POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in
Ingegneria Matematica

POLITECNICO
MILANO 1863

Comparing Networks: Methods and Applications

Relatore: Prof. Carlo PICCARDI

Correlatore: Prof. Lucia TAJOLI

Tesi di Laurea di:

Mattia TANTARDINI Matr. 858603

Anno Accademico 2017 - 2018

A te Benedetta,
che ogni giorno mi sei accanto
e mi accogli per ciò che sono.

Ringraziamenti

Desidero ringraziare i miei relatori Carlo Piccardi e Lucia Tajoli per aver accettato di lavorare con me, per la loro grande disponibilità e per il loro aiuto nella stesura della tesi. Ringrazio anche la professoressa Francesca Ieva per l'iniziale interessamento e disponibilità a lavorare con noi.

Ringrazio il dottor James Bagrow per la sua disponibilità e il suo aiuto nel rivedere e correggere il codice relativo alla distanza Portrait Divergence, che ho utilizzato nel mio lavoro di tesi. Ringrazio anche il dottor Daniele Durante per aver condiviso il codice riguardante il metodo bayesiano per modellizzare una popolazione di networks da lui sviluppato.

Ringrazio la mia famiglia, per avermi sostenuto nel continuare gli studi e per avermi permesso di raggiungere questo traguardo.

Voglio ringraziare i miei compagni di avventura in questo lungo viaggio: Pietro, Luca, Dani, Cami, Teo, Chiara, Ste, Richi, Fra, Teme, Elisa, Romario. Grazie per l'aiuto, il sostegno, lo studio insieme, le discussioni animate, il divertimento, le nostre innumerevoli partite a carte, le camminate insieme in montagna e per avermi insegnato a non prendermi troppo sul serio. Grazie a Ilaria e a Monica, fidate compagne di progetto. Grazie anche ai compagni di viaggi più concreti, quelli in treno per raggiungere l'università e tornare a casa, perché insieme a loro la giornata cominciava e finiva al meglio.

Mattia Tantardini

Contents

List of Figures

List of Tables

Abstract

The flexibility of network modelling and the subsequent growth of available networked data open the problem of devising effective methods for the comparison of networks. Plenty of methods have been designed to accomplish this task. Most of them only deal with undirected and unweighted networks, but it is crucial to find methods able to handle also directed and weighted networks, to properly exploit all the available information. In this work, we give three main contributions. Firstly, we review a collection of such methods, highlighting the criteria they are based on and their advantages and drawbacks, and we suggest a novel classification. Secondly, we test the methods on synthetic networks and we asses their usability and the meaningfulness of the results they provide. Finally, we apply the methods to three real-world datasets: the European Air Transportation networks, the FAO Trade Networks and the World Trade Networks. The comparison of economical networks reveals important characteristics and patterns in the international trade flows.

Keywords: network comparison, network distance measures, clustering, multilayer networks.

Sommario

La flessibilità della modellistica tramite reti e la conseguente ampia disponibilità di dati pone il problema di sviluppare metodi efficaci per il confronto tra reti. Numerosi metodi sono stati proposti a questo scopo. La maggior parte di essi può essere utilizzata per confrontare solo reti non dirette e non pesate, ma è cruciale poter sviluppare metodi capaci di confrontare anche reti dirette e pesate, per sfruttare appieno tutte le informazioni disponibili. In questo lavoro, si forniscono tre contributi principali. Inizialmente, si esaminano una parte di tali metodi, mettendo in luce i criteri su cui si basano e i loro vantaggi e svantaggi, e si propone una nuova classificazione. In secondo luogo, si testano questi metodi su reti sintetiche e si valuta la loro utilizzabilità e la significatività dei risultati che forniscono. Infine, i metodi vengono utilizzati per l'analisi di dataset reali riguardanti la rete del trasporto aereo europeo e le reti di commercio mondiale di vari tipi di prodotti. Il confronto tra reti economiche rivela importanti caratteristiche e strutture nei flussi di commercio internazionale.

Parole chiave: confronto tra networks, distanze tra networks, clustering, multilayer networks.

Introduction

The research on complex networks has exploded in recent years due to the great power of network models to describe and give insights in many real-world applications coming from very different scientific and application areas, including economics, computer science, social sciences, biology, telecommunications, transportation and many others.

The typical study on complex networks is aimed at finding the best graph model fitting the real data, thus giving information about the main characteristics of the considered networks. However, the growth of available data, along with methods to handle and analyse them, leads the research to the field of network comparison, which is aimed at finding and quantifying differences between networks. Among the many possible applications of network comparison, we mention spotting of anomalous situations in a temporal series of graphs (for instance a big event that changes the connections of a Twitter network), or clustering networks to extract informations about possible groups existing in the data (for instance given a set of brain networks, find whether the networks of the ill patients are significantly different from the networks of the healthy ones, and in what they differ).

To perform network comparison, a distance measure between graphs has to be defined. This is an hard task, that combines computational efficiency, interpretability, effectiveness of the results and the specific domain of application of the measure. The literature about this topic is huge and plenty of different methods have been designed, each one based on a different criterion. Therefore, an analysis about the performances of each method has to be carried out in a systematic framework. The aim is to find which are the best methods and in which situations, and to understand the new insights that they can provide when used to compare a dataset of real-world networks.

This work is organised as follows. In Chapter 1, we present the problem of network comparison and some of the most interesting and promising methods we found in literature, from the oldest and naïve approaches to the most recent ones. We then give an overview of the state-of-the-art tools and of the strategies for network comparison and their evolution. We describe the approaches that the different methods adopt and their capabilities, highlighting advantages and drawbacks. In Chapter 2, we firstly present the motivations for the choice of the network distance measures we used for our analyses. Then, we describe in details the tests we carried out on synthetic networks and we discuss the obtained results. In Chapter 3, we present some relevant applications to real-world networks. We give a description of the three datasets we considered, which relate to air transportation

in Europe and to international trade of various goods, and of the analysis we carried out, illustrating the most relevant results. In Chapter 4, we present the main conclusions of this work and we suggest possible future developments and improvements.

Chapter 1

Review of network distance measures

The graph comparison problem arises from the graph isomorphism problem. Two networks $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$ are *isomorphic* if there exists a bijection Φ from $\mathcal{V}_1$ to $\mathcal{V}_2$ such that the edge $uv \in \mathcal{E}_1$ if and only if the edge $\Phi(u)\Phi(v) \in \mathcal{E}_2$ [16]. The graph isomorphism problem checks whether two finite graphs are isomorphic. The complexity of this problem is unknown and it is widely believed that it is between deterministic polynomial-time and NP-completeness [75]. Isomorphism is an *exact graph matching*, that is, it can be used as a distance for comparison that provides a binary outcome: the graphs either are isomorphic, and then identical, or they are not. This is a very poor information, and very difficult to obtain. To efficiently compare graphs, we would like to have an *inexact graph matching*, that is a range-valued distance with the property that it converges to zero as the networks approach isomorphism.

The need to have accurate and effective tools to compare networks has pushed the research in many different directions, leading to a wide variety of methods and algorithms. We now present a short review of the main, most used and interesting approaches to network comparison found in literature, pointing out that the scientific production concerning this topic is huge and we cannot consider it exhaustively here.

1.1 Classification of methods

A clear classification of the various methods proposed over the years is not present in the literature. This is mainly due to the fact that there are plenty of different ideas and approaches, each one tailored to a given application field and to the specific task the comparison had to be performed.

Emmert-Streib, Dehmer, and Shi [28] made a quite complete review of the existing methods, though they did not consider all the approaches we found in literature. They propose to classify methods with respect to the type of graphs they can be applied to: methods for deterministic and for random graphs. A deterministic network is defined as a graph $\mathcal{G}$ with constant edge and vertex sets, where constant means that neither over time nor in parallel does not exist another graph $\mathcal{G}'$ with

different edge and vertex sets which represent the same object represented by $\mathcal{G}$. A random network is defined as a graph that is not deterministic. For example, the Protein-Protein Interaction (PPI) networks of humans (as well as of other species) is a random graph, because it is not known if the same structure is common to all humans and if this structure is always the same over time. Emmert-Streib, Dehmer, and Shi proposed an application-driven classification, arguing that the nature of the underlying network is an important starting point to better understand advantages and drawbacks of each method and to provide a more interdisciplinary point of view.

In this work we want to propose a new kind of classification which is more focused on the conceptual meaning of the comparison performed by the method, with interesting consequences in applications. We divide the comparison methods based on whether the induced distances are dependent from the correspondence of the nodes of the compared graphs or instead are independent on it. In the latter, ideally every kind of graph (different sizes, densities, even from different application fields) can be compared: indeed these type of methods try to summarize the global topological structure of the graphs into one or more statistics, which are compared to produce the distance. This gives a notion of distance that reflects the global difference in the topological structure of the networks. The former approach instead requires that the correspondence of the nodes of the two graphs is known: this means that only graph of the same size and coming from the same application domain can be compared; some properties of the nodes (which can be some statistics of the connectivity or of the importance) are taken into account, and are compared pairwise to produce the output distance. This classification will be better explained in Chapter 2.

