relatively sparse: if it is too dense, then the community structure cannot exist. The presence of a community structure increases the average distance: the sparsity of direct connections between nodes from different communities makes shortest paths longer, in average.

Local vs. local. Local properties take the form of distributions, so it is not possible to compare them directly using Pearson's coefficient. Instead, we considered two series constituted of the distances between these distributions, for each pair of networks in our dataset. So, we insist on the fact we do not consider the direct correlation between two measures here, but rather the correlation of the distances based on these measures. The idea is to quantify how much distributions are correlated. The results are presented in Table 6-3. Some measures are not correlated with any other: it is the case for edge-betweenness. On the contrary, we observe a relatively strong correlation between the remaining measures. This is particularly true of degree and local transitivity (1.00), which indicates their distributions change similarly from one network to another. This does not necessarily mean degree and transitivity are directly linearly dependent, but rather that when two networks have a similar degree distribution, they also have a similar

transitivity distribution, and vice-versa. Betweenness centrality is also correlated with both transitivity and degree distribution (0.59 for both), which indicates that networks with similar degree and transitivity distribution show similar betweenness centrality distribution. Eccentricity and betweenness centrality are also very strongly correlated (0.79), meaning that when the maximal distance of the nodes is distributed similarly between two networks, then the numbers of shortest paths going through nodes are also distributed similarly. The fact all measures except edge-betweenness are relatively correlated indicates there is a certain redundancy in the information conveyed by these properties.

Table 6-2 Correlation between global properties

	Density	Diameter	Transitivity	Modularity	Average Distance	Average Degree	Radius
Density	-	0.02	0.76	-0.71	-0.45	0.16	-0.14
Diameter	-	-	0.04	-0.12	-0.09	-0.01	-0.03
Transitivity	-	-	-	-0.51	-0.43	0.12	-0.09
Modularity	-	-	-	-	0.60	-0.13	0.16
Average Distance	-	-	-	-	-	-0.16	0.59
Average Degree	-	-	-	-	-	-	0.00
Radius	-	-	-	-	-	-	-

Global vs. local. To study the correlation between global and local properties, we also used the distances. Here again, it is important to be cautious with our interpretation: a strong correlation means that when two networks are similar in terms of some global measure, they are also similar regarding the distribution of the considered local measure. Table 6-4 shows the obtained results: most of the measures are not correlated. However, we observe a relatively strong positive correlation for some of them. The highest is observed for density and eccentricity (1.00). This means that, when two networks have the same density, they tend to have the same eccentricity distribution. At a lesser extent, the same remark can be made for the closeness and betweenness centrality.

Table 6-3 Correlation between local properties

	Degree Distribution	Betweenness centrality	Closeness centrality	Local Transitivity	Eccentricity	Edge-betweenness
Degree Distribution.	-	0.55	0.24	1.00	0.34	0.10
Betweenness Centrality	-	-	0.45	0.43	0.79	0.01
Closeness Centrality	-	-	-	0.33	0.40	-0.01
Local Transitivity	-	-	-	-	0.23	0.01
Eccentricity	-	-	-	-	-	-0.01
Edge-betweenness	-	-	-	-	-	-

Table 6-4 Correlation between global and local properties

	Density	Diameter	Transitivity	Modularity	Avg. Dist	Avg. Degree	Radius
Degree Distribution	0.10	-0.09	0.25	0.13	0.01	0.00	0.00
Betweenness Centrality	0.43	-0.09	0.23	0.00	0.01	0.00	0.00
Closeness Centrality	0.44	0.01	0.18	0.04	-0.04	0.00	0.00
Local Transitivity	0.31	-0.06	0.33	0.02	0.01	0.00	0.00
Edge-betweenness	0.24	-0.25	0.04	-0.01	-0.01	0.00	0.00
Eccentricity	1.00	-0.12	0.43	-0.01	0.07	0.00	0.00

6.3 **Domains Comparison**

We applied an ANOVA followed by Tukey's post-hoc test, in order to identify which properties allow discriminating domains. The ANOVA reveals 4 properties are significantly different in at least one domain: average distance ($p = 3 \times 10^{-3}$), density ($p = 6 \times 10^{-6}$), modularity ($p = 4 \times 10^{-3}$) and transitivity ($p = 7 \times 10^{-6}$). We performed Tukey's post-hoc test to identify which domains have different average values for these properties. Our results are displayed in Table 6-5.