Tthe aforementioned classifications are both valid adn useful, yet we prefer to adopt another criterion to present some of this methods more in detail. We decide to group them according to the approach used to perform the comparison, that is, which particular property or characteristic of the network is used to set up the distance measure. We argue that this kind of grouping can give interesting insights in the wide variety of the possible methodologies that can be used to perform network comparison. The grouping is the following:

- methods that uses global statistics of the networks [53, 66, 77];
- methods based on community analysis [62];
- alignment-based and alignment-free methods, which are names that denote methods mainly coming from biology [1, 30, 50, 66, 67, 69, 77];
- spectral methods [35, 76];
- some other methods that cannot be grouped in none of the previous classes, but are anyway very interesting for the particular approach they use [6, 27, 47, 48, 54].

We now give a closer look at each class of methods in the following sections.

1.2 Methods based on global statistics

The simplest and naïve way to compare two networks is to compare their global statistics such as degree distribution [66, 77], diameter [66, 77], clustering coefficient [53, 66, 77], and others, or some kind of centrality, for instance betweenness centrality [53]. Hence, using these distances, two graphs result similar if they have similar values of the global statistics chosen for the comparison.

This approach is very simple and in general very efficient, since most of the network's global statistics are computable in linear time with respect to the number of nodes or edges. Moreover, choosing a proper global statistics, we are able to compare all classes of networks, from undirected and unweighted to directed and weighted.

Despite this ease of use, the method has a number of critical drawbacks:

- If two networks have similar values of the chosen statistics, this does not mean that the two networks are topologically similar: indeed, it is proven that one can build two very different graphs, even from different generative models, which have similar values of their global statistics [66].
- The global statistics used to perform the comparison may fail in catching important local and topological feature of the two networks.
- If the data used to build the networks are noisy or incomplete, the computed values of the global statistics are biased with respect to their true values in the true but unknown networks.

For these reasons, it turns out that basing network comparison upon the network's global statistics often gives misleading results. More accurate and reliable methods are needed for the task.

1.3 Methods based on community analysis

The idea underlying community-based methods is that the mesoscopic properties of a network, i.e. its community structure, reveals important features and functionalities of the network itself [62]. Thus, the methods find the community structure of the two networks and then take summary statistics of the communities to perform the comparison. These methods are a first refinement of those based on global statistics: indeed, they are now computed for each community found, thus being able to catch more accurately some local features, dependent on the community structure. On the other hand, the main drawbacks of the global statistics-based methods still remain. Moreover, the community analysis itself requires a good community detection algorithm and, in any case, it is significant only for graphs that do have a community structure.

Onnela et al. [62] proposed an interesting community based method, based on a multi-resolution community analysis and on a particular choice of the statistics to describe the communities. First of all, they used the multi-resolution Potts

method [32], which is a generalization of the well-known modularity optimization algorithm, to perform community detection. In Potts method, the quantity to be optimized is called Hamiltonian, and it is dependent on a parameter λ which tunes the fragmentation of the network into communities. In other words, in their work Onnela et al. were able to analyse the community structure at different mesoscopic scale by tuning the parameter λ: it assumes values from $\Lambda_{\min}$ (when all the nodes are forced in a unique community) to $\Lambda_{\max}$ (when each node is forced to be a community). Given this capability to catch communities characteristics at different scales, they select three quantities as summary statistics to describe how the network disintegrates into communities:

- the Hamiltonian $\mathcal{H}(\lambda)$, which is a generalization of the network modularity and quantifies the energy of the system;
- the partition entropy $S(\lambda)$, which is a characterization of the community size distribution;
- the number of communities $\eta(\lambda)$.

These quantities are then normalized to be able to compare them across different networks and the resulting functions are called *Mesoscopic Response Functions* (MRFs). The algorithm used for the community detection was the Louvain algorithm [12], and the method is applicable to undirected networks, both weighted and unweighted.

The MRFs summarize many important features of the networks, among which the relative weights of inter- versus intra-community edges and how the community fragments (whether it halves or if only few nodes leave the community), and many others. Computing the MRFs of some real-world networks shows that they are very different from network to network, capturing the different characteristics of each one. Onnela et al. also performed a comparison among Erdős-Rényi, Barabási-Albert and Watts-Strogatz networks, showing that each class of models has its characteristic shapes of the MRFs. Thus, the MRFs can be a useful tool to represent and compare networks.

The definition of the distance between two graphs is however problematic. Onnela et al. define the distances between each one of the MRFs as the usual distance between functions, i.e. the integral of the absolute value of the difference. This gives the distances desirable properties, but then they have to be combined to produce the final output distance between the graphs. Onnela et al. proceed by evaluating the distances between each pair of network in their dataset, and then taking as distance the first principal component obtained performing a Principal Component Analysis on those distances. This approach is quite unusual, since the output distance depends on the dataset that is being analysed, and the result coming from two different datasets cannot be compared.

Onnela et al. applied their method to a dataset composed of 189 networks belonging to 12 different categories assigned a priori. They computed the distances between each pair and built a dendrogram to summarize the results Figure 1.1. The dendrogram shows that the method is able to gather together the networks of

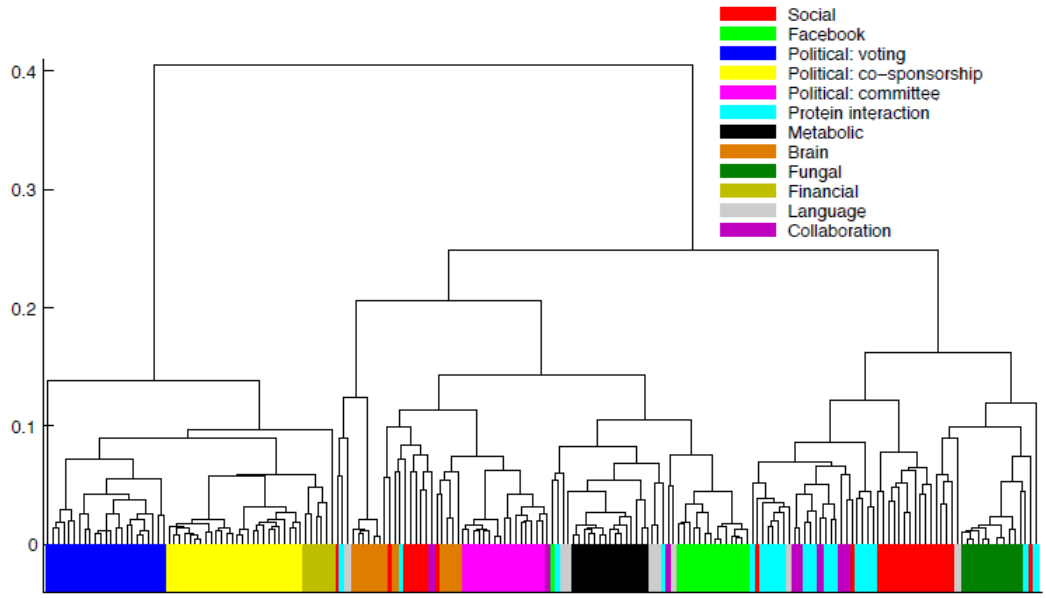


Figure 1.1: Dendrogram (built using average linkage) obtained computing MRFs distances between the 189 networks analysed in [62]. Leaves are coloured with respect to the class they belong to.

a few classes, while for many of the classes the networks are more scattered. The interpretation that Onnela et al. gave relies on the fact that contiguous networks in the dendrogram are assigned to the same cluster. This is true only if the networks are clustered together at small distances, and this happens for some classes; but they do not provide any analysis of the global accuracy and of the ability of the method in recovering the original subdivision in 12 categories.

1.4 Alignment-based and alignment-free methods

Biology is an application field with a very prolific research on network comparison methods. Biological data (molecules, proteins, genes, interaction between species, etc.) can often be represented as networks, and the ability of comparing them can give important insights. In particular, network analysis can be very useful if applied to protein-protein interaction (PPI) networks, where the nodes are the different proteins of a species, and the edges identify the interacting proteins. Two main approaches was developed in this field to compare PPI networks: *alignment-based* and *alignment-free* methods. They compare graphs from different perspectives. Alignment-based methods are aimed at finding a mapping between the nodes of the two compared networks trying to preserve many edges and to maximize the largest possible common subgraph. This approach can give evolutionary insights (the large subgraph identified can correspond to a evolutionary conserved part of the network, thus enabling the construction of a phylogenetic tree), important information on the functionality of the aligned parts of the networks, as well as structural similarities among proteins. On the other hand, alignment-free methods are aimed at comparing the overall topological similarity of the networks, without providing any mapping

among the nodes. This approach is more suitable to understand to which network model a real-world network is best fitted, to analyse the evolution of a network over time, and to cluster networks on the base of their topological similarity (and in this case, being able to reconstruct phylogenetic trees as well) [78]. In general, alignment-based approaches are much more computationally demanding.