Significant difference in terms of density concerns only social networks, which differ from Biomolecular, Citation and Computer networks. This means there is no relevant difference between the other domains in terms of how dense the networks are. So if we

were to partition domains depending on density, density alone would not be sufficient: Social networks are not significantly different from Ecological, Transportation and Communication networks, but these are themselves not significantly different from Biomolecular, citation and computer networks.

The other discriminant properties are more widespread than density. In terms of transitivity, Biomolecular networks are different from all other domains except computer and communication networks. Computer networks themselves are different from ecological and social networks. Again, it is not possible to partition the domains here by putting Biomolecular and computer networks aside: communication networks are neither significantly different from Biomolecular or Computer networks, nor from the rest of the domains.

In terms of modularity, Ecological networks are significantly different from all domains except Citation and Transportation networks. However, like for the other properties, there is no clear separation based on this property only. Similarly, Biomolecular and Computer networks significantly differ from half the domains in terms of average degree, but those domains only partially overlap, therefore preventing any clear separation.

In conclusion, it seems possible intuitively to distinguish different groups of domains, when they differ by several properties at the same time. This is noticeably the case for Biomolecular and Computer networks on one side, and Social networks on the other side. However, no objective ANOVA results really back this observation. This is the reason why we also conducted a cluster analysis to complete our view of the dataset.

Table 6-5 Tukey test result, significant properties for network domain pairs

	Biomolecular	Citation	Computer	Ecology	Transportation	Social	Communication
Biomolecular		Transitivity		Transitivity Modularity Avg. dist. Density	Transitivity Modularity Avg. dist.	Density Transitivity Modularity Avg.dist	Transitivity Avg.dist
Citation			Avg. dist.	Density	Modularity	Density	Avg. dist.
Computer				Transitivity Modularity Avg. dis Densityt	Modularity	Density Transitivity Avg.dist Modularity	Avg.dist
Ecology						Modularity	
Transportation							
Social							Density
Communication							

6.4 **Network Clusters**

As mentioned in section 0, we have applied 4 clustering algorithms (Agnes, Diana, DBscan and Pam) over the whole dataset; using the Silhouette measure to identify the best partitions, and the Adjusted Rand Index (ARI) to compare them. All methods reach their maximal Silhouette value for 2 clusters, which is a strong agreement. Diana has the highest Silhouette with 0.44, the second being Pam with 0.42, followed by DBscan with 0.40 and Agnes with 0.39. These values are not very high (the Silhouette upper bound being 1), but they still show there is a non-random separation between two groups of networks. Table 3-1 shows the ARI values obtained when comparing the clusters estimated by the different clustering tools. The clusters found by Diana and Agnes have largely similar structures, with an ARI of 0.75. After them, Pam and Agnes shows the second highest similarity with a 0.45 ARI, and Diana and Pam reach the value 0.41. On the contrary, the clusters found by DBscan are very different, since the ARI is almost zero when compared with all three other methods. Because of the nature of this

algorithm, it certainly means it found non-convex clusters. In the rest of this work, we focus on the clusters identified by Pam, because it is highly similar with hierarchical both algorithms, and is very close to Diana in terms of Silhouette. Therefore, we are aiming at making a trade-off between the cluster quality and agreement between algorithms.

Table 6-6 ARI results for 4 algorithms

	Agnes	Diana	Dbscan	Pam
Agnes		0.750	0.003	0.450
Diana	-		0.001	0.410
DBscan	-	-		0.010
Pam	-	-	-	

Table 6-7 represents the distribution of networks of different domains over the two clusters detected by Pam. While Biomolecular, Citation and Computer networks are largely grouped in the first cluster, Ecological, Transport, Social and Communication networks are mostly grouped in the second cluster. The first cluster is dominated by Biological networks, whereas Social and Communication clusters dominate the second one.