1.4.1 Alignment-based methods: MI-GRAAL

We first present an alignment-based approach, called Matching-based Integrative GRAph ALigner (MI-GRAAL) [50], that performs global network alignment. Given two graphs $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$ with $|\mathcal{V}_1| \leq |\mathcal{V}_2|$, a global network alignment is a total and injective function $f : \mathcal{V}_1 \to \mathcal{V}_2$, where total means that all elements of $\mathcal{V}_1$ are mapped into some elements of $\mathcal{V}_2$ [50]. In practice, the resulting alignment ensures that all nodes of the first (smaller) graph are aligned to different nodes in the second (larger) graph. Note that the global network alignment problem is NP-complete (due to the underlying subgraph isomorphism problem) [19], then only approximated algorithms and solutions can be provided. As a measure of the topological quality of the alignment f, the *Edge Correctness* (EC) is defined:

$$EC = \frac{|f(\mathcal{E}_1) \cap \mathcal{E}_2|}{|\mathcal{E}_1|},$$

where $f(\mathcal{E}_1) = \{(f(u), f(v)) : (u, v) \in \mathcal{E}_1\}$. This quantity represents the percentage of edges in $\mathcal{E}_1$ that are aligned to edges in $\mathcal{E}_2$. The edge correctness is a similarity score: an EC of 100% indicates that the two graphs are identical, or that the smaller one is exactly contained in the larger one, a low EC indicates that the two graphs are so dissimilar that can be hardly aligned. The goal is to find algorithms that maximize the EC, thus maximizing the number of aligned edges.

The MI-GRAAL algorithm works by assigning to each pair of nodes, one from $\mathcal{V}_1$ and one from $\mathcal{V}_2$, a *confidence score*, built taking into account various similarities between the nodes; then, it aligns pair of nodes starting from those which have the highest confidence score, so that in principle very similar nodes in the two graphs are paired in the alignment. More in detail, the algorithm works as follows. The first step is to build the confidence matrix $C = [c_{ij}]$, $i \in \mathcal{V}_1$, $j \in \mathcal{V}_2$, whose entries c_{ij} denote the confidence that the algorithm has in aligning node i of $\mathcal{G}_1$ with node j in $\mathcal{G}_2$. The confidence scores c_{ij} are built by combining some similarity measures between nodes. In particular, MI-GRAAL can build matrix C using any kind and any number of these similarity measures (for instance the degree, the clustering coefficient, the betweenness centrality, or, for the biological applications, some scores giving information about the sequences of aminoacids in the proteins), which can encode different kind of information. We call $X_k = [x^k_{ij}]$ the matrix containing the similarity scores between nodes related to the k-th chosen similarity measures. Since a perfect alignment should minimize each similarity measure k, the confidence score c_{ij} is defined as

$$c_{ij} = \sum_k \mathrm{conf}^k_{ij},$$

where conf_{ij}^k is the fraction of elements in the i-th row of matrix X_k that are strictly greater than x_{ij}^k (here two nodes i and j are considered similar if x_{ij}^k is small). Then, if for instance x_{ij}^k is the smallest entry in row i of matrix X_k, then the similarity measure k is 100% confident to align node i with node j. All the confidences provided by each similarity matrix are summed up to obtain the final confidence matrix C, which align node pairs in decreasing order with respect to their confidence scores c_{ij}; if there are some tie, they are broken randomly. This definition of the confidence matrix C allows to overcome some usual drawbacks in other alignment-based method, such as the higher number of ties arising if only one similarity measure is considered and the contradiction between different similarity measures; moreover, this approach makes the algorithm more robust to minor errors in the individual similarity matrices and yields more stable alignments, in the sense that they differ very little in different runs.

Kuchaiev and Pržulj [50] applied this algorithm to PPI networks. Besides showing that MI-GRAAL is capable to reach a much higher EC than the existing alignment-based algorithms, they also managed in reconstructing the phylogenetic tree (i.e. the evolutionary tree) of five herpes viruses from their PPI networks using only topological similarity scores as parameters for MI-GRAAL and the EC as distance between the networks.

The results obtained by Kuchaiev and Pržulj prove that it is also possible to use MI-GRAAL for the task of clustering. Furthermore, the method is very flexible and customizable, since it allows to use any kind of user-defined similarity score between nodes to build the confidence matrix for the alignment. This has important consequences. First, MI-GRAAL enables the comparison between directed, weighted and directed and weighted graphs, just by choosing proper similarity scores between the nodes, such as (in- and out-) strength. Secondly, MI-GRAAL can be used to compare very different types of networks, not only biological but also from other application fields; moreover, the alignment provided can give new insights, especially for those graphs which have strong correspondence between nodes. Think for example about the comparison of two financial or trade networks: the alignment can give information about the role or the correlation of the individual nodes in the two different graphs. On the other hand, the main drawback of MI-GRAAL is its computational efficiency: the running time is more than quadratic in the number of nodes, so that the algorithm is not suitable to analyse large graphs.

1.4.2 Alignment-free methods: Graphlets-based measures

The other class of methods most used in biology is that of alignment-free methods. As already explained, the name is related to the fact that this kind of methods does not provide an alignment between nodes, but it compares the overall topological structure of the networks. In almost all alignment-free methods the topological comparison is performed using *graphlets*, which are connected sub-networks with a small number of nodes [66]. Graphlets are different from network motifs [59]: the latter are patterns of interconnections that occur in a complex network at numbers which are significantly higher than those in randomized networks; therefore, motifs

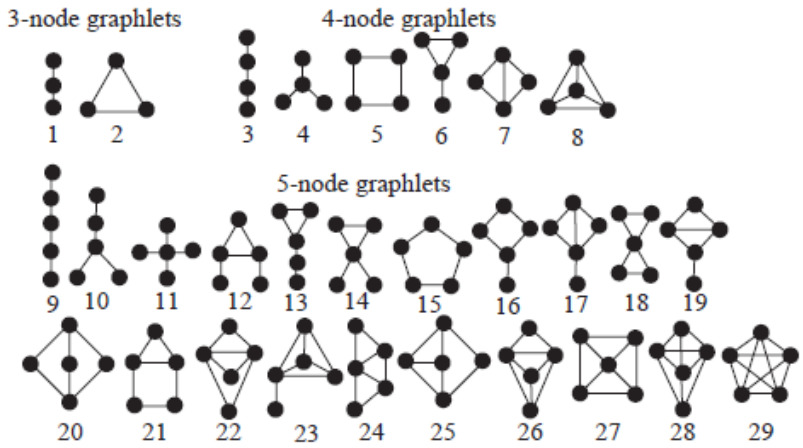


Figure 1.2: All the 29 graphlets from 3 to 5 nodes [66].

can be different from network to network, while graphlets are always the same and represent and catch specific topological structures. Moreover, since graphlets have this unique definition, they can be used as a summary statistic to describe the network, as we will explain hereafter. Although graphlets are well suited for comparing the overall topology of networks, they can also be used to build similarity scores for alignment-based algorithm, as in [49, 57]. We now introduce more in detail graphlets and the alignment-free distance measures.

Relative Graphlets Frequency Distance (RGFD)

Graphlets were defined for the first time by Pržulj, Corneil, and Jurisica [66] as "small connected non-isomorphic induced subgraphs of a large network" [67]; the graphlets from 3 to 5 nodes are presented in Figure 1.2. Only graphlets with these number of nodes were considered: on one hand, larger graphlets contain redundant information, because in their structure we find repeated many times the smaller graphlets; on the other hand, the number of graphlets grows exponentially in the number of their nodes, so that it is computationally more efficient to consider only small graphlets.

The *Relative Graphlets Frequency Distance* (RGFD) is introduced and defined in [66]. The basic idea is that graphlets are structures able to describe the local topology of a graph and thus the number of occurrences of the graphlets in a network summarizes its topological properties and characteristics. Then, a first simple way for graph comparison is to simply compare the number of occurrences, or frequencies, of the graphlets in two graphs. The graphlets frequencies of a graph $\mathcal{G}$ are defined as

$$f_i(\mathcal{G}) = \frac{N_i(\mathcal{G})}{T(\mathcal{G})}, \quad i \in \{1, \dots, 29\},$$

where $N_i(\mathcal{G})$ is the number of graphlets of type i and $T(\mathcal{G}) = \sum_{i=1}^{29} N_i(\mathcal{G})$ is the total number of graphlets in the graph. Then, the RGDF $d(\mathcal{G}, \mathcal{H})$ is defined as

$$d(\mathcal{G}, \mathcal{H}) = \sum_{i=1}^{29} |F_i(\mathcal{G}) - F_i(\mathcal{H})|, \tag{1.1}$$

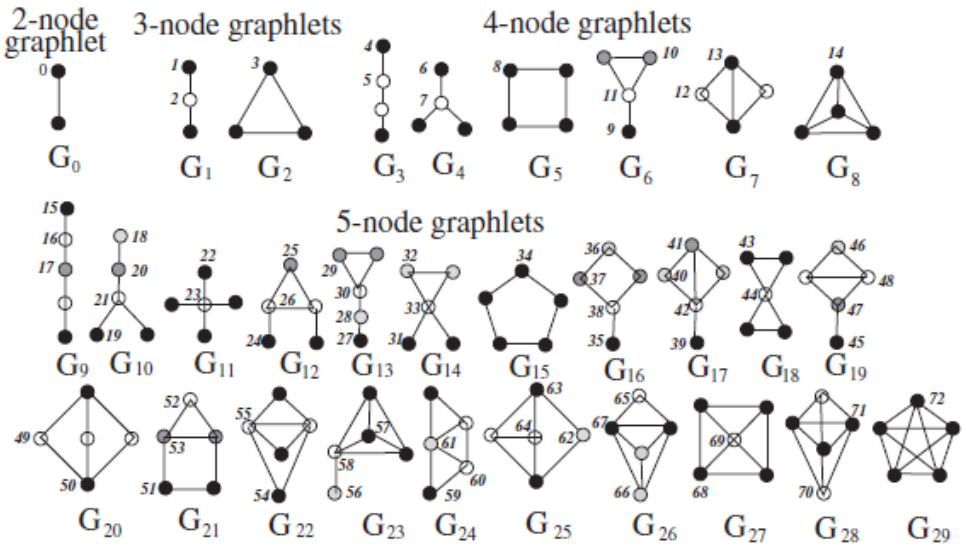


Figure 1.3: All the 73 automorphism orbits of graphlets from 3 to 5 nodes [67].

where $F_i(\mathcal{G}) = -\log_i(f_i(\mathcal{G}))$. The logarithm of graphlets frequencies is taken to avoid that the most frequent graphlets dominate the distance measure, since the graphlets frequencies may differ by several order of magnitude.

In her work, Pržulj, Corneil, and Jurisica [66] used RGFD to compare real-world networks to graph models, to understand which model best fitted the data. In particular, they compared some PPI networks to Erdős-Rényi, Barabási-Albert and geometric random graphs and they showed that, according to the new measure and some other classical network statistics, the PPI networks were better described by geometric random graph, rather than by scale-free models as many studies showed previously.

However, RGFD, as it is defined, is a very naive measure and can be improved a lot, as discussed in the next sections.

Graphlet Degree Distribution Agreement (GDDA)

A first improvement of the RGFD is presented by Przulj [67]. She proposed a generalization of the degree distribution based on graphlets to set up a distance which performed network comparison considering several graph characteristics.

The degree distribution is defined as the number of nodes with degree k, i.e. the number of nodes "touching" k edges. An edge is the same as the graphlet G_0 in Figure 1.3; thus, we can say that the degree distribution measures the number of nodes that touch k graphlets G_0. As we took G_0, we can equivalently consider any of the other 29 graphlets. We only need to take care about an important topological property of some graphlets. Take for instance graphlet G_1. If we ask how many nodes touch G_1, we should distinguish whether the node touches G_1 in its middle node or in its end nodes, since they are topologically different. This distinction is due to the *automorphism* property of a graph, that is a map from its nodes to themselves preserving the topological structure. Then, if a graphlet contains nodes

that are topologically different with respect to the automorphism property, we assign the nodes to different *automorphism orbits* (or just *orbits* for brevity). In this way, starting from the 29 graphlets from 3 to 5 nodes, we identify 73 different orbits. Therefore, the generalization of the degree distribution is as follows: we create 73 *graphlet degree distributions* (GDD) counting how many nodes touch a graphlet in one of its automorphism orbits. To do an example, we will have one GDD from graphlet G_2 counting the number of nodes that touch k times a triangle (in each one of the vertices, since they are all topologically equivalent); but we will have two different GDDs from graphlet G_1, one counting the number of nodes that touch k times a G_1 in an end point, and the other one counting the number of nodes that touch k times a G_1 in the middle node, since end points and middle node belong to two different automorphism orbits. Thanks to this generalization, we can compare graphs imposing a large number of constraints on their structure and then we are more confident that two graphs are really similar if they are similar in all of these statistics. Note that it is in principle possible to impose a larger number of constraint just by considering larger graphlets, with the only limit given by the computational time.
In [67] it is also noticed that the 73 distributions are unlikely to be statistically independent, but the details of this issue were not investigated there and are better explained in [77].

The *Graphlet Degree Distribution Agreement* (GDDA) is defined as follows [67]. Given the two networks $\mathcal{G}$ and $\mathcal{H}$ to be compared, first of all their 73 GDDs are computed. We call $d^j_{\mathcal{G}}(k)$ the graphlet degree distribution, which is the number of nodes in graph $\mathcal{G}$ touching k times orbit j. Each $d^j_{\mathcal{G}}(k)$ is scaled as

$$S^j_{\mathcal{G}}(k) = \frac{d^j_{\mathcal{G}}(k)}{k}$$

to take into account that most of the information of the distribution is contained in the lower degrees. This information was derived from the analysis of the real datasets discussed in the paper. To make all the distributions comparable to each other, a normalization is performed:

$$N^j_{\mathcal{G}}(k) = \frac{S^j_{\mathcal{G}}(k)}{T^j_{\mathcal{G}}}\,,$$

where $T^j_{\mathcal{G}} = \sum_{k=1}^{\infty} S^j_{\mathcal{G}}(k)$ is the total area under the j-th GDD. This sum is in practice finite since k is not really unbounded, but it is finite due to the finite size of the graph. In the last step, the distance between $\mathcal{G}$ and $\mathcal{H}$ in the j-th GDD is defined as the euclidean distance of the corresponding normalized graphlet distribution distances:

$$d^j(\mathcal{G},\mathcal{H}) = \frac{1}{\sqrt{2}}\sqrt{\sum_{k=1}^{\infty}\left(N^j_{\mathcal{G}}(k) - N^j_{\mathcal{H}}(k)\right)^2}\,,$$

where again the sum is in practice finite. Instead of the distance, Przulj preferred to define an *agreement*, that is just

$$A^j(\mathcal{G},\mathcal{H}) = 1 - d^j(\mathcal{G},\mathcal{H})\,.$$

In the end, the GDDA between two graphs is defined as the geometric or the arithmetic mean among all the j-th agreements:

$$A_{\text{arith}}(\mathcal{G},\mathcal{H}) = \frac{1}{73}\sum_{j=0}^{72} A^j(\mathcal{G},\mathcal{H})\,, \quad A_{\text{geo}}(\mathcal{G},\mathcal{H}) = \Big(\prod_{j=0}^{72} A^j(\mathcal{G},\mathcal{H})\Big)^{\frac{1}{73}} \tag{1.2}$$

which are both quantities belonging to $[0,1]$ where 1 means total agreement and 0 means very dissimilar graphs.

As in [66], Przulj [67] applied GDDA to better understand which graph model performs best in fitting PPI networks. She compared 14 PPI networks related to different species and built with various experimental techniques to different realizations of 4 graph models (Erdős-Rényi, Erdős-Rényi with fixed degree distribution, scale-free Barabási-Albert and 3D-geometric networks). Again, she found that the geometric random graph model is the one that has the better agreement with all the PPI networks but one. Moreover, these agreement values were very close to the mean agreement measured among different realizations of the same graph model, giving a strong evidence that, at least under this kind of measure, PPI networks are better modelled by geometric random graphs rather than by scale-free networks.