Table 6-7 Participation of networks from different domains in clusters

	Cluster 1	Cluster 2
Social	4	21
Citation	17	3
Communication	7	25
Biomolecular	28	4
Ecology	0	20
Computer	16	5
Transportation	0	5

We have applied an ANOVA over the two clusters to find out the properties which have influence on the cluster division. The ANOVA indicates 9 properties undergoing significant changes over the clusters: transitivity ($p = 2 \times 10^{-16}$), diameter ($p = 0.01$), modularity ($p = 2 \times 10^{-16}$), average distance ($p = 1 \times 10^{-8}$), density ($p = 3 \times 10^{-9}$), and average degree ($p = 1 \times 10^{-3}$), closeness centrality ($p = 4 \times 10^{-3}$), local transitivity ($p = 3 \times 10^{-9}$) and edge betweenness ($p = 3 \times 10^{-9}$). This result shows that except radius, betweenness centrality and eccentricity all other global properties which we have used were significant while distinguishing clusters in our dataset. Here we see that the transitivity, density, modularity and average distance are highly significant compared to diameter and average degree. When we consider the results collected under the ANOVA of domains which are explained in 6.3, it is obvious that same properties were significant in distinguishing the domains; hence these four properties can be assumed as the leading properties for differentiation of the clusters.

7 CONCLUSION

The goal of this work was to study the topological properties of complex networks using a systematic approach. For this purpose, we first constituted a dataset of 152 networks representing real-world systems. We distinguished 7 different domains: Biomolecular, Social, Ecology, Citation, Computer, Transport and Communication. We then processed a selection of 14 topological measures for each of these networks, including both local and global measures. We performed various analyses on these values. First, we analyzed the topological properties individually. We observed that Social, Communication, Ecology and Transportation networks shows the similar properties which differentiate them from Citation, Biomolecular and Computer domains. Second, we made a correlation study, and identify strong correlations between certain properties, which seem to be related to the network domains. Third, we study how the domains compare in terms of global properties. An ANOVA followed by Tukey's test revealed certain domains like Biomolecular-Ecology, Biomolecular-Social, Biomolecular-Transportation, Citation-Ecology, Computer-Ecology, Computer-Social have significantly different density, transitivity, modularity and average degree. However, these differences are not consistent enough to allow classifying the domains on the basis of individual topological measures. Therefore, to complete this study, we applied several cluster analysis tools (Agnes, Diana, Pam, DBscan). All agree on the presence of two clusters, however the agreement is not as strong regarding the nature of these clusters: Adjusted Randi Index (ARI) values range from from $0.1\ to\ 0.75$. We selected the most separated clusters, according to the Silhouette measure (0.44, and studied how domains were distributed over them. The separation is very clear, with 3 domains (Social, Communication and Ecology) belonging to one cluster and the 4 other domains (Biomolecular, Transport, Citation, Computer) to the other cluster. Additional ANOVA and Tukey's test revealed and it is seen that the two clusters are significantly different from each other in terms of

transitivity, modularity, average distance, average degree, diameter and density in which there exist also the same properties observed also in domain ANOVA.

The main contribution of our work was to take advantage of data mining approaches to perform the first systematic study of complex network topological properties. By comparison, previous works only focused on a limited number of networks, and on 1 or 2 properties. The second contribution was the constitution of a large network dataset.

However, our work also suffers from certain limitations. First, we could not include in our dataset all the freely available networks, because their normalization is a long process. Moreover, a manual verification must be performed to ensure the same network is not present twice in the dataset. But this limitation can be easily overcome with more time. Second, we expected *a priori* to get a larger number of clusters, and a clearer separation between them. We think one reason for that can be we did not use the information allowing to distinguish more clusters. This limitation can be overcome by considering more topological properties.

REFERENCES

[Albert, R. and A.-L. Barabasi (2002). "Statistical mechanics of complex networks." **74**.

Aranganayagi, S. and K. Thangavel (2007). Clustering categorical data using silhoutte coefficient as a relocation measure. International Conference on Computational Intelligence and Multimedia Applications**:** 13-17.

Brandes, U., M. Eiglsperger, et al. "GraphML Primer." from http://graphml.graphdrawing.org/primer/graphml-primer.html.

Chen, Q., H. Chang, et al. (2002). "The Origin of Power Laws in Internet Topologies Revisited." 10.