Graphlet Correlation Distance (GCD)

A detailed study of the interdependencies and correlations among graphlets has been carried out by Yaveroglu et al. [77]. They noticed that some orbits are redundant, in the sense that their count can be derived from the counts of other different orbits. As an example, take graphlet G_1 and, for clarity, call C_i the i-th *graphlet degree* of a node, i.e. the number of times the node touches the orbit i of a graphlet (note that C_0 is the degree of the node). The set of all C_i, $i \in \{1,\dots,73\}$, for a node is called *graphlet degree vector*, or *graphlet degree signature*, of that node [58]. Consider now a node; its neighbours are either connected, or they are not. In the first case, they contribute to the count C_3 of the triangles that the node touches; in the second case, they contribute to the count C_2. These are the only two alternatives, so that the number of ways in which C_0 neighbours of a node can be connected is $\binom{C_0}{2} = C_2 + C_3$. Hence, one of the three orbit is redundant, since having two of them, the third can be computed. This reasoning can be applied to the different orbits and leads to the identification of many redundant orbits; in particular, there are 56 non-redundant orbits among the 73 orbits for up to 5-nodes graphlets and there are 11 non-redundant orbits among the 15 orbits for up to 4-nodes graphlets. The non-redundant orbits for graphlets up to 4-nodes are shown in Figure 1.4; note that this is only one of the possible several choices for the non-redundant orbits. Thus, by eliminating these redundancies, one can get a more sensible measure and gain in computational efficiency. Anyway, even eliminating redundancies, some dependency, i.e. correlation, among graphlets remains. In particular, there are fewer dependencies between 4-node graphlets than between 5-node ones and this means that taking into account only up to 4-node graphlets should result in a less noisy statistics (and again this allows not to compute all the graphlets counts up to 5 nodes).

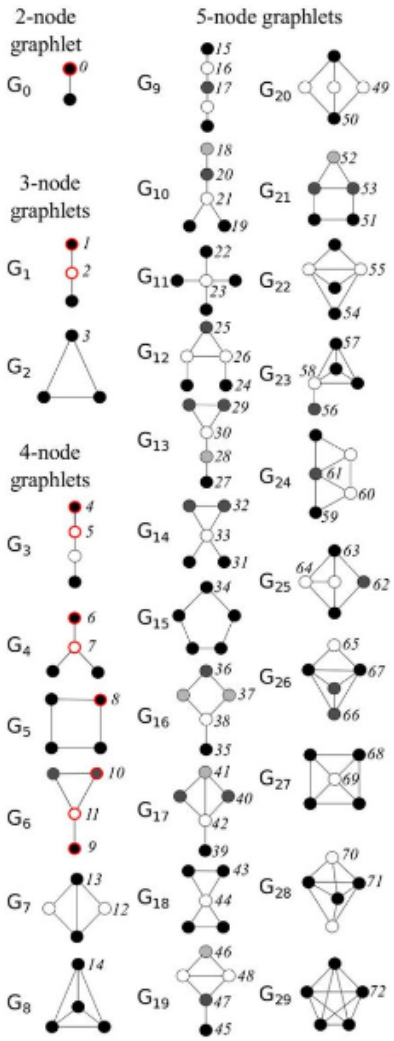


Figure 1.4: Graphlets from 2 to 5 nodes [77]. A possible choice of the non-redundant orbit for up to 4-nodes graphlets are highlighted in red.

The new network graphlet-based statistic designed by Yaveroglu et al. [77] takes advantage from this analysis. They consider the graphlet degrees of the nodes counting only the 11 non-redundant orbits for up to 4-node graphlets, although the same construction can also be applied when considering redundant orbits and up to 5-node graphlets, or both. They then built a matrix whose dimensions are $N \times 11$, where N is the number of nodes in the network; in practice, the matrix is built appending row by row the graphlet degree vector of each node. The last step is to compute the new network statistic, called *Graphlet Correlation Matrix* (GCM), which is simply another matrix obtained computing the Spearman's Correlation coefficient [72] between all pairs of columns of the above described matrix. The result is that the whole local topology of the network is summarized in a 11×11 matrix, whose entries has values in $[-1, 1]$. The GCM of a graph can be qualitatively analysed to better understand the topological structure of the network; for instance, it was observed that in a Barabási-Albert network orbits 0, 2, 5 and 7, whose structure suggest the existence of hubs, form a cluster of correlated orbits, while orbits 1, 4, 6 and 9, whose structure is characteristic of existence of a large number of degree 1 nodes, form another cluster of correlated orbits as well, and the two clusters are not correlated each other [77].

The GCM statistics also induces a new distance measure between two graphs, called *Graphlet Correlation Distance* (GCD). Given two graphs $\mathcal{G}$ and $\mathcal{H}$, the GCD is defined as the Euclidean distance of the upper triangular parts of the matrices $\mathrm{GCM}_{\mathcal{G}}$ and $\mathrm{GCM}_{\mathcal{H}}$. To take into account the different dimension of graphlets and the redundant orbits that one can consider, Yaveroglu et al. distinguished between four different GCD distances: GCD-11, that is the one explained above; GCD-15, that considers all the orbits for up to 4-node graphlets; GCD-56, that considers only the non-redundant orbits for up to 5-node graphlets; and GCD-73, which takes into account all the orbits for up to 5-node graphlets.

Yaveroglu et al. first evaluated on synthetic networks of different size and density how well their distance could discriminate among 7 different classes of network models, among which Erdős-Rényi, scale-free Barabási-Albert and geometric random graph, and they compared these results to those obtained using different network measures, such as degree distribution, clustering coefficient, diameter, spectral Euclidean distance, RGFD and GDDA. Moreover, they considered all the four Graphlet Correlation Distances they defined, to find out which one performed best. To compare all these network distances, they use a Precision-Recall framework (see Appendix B) and computed the Area Under the Precision-Recall curve (AUPR), which is an index of the average quality of the clustering provided by the method. Moreover, a robustness test to noise and missing data is performed in the two following ways: they rewired randomly up to 90% of the edges and they deleted up to 40% of the nodes in some of the simulated networks. The results of these tests showed that GCD-11 is capable to identify and to group together the different realization of the 7 network models, and the grouping is even better (i.e. the realizations are found to be at smaller distances) when graphs of the same sizes and densities are compared. As far as the performance among all the network distances is concerned, GCD-11 is again the best performing one with the highest AUPR. It is also confirmed that is the best performing of the four GCD measures, since

it has fewer orbit dependecies and no redundancies. Also in the robustness tests, GCD-11 is the best performing measure with respect to introduction of noise and presence of missing data.

Real-world networks were also considered. A first analysis was aimed at finding the best fit of some real-world networks from 5 different domains (autonomous systems, Facebook, metabolic, protein structure and world trade networks) to the previously used 7 graph model. The results showed that most of the real-world networks are best modelled by geometric random graphs. Another deeper and somehow different analysis was carried out on the world trade networks from 1962 to 2010. The information given by the GCD-11 distance along with an analysis of the graphlet-based position of a country were able to recover many historic dynamics that are well supported in economics literature.

Extension to directed networks

All the previously presented graphlet-based distance measures were originally defined for undirected and unweighted graphs. Sarajlić et al. [69] presented an extension of each one of those measures to directed networks by introducing *directed graphlets*, shown in Figure 1.5. Directed graphlets are defined as small induced subgraphs of a larger directed network, without anti-parallel directed edges [69], i.e. if a graphlet contains an edge from node u to node v, then it cannot also contain the edge from node v to node u. This mean that, if a network contains anti-parallel edges, each one of them will increase the count of two different graphlets and not of the same one.

Sarajlić et al. extended all the statistics and the distance measures defined in [58, 66, 67, 77]. In particular:

- the *Directed Graphlet Degree Vector* of a node is a vector whose i-th component is the count of the directed graphlets touching that node at orbit i;
- the *Directed Relative Graphlet Frequency Distance* (DRGFD) is defined analogously to Equation (1.1);
- the *Directed Graphlet Degree Distribution Agreement* (DGDDA) is defined analogously to the arithmetic mean in Equation (1.2);
- the *Directed Graphlet Correlation Distance* (DGCD) is defined as the Euclidean distance between the upper triangular parts of the two *Directed Graphlet Correlation Matrices* (DGCMs) built as explained in [77], using the directed graphlet degree vectors. Also in this case redundancies between orbits are identified and can be eliminated, leading to the definition of different DGCD measures. Sarajlić et al. [69] consider DGCD-129, built with all the 129 orbits for up to 4-nodes directed graphlets, and DGCD-13, built with only the 13 non-redundant orbits for up to 3-nodes directed graphlets.

Sarajlić et al. tested all their directed measures to find out which is the best performing. To do this, they used the usual framework: they generated some realizations from 6 different network models (among which directed Erdős-Rényi, directed scale-free Barabási-Albert, and directed geometric random graphs) and

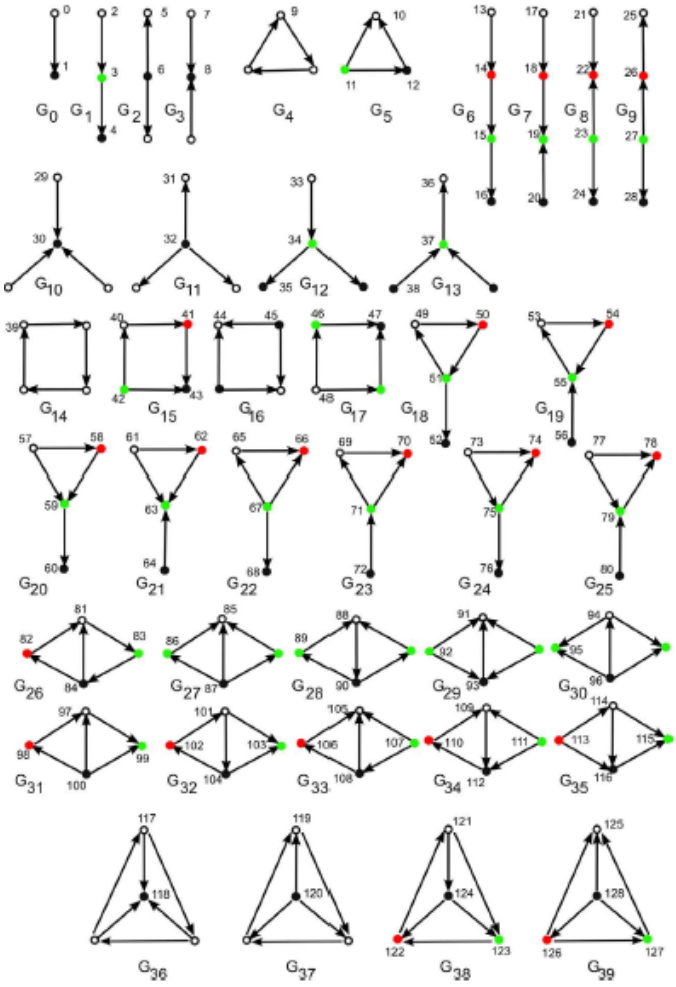


Figure 1.5: The 40 up to 4-node directed graphlets [69]. In each graphlet, nodes belonging to the same automorphism orbit are labelled with the same colour.

they used the aforementioned measures to cluster them. The results were evaluated with Precision-Recall and ROC curve analysis [4] and computing AUPR (see Appendix B) and the area under the ROC curve (AUC). The results showed that DGCD-13 was the best performing measure among the directed graphlet-based distances, probably due to the fewer dependencies between 2-node and 3-node graphlets. Moreover, they also showed that the analogous undirected measures (RGFD, GDDA, GCD) were not able to achieve performances as large as the ones of directed measures; this means that taking into account directionality leads to more accurate analyses. Robustness to edge rewiring and deletion were also tested, showing again a superiority of DGCD-13.

The analysis of directed world trade networks from 1962 to 2013 was also considered, to extend the analysis done on undirected world trade networks in [77]. The usage of the new measures gave more relevant information and was able to uncover the economical structure of the networks and the relationship and the economic roles of the various countries, as well as to predict the values of some economic indicators.

Computational efficiency of graphlet counts

The computation of all the graphlets and the graphlet degree vectors is in principle a very demanding task. Given a graph with N nodes and L edges and using a complete enumeration strategy, the worst case running time for counting 2 to k-node graphlets (both for undirected and directed case) is $O(N^k)$, and a tighter upper bound is $O(Nd^{k-1})$, where d is the maximum node degree in the network [69, 78]. In practice, these pessimistic upper bounds are never reached because of the sparsity of most of real-world networks, but it is clear the need of wiser counting strategies and faster algorithms to make the methods scalable to very large graphs.

Hočevar and Demšar [43] proposed an effective strategy to compute undirected graphlet counts. Their strategy is based on the fact that all the counts for k-node graphlets can be derived starting from all the counts of $k-1$-node graphlets and a complete enumeration of only a single k-node graphlet, by solving a system of linear equations encoding the contribution of each node in different orbits. The single k-node graphlet to be enumerated is chosen to be a complete (all-to-all) graphlet, because, given the usual sparsity of graphs, this is less likely to be found and then much faster to be enumerated. Using this strategy results in gaining at least an order of magnitude in the computational time: if we want to compute 5-node graphlet counts, the computational complexity is the same as computing 4-node graphlets counts. Specifically, the total time complexity for computing all orbits of 4-node graphlets is $O(Ld + T_4)$, where $O(T_4)$ is the time needed to enumerate complete 4-node graphlets, which is done in $O(Nd^3)$. Instead, the total time complexity for computing all orbits for 5-node graphlets is $O(Ld^2 + T_5)$, where again $O(T_5)$ is the time needed to enumerate complete 5-node graphlets, which is done in $O(Nd^4)$. Then, the algorithm has the same upper bound of the naive enumeration algorithms, but experiments show that the contribution of the terms $O(T_4)$ and $O(T_5)$ are negligible in a large interval of graph densities and that the actual running time are smaller at least by an order of magnitude. The space complexity needed for the algorithm is $O(N + L)$ for counting 4-node graphlets and $O(Ld)$ for 5-node graphlets. Moreover, this strategy allows for fully parallelization, so that a better speed up is possible. The resulting algorithm was called ORCA, which stands for ORbit Counting Algorithm.

In a later work, Aparıcio, Ribeiro, and Silva [3] introduced a novel methodology for computing graphlet counts. The novelty of the algorithm is based on a particular data structure, called *G-Trie* [68], which is able to deal with both undirected and directed graphlets. Its efficiency in enumerating graphlets is due to two main algorithmic ideas. First, instead of enumerating all the graphlets to be found, the algorithm starts by looking at common subtopologies between graphlets, and then it performs the necessary expansions to find larger graphlets. Secondly, symmetry conditions are added to avoid counting multiple time the same graphlet (for instance, when considering permutations of the graphlet's nodes one finds again the same graphlet). The computational efficiency of the G-Trie algorithm was evaluated against other existing methods, including ORCA. G-Trie greatly outpeformed all existing methods, and it was twice faster than ORCA in counting undirected

graphlets. Moreover, G-Trie is able to enumerate graphlets of any size, as opposed to other methods which count up to 5-node graphlets, and to enumerate directed graphlets, making it a more general tool for graphlet counting. Though, it is not clear which are the theoretical upper bound for time and space complexity, to properly compare it against other existing methods.

1.4.3 Alignment-free methods: NetDis

As far as the study of biological networks, and in particular of PPI networks, is concerned, Ali et al. [1] proposed another alignment-free method based on graphlet counts. The idea underlying their approach is not to merely compare the graphlet counts, but to compare graphlet counts in neighbourhoods of network nodes. In other words, they try to compare networks by comparing the graphlet counts in some, possibly overlapping, local subparts of the networks. This is motivated by the fact that some factors, like graph density and size, strongly influence the graphlet counts in the entire network; thus, they would like to have a measure capable to compare graphs independently from size and density. They chose as local subparts, to be taken into account, the 2-step ego-networks of each node. Given a node u, its 2-step ego-network is the subgraph consisting in all the nodes within two edges from u and all the edges among these nodes. Obviously different radii can be choosen for the ego-networks, but this parameter has to be tuned carefully to catch only local parts of the graph and at the same time to catch reasonable local variability.

The algorithm works as follows. First, all the graphlet counts (for graphlets up to 5 nodes) are computed from 2-step ego-network of each node. Then, all these counts are normalized with respect to the expected counts from a null model. Since no good probabilistic model of PPI network was available, they used a gold-standard model as an approximation for the expected counts. $S_w(\mathcal{G})$ denotes the sum, over all ego-networks, of the normalized counts for graphlet w in graph $\mathcal{G}$. In the last step, the quantity $netD_2^S(k)$ is computed as

$$netD_2^S(k) = \frac{1}{M(k)} \sum_{w \text{ of size } k} \left(\frac{S_w(\mathcal{G}) S_w(\mathcal{H})}{\sqrt{S_w(\mathcal{G})^2 + S_w(\mathcal{H})^2}} \right),$$

where $k = 3, 4$ or 5 denotes the size of the graphlets and $M(k)$ is a normalizing constant to force $netD_2^S(k) \in [-1, 1]$ by Cauchy-Schwarz inequality. The distance measure NetDis is defined as

$$netd_2^S(k) = \frac{1}{2}\Big(1 - netD_2^S(k)\Big) \quad \text{for} \quad k = 3, 4, 5\,,$$

which belongs to $[0, 1]$. Note that the NetDis measure depends on the size k of the considered graphlets, so that this is an additional parameter to set when using the distance.

The performances of NetDis were evaluated in the task of clustering different realizations, of various sizes and densities, from 5 different graph models. NetDis was able to perfectly discriminate among each class of graph models. It was also compared to MI-GRAAL [50] and MRFs [62] methods in the task of reconstructing

the taxonomies of the networks used in [62]. The results showed that NetDis was the best method both in accuracy and in computational time, while MI-GRAAL was very slow and failed to compare many pairs of graphs due to their too large size or density.

Yaveroglu, Milenkovic, and Przulj [78] more deeply analysed the performance of NetDis. They argued that Ali et al. [1] did not propose a fair evaluation, since their new alignment-free method was compared to MRFs, which is a completely different approach to network comparison, and to MI-GRAAL, which is an alignment-based method and therefore it has a different approach and different purposes, and it was not compared with other existing alignment-free methods, which are closer relatives; moreover, it was not evaluated within a systematic precision-recall framework. Yaveroglu, Milenkovic, and Przulj also depicted a number of crucial drawbacks in NetDis, among which the computational efficiency (NetDis counts many times the same graphlets, since the ego-networks are very likely to be overlapping) and the dependency on a gold-standard model. The results of the experiments they performed (again in the task of clustering synthetic networks of different sizes and densities from 7 different graph models, with a precision-recall evaluation, and comparing NetDis to other alignment-free methods) showed that NetDis was not as good as claimed. In particular, GCD-11 resulted to be the most accurate method with the highest AUPR, and NetDis showed a large difference of its performances when comparing networks of different sizes and densities, performances that are also very different when considering different gold-standard networks for the normalization of graphlet counts. In fact, it is not possible to choose a priori a good gold-standard model for all the real-networks to be compared, especially because each network may require a different null model. Ali et al. [2] tried to address the problem of the computational complexity and they found that using a sub-sampling strategy in which only 10% of the ego-networks are taken into account, NetDis is able to achieve results close to the ones obtained without sampling. However, the other major problems explained above still remain and make NetDis impractical to be used.

1.4.4 Alignment free methods: GRAFENE

Another recently proposed approach that uses graphlets is GRAFENE [30]. This method can be used to compare networks coming from all application fields, but it has a particular focus on the comparison of biological networks through the addition of some ad hoc expedients.

The method works as follows. First of all the graphlet degree vectors are computed for each node. Note that the graphlets that are considered are the 29 graphlets $G_1, \ldots, G_{29}$ in Figure 1.2, that is, the different orbits are not taken into account. Then the components of each graphlet degree vector are scaled between 0 and 1 just by dividing each component by the sum of all the counts of that graphlet in the whole network. Finally, Principal Component Analysis (PCA) is performed over the rescaled graphlet degree vectors and the first r components that account for at least 90% of the total variability are kept as a summary statistic for the

graph. The distance between two network is defined as

$$1 - d^{cos}(R_1, R_2)\,,$$

where d^{cos} is the cosine similarity and R_1, R_2 are the first r principal components (as previously defined) of the two graphs.

GRAFENE was first designed to perform protein comparison, which is the problem that aims at quantifying the similarity between proteins with respect to their sequence or 3-dimensional structural patterns. The new way of using graphlets through PCA was introduced, but, at the same time, to perform at best in the particular task of protein comparison, extended versions of the method were also designed. In particular, since a protein is better described when both sequence and 3-dimensional structures are considered, *ordered graphlets* [55] were extended and used to properly take into account the sequence information that each protein has; in this case, ordered graphlet degree vectors are computed by GRAFENE instead of the usual ones. Another improvement of the method was made about the different importance of aminoacids that may be very far away from each other in the sequence structure but may be very close in the 3-dimensional structure. Although these improvements made GRAFENE the best performing method in the comparison and classification of protein structure networks, it is worth notice that these improvements can only work for this specific task.

GRAFENE has some interesting features. The rescaling in the $[0, 1]$ interval of the graphlet counts makes this method more robust when analysing graphs of different sizes, as the numerical results show; moreover, it enhances the quality of the results given by the PCA, since very often graphlet counts vary by several orders of magnitude (in general, counts of 3-node graphlets are much larger than counts of 5-node graphlets). The PCA is a novel idea that enhances description of the results and reducibility of the problem, thus providing also a little speed-up in the computations. Indeed, performing PCA means to reduce the comparison of all the graphlets counts to the comparison of two small vectors of principal components for each pair of networks. Furthermore, PCA highlights the more relevant graphlets in terms of the variability of their graphlet counts in the network, so that one can get more precise information about the characteristics of the network's structure and identify the most relevant structural patterns.

The performances of GRAFENE were evaluated both on synthetic (Erdős-Rényi, Barabási-Albert and geometric random graphs) and real networks (protein structure networks) and within the usual precision-recall framework. The method was tested against other alignment-free (RGFD, GDDA and GCD) and alignment-based (GR-Align [55]) methods. The results on synthetic networks showed that GRAFENE is performing at least as well as the other alignment-free methods when comparing graphs of different sizes, and it is always better than them when comparing only graphs of the same size. On the other hand, the results on real networks showed that GRAFENE outperforms all the other methods, achieving the highest values of AUPR.

1.5 Spectral methods

In this class of methods the distance between two networks is defined as some kind of distance between the spectra of their respective representation matrix, which can be for instance the adjacency or the Laplacian matrix. Wilson and Zhu [76] define the spectral distance as the euclidean distance between the spectra. Gera et al. [35] propose to use a non-parametric statistical test to compare them, and take the p-value of the test as distance.

Spectral methods present a number of potential drawbacks [76]:

- The graph spectrum is highly dependent on the matrix representation (adjacency matrix, Laplacian, normalized Laplacian, and others) chosen for the graph. The choice influences the performances and the robustness of the method in the task of network comparison.
- Little changes in the graph's structure (for instance the removal, the addition or the switching of an edge) can produce large changes in the spectrum, thus producing incorrect estimates in the comparison.
- The problem of cospectrality arises: does two different (i.e. non isomorphic) graphs taken with the same representation always have a different spectrum? If not, we may wrongly classify as identical two different graphs.

The problem of cospectrality for general graphs was investigated by Haemers and Spence [40]. Analysing the number of cospectral graphs for networks up to 11 nodes and on the basis of a theoretical result in [37], they claimed that possibly almost no graphs of large size has a cospectral mate. Moreover, they observed that the matrix representation of the graphs influences the number of cospectral mates and in particular the signless laplacian (defined as $|\mathbf{L}|$) is the best representation, meaning that it produces the minimum number of cospectral graphs.

Wilson and Zhu [76] investigated which matrix representation, and thus which spectrum, is the best in identifying and representing the graph. They expect that if the spectrum is a good representation for a graph, then similar graphs will have similar spectra. They set up a simple classification experiment for this purpose, by generating 50 random graphs with 50 nodes and 200 edges. Each one of this graphs is assumed to be a separate class. They then build a second dataset with graphs to be classified in the following way: for each one of the first 50 graphs, they perform successive edit operations, which consists in the addition or the removal of a random chosen edge; in this way they expect to obtain very similar graphs from one edit operation to the successive. This procedure is repeated up to 40 edit operations. Then the resulting dataset is classified with a simple 1-NN classifier [42]. The results they provide show that for this task the best matrix representation is the Laplacian. Nonetheless, even if the results suggests that the Laplacian is the representation matrix which has the best properties, this experiment does not seem to have strong foundations. First of all, the experiment is done only for one value of dimension and density of the graphs; secondly, it is not clear what "random graphs"

means, i.e. from which graph model they generate their data. Above all, at the considered dimension (50 nodes) all graphs are more or less similar, even if they come from different generative models, and thus just after few edit operations we can get a graph that is equally similar to all the original graphs, thus potentially producing a large number of incorrect classifications. Moreover, taking a 1-NN classifier with this initial situation where the different classes are not well separated can lead to a very noisy and error-prone classification.

Gera et al. [35] proposed an alternative approach to compare the spectra of two graphs. They see the spectra as realizations from a certain distribution, claiming that similar graphs must have similar spectra distributions. Therefore they perform a nonparametric statistical test in which the alternative hypothesis is that the two spectra come from a different distribution. In this way, they require strong evidence to claim that the two spectra differ. The similarity between the two graphs is expressed as the p-value resulting from the test, with value 1 meaning that the two graphs are identical, and values near to 0 meaning that the two graphs strongly differ.

Despite the aforementioned drawbacks, in the continuation of this work we will consider the approach by Wilson and Zhu, because of the straightforward idea and the ease of implementation. The method was originally thought for undirected and unweighted graphs, and it is possible to extend it to undirected weighted graphs. An extension to directed graphs, both unweighted and weighted, is possible by taking as representation matrix the Symmetric Normalized Laplacian as defined in Equation (A.1) and in Appendix A.1, respectively. However, note that this representation can be applied to directed graphs but it considers them as undirected, so that the additional information carried by the edge directions is not exploited.

1.6 Other methods

Plenty of methods that do not use any of the previously presented approaches can be found in literature. We present here a small selection of the most recent, original and promising methods we found, also in terms of their applicability to the study of many, large scale, weighted and directed graphs.

1.6.1 DeltaCon

DeltaCon [48] is a method proposed to compare graphs with known node correspondence. The idea is to first compute a certain kind of similarity between each node pair in each graph, and then compare the two graphs by comparing their node similarities. This procedure is motivated by the fact that edges do not always have the same importance, even in undirected and unweighted graphs: for instance, a bridge between two highly connected regions, or a long distance connection, is much more relevant than the other edges for the structure of the graph in terms of flow of information. This fact is evident when removing the bridge or the long distance connection: we get a disconnected graph in one case and we have a high increase in the path lengths in the second case. Therefore, just measuring the

overlap of the two edge sets does not work in practice, but instead also differences between 2-step, 3-step, etc. neighbours should be considered. Starting from these considerations, Koutra, Vogelstein, and Faloutsos [48] proposed to take into account the effects of 2-step, 3-step, etc. paths by considering pairwise node affinities; in particular, they proposed to use the similarities given by Fast Belief Propagation [46]. This choice is due both to computational and theoretical reasons. Using this method, the pairwise node affinities s_{ij} for an undirected and unweighted network are are given by

$$\mathbf{S} = [s_{ij}] = [\mathbf{I} + \epsilon^2\mathbf{D} - \epsilon\mathbf{A}]^{-1}\,, \tag{1.3}$$

where $\mathbf{A}$ is the adjacency matrix, $\mathbf{D}$ is the matrix whose diagonal contains the degrees of the nodes, and ϵ is a small constant capturing the influence between neighbours. Notice that Equation (1.3) agrees with the intuition to compare 2-step, 3-step, etc. neighbourhoods: indeed, by neglecting the term $\epsilon^2\mathbf{D}$, the remaining expression can be expanded as

$$\mathbf{S} = [\mathbf{I} + \epsilon^2\mathbf{D} - \epsilon\mathbf{A}]^{-1} \cong [\mathbf{I} - \epsilon\mathbf{A}]^{-1} = \mathbf{I} + \epsilon\mathbf{A} + \epsilon^2\mathbf{A}^2 + \dots\,,$$

in which the term $\mathbf{A}^k$ encodes the possibility to reach node j from node i in k steps, and the coefficient ϵ^k, with $\epsilon < 1$, gives less similarity to more distant nodes.

Koutra, Vogelstein, and Faloutsos proposed two algorithms for the DeltaCon comparison method, one exact but quadratic in the size of the graphs and thus less efficient on large graphs, and the other approximated but linear in the number of edges. The exact version works simply as already described: all the pairwise node affinities $S_1 = [s^1_{ij}]$ and $S_2 = [s^2_{ij}]$ are computed using Equation (1.3) for both graphs, and then they are compared using the Matusita distance, which is the same as the Euclidean Distance but with rooted terms:

$$d = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\sqrt{s^1_{ij}} - \sqrt{s^2_{ij}}\right)^2}.$$

The distance is then converted into a similarity ranging in $[0, 1]$ by taking

$$sim_D = \frac{1}{1+d}\,.$$

The approximated version of DeltaCon relies on the fact that not all the pairwise node affinities are computed. The nodes are grouped randomly in g groups and the affinities are computed between each node and each group, so that the new affinity matrix $\mathbf{S}'$ has dimension $n \times g$, where n is the number of nodes. The algorithm proceeds as before, comparing the affinity matrices of the two graphs with the Matusita distance and getting the final similarity. The result obtained with the approximated algorithm is proved to always be an upper bound for the exact result.

Koutra, Vogelstein, and Faloutsos tested DeltaCon to show that it is a network distance with good, intuitive and reasonable properties. These properties are formulated in terms of how much the distance should quantify some particular changes that happen in a graph:

- changes that create disconnected components should be penalized more than changes that preserve the connectivity of the graph, and then some edges (like bridges) are more important than others;

- in weighted graphs, the bigger the weight the more important the edge, so that changes on edges with large weight should impact more;

- the same change should have more impact in graphs with few edges than in denser graphs;

- random changes should impact less than targeted changes of the same extent.

To test whether DeltaCon satisfies these properties, Koutra, Vogelstein, and Faloutsos considered small and simple graphs with classic topologies (like stars, clique, circles, paths, etc.) and they found that DeltaCon satisfied the aforementioned four properties. They also considered some other network distance measure already present in literature, like Edge Overlap [63], Graph Edit Distance [13] and the spectral distance, and they found that all of them failed in satisfying one or more properties. While a counterexample is enough to show that a measure does not satisfy a property, the claim that instead DeltaCon satisfies them should be carried out with more than ten comparison and with larger and more general graphs like the ones considered, but in a later work Koutra et al. [47] generalized and proved the first and the third property and provided stronger evidence for the other two.

DeltaCon was also tested on real graphs, with two different approaches. In the first test, DeltaCon was used to detect anomalies in a time-varying graph, in particular in a day by day who-emailed-who graph of a company. The data consist in the same set of nodes (i.e. with known node correspondence) with edges evolving over time; the nodes are all the employees of the company and edges are present between employees that emailed each other. The graphs are compared day by day. The results showed that DeltaCon was able to spot "anomalous" changes in the graphs, when the similarity between two consecutive days falls under a certain threshold. All the days spotted were related to crucial events in the company's history, like the change of the CEO or the announcement that the company overstated profits. In the second test, DeltaCon was used on a dataset of brain connectivity graphs for cluster analysis. They found two main clusters, and having other data available for each graph, they found also that the two groups mainly differed for some indices quantifying creativity and open-mindedness, meaning that brain connectivity is different in people that are creative and in people that are close-minded.

In their later work, Koutra et al. [47] also extended DeltaCon making it able to find which nodes or edges are responsible for the difference between the graphs compared. This is an important feature for a network distance since one can draw much more accurate conclusions on the reasons why the network changed and on the importance of certain edges or nodes for the connectivity of the graph. The new version of DeltaCon was tested in a similar way as explained before both on synthetic and real data, giving good results as well.

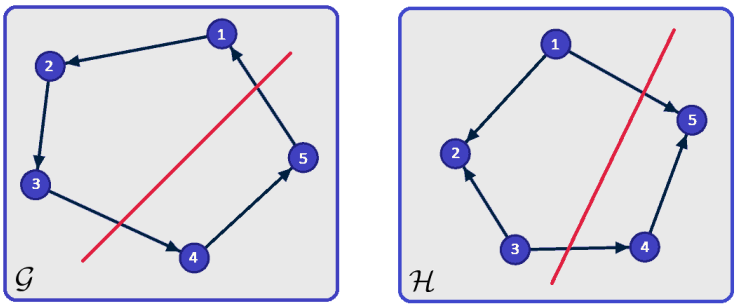


Figure 1.6: Comparison of two directed networks. The dot line represents the cut which separates the two sets $S = \{1, 2, 3\}$ and $S^C = \{4, 5\}$. For this particular cut, the cut weights are $e_{\mathcal{G}}(S, S^C) = 1$ and $e_{\mathcal{H}}(S, S^C) = 2$.

1.6.2 Cut distance

In a recent work, Liu, Dong, and Wang [54] presented a novel approach which is based on the *cut distance*. They were inspired by the notion of cut distance that is used in some community detection methods [32, 56]. The advantage of this approach is that it can be used to compare even directed and weighted graphs.

To define the cut distance, the notion of cut weight has to be introduced first. In a weighted directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with edge weights w_{ij}, $i, j \in \mathcal{V}$ and for each subset $S, T \subset \mathcal{V}$, the cut weight is defined as

$$e_{\mathcal{G}}(S, T) = \sum_{i \in S, j \in T} w_{ij}(\mathcal{G}) \,.$$

This is the total weight of the edges traversing the cut from S to T; in an unweighted graph, $e_{\mathcal{G}}(S, T)$ is equal to the number of edges with source in S and end in T. An example is provided in Figure 1.6.

Given two graphs $\mathcal{G}$ and $\mathcal{H}$ with the same node set $\mathcal{V}$ (that is, the correspondence between nodes must be known), the cut distance is simply defined as the maximum cut weight over all the possible splits of the vertex set:

$$d(\mathcal{G}, \mathcal{H}) = \max_{S \subset \mathcal{V}} \frac{1}{|\mathcal{V}|} |e_{\mathcal{G}}(S, S^C) - e_{\mathcal{H}}(S, S^C)| \,,$$

where $S^C = \mathcal{V} \setminus S$. The maximization is performed through genetic algorithms, so that the efficiency of this method is highly related to the efficiency of the genetic algorithm used. The main drawback with this choice is that the comparison of networks of more than thousands of nodes is computationally infeasible.

Tests on synthetic and real networks were performed to evaluate the accuracy of this method. The tests on synthetic networks were performed on three small sets of Erdős-Rényi graphs: the first composed of undirected unweighted networks, the second of directed unweighted networks and the third of directed weighted networks. In each set, the Erdős-Rényi networks were generated with different

number of nodes and density. In all the three cases, the cut distance was able to effectively cluster together graphs with similar size and density. No comparison using Barabási-Albert graphs is provided.

The experiments on real networks were performed using a small biological dataset of networks of chemical molecules, where each network was labelled