Costa, L. d. F., N. O. J. Osvaldo, et al. (2011). "Analyzing and modeling real-world phenomena with complex networks: a survey of applications." Advances in Physics **60**: 84.

de Souto, M. C. P., D. A. S. Araujo, et al. (2008). Comparative Study on Normalization Procedures for Cluster Analysis of Gene Expression Datasets. IEEE International Joint Conference on Neural Networks**:** 2792-2798.

Dehmer, M. (2011). Structural Analysis of Complex Networks, BirkhaÃser.

Ester, M., H.-P. Kriegel, et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. 2nd International Conference on Knowledge Discovery and Data Mining**:** 226-231.

Estrada, E., M. Fox, et al. (2010). Complex Networks: An Invitation. Network Science - Complexity in Nature and Technology, Springer**:** 1-11.

Fortelny, N. (2010). Metabolic network analysis:Towards the construction of a meaningful network.

Freeman, L. C. (1978). "Centrality in Social Networks I: Conceptual Clarification." Soc Networks **1**(3): 215-239.

Ghoshal, G. (2009). Structural and dynamical properties of complex networks, The University of Michigan.
Girvan, M. and M. E. J. Newman (2001). "Community structure in social and biological networks." PNAS **99**(12): 7821-7826.

Gonzalez, M. C., P. G. Lind, et al. (2006). "System of Mobile Agents to Model Social Networks." Physical Review Letters.

Gursoy, A., O. Keskin, et al. (2008). "Topological properties of protein interaction networks from a structural perspective." Biochem Soc Trans **36**(Pt 6): 1398-1403.

Heymann, S. (2009). "GEXF File Format." from http://gexf.net.

Hubert, L. and P. Arabie (1985). "Comparing partitions." Journal of Classification **2**(1): 193-218.

Kaufman, L. and P. J. Rousseeuw (1990). Finding Groups in Data: An Introduction to Cluster Analysis. New York, US-NY, Wiley.

Lancichinetti, A. and S. Fortunato (2009). "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities." **9**.

Leydesdorff, L. (2007). "The position of Tibor Braun's Œuvre:Bibliographic journal coupling."

Newman, M. E. J. (2003). "The structure and function of complex networks." SIAM Review **45**.

Newman, M. E. J. (2006). "Modularity and community structure in networks."

Odum, E. P. and G. W. Barrett (1953). Fundemantals of Ecology.

Orman, G. K. (2010). Community Detection in Complex Networks MSc, Galatasaray University.

Rand, M. (1971). "Objective criteria for the evaluation of clustering methods." Americal Statical Association **66**.

Redner, S. (1998). "How popular is your paper? an empirical study of the citation distribution." The European Physical Journal.

Reynolds, A., G. Richards, et al. (1992). "Clustering rules: A comparison of partitioning and hierarchical clustering algorithms." Journal of Mathematical Modelling and Algorithms **5**: 475-504.

Rubner, Y., Y. C. Tomasi, et al. (1998). The Earth Mover's Distance as a Metric for Image Retrieval. ICCV**:** 59-66.
Schlitt, T. and A. Brazma (2007). "Current approaches to gene regulatory network modelling." **8**: 22.

Takemoto, K. (2012). "Current Understanding of the Formation and Adaptation of Metabolic Systems Based on Network Theory." metabolites: 29.

Vazquez, A. (2001). "Statistics of citation networks."

Watts, D. J. and S. H. Strogatz (1998). "Collective dynamics of 'small-world' networks." Nature **393**(6684): 409-410.

Zhang, F., C. Hui, et al. (2011). "An interaction switch predicts the nested architecture of mutualistic networks." Ecology Letters.

Zhao, J., H. Yu, et al. (2006). "Complex networks theory for analyzing metabolic networks." Chinese Science Bulletin **51**(13): 1529-1537.

BIOGRAPHICAL SKETCH

Burcu Kantarcı has graduated from Beşiktaş Atatürk Anatolian High School, and obtained a BSc Degree in Computer Engineering at the Galatasaray University in 2011. She was awarded the Erasmus scholarship which allowed her studying at the Joseph Fourier University - Ecole Polytechnique in Grenoble, France in 2009-2010. While pursuing the Master's degree in Galatasaray University, an article on her master thesis is published with the same name in 2013. Besides she has one year experience of finance in Garanti Technologie as software engineer in 2011-2012. She is currently working in Tradesoft as software engineer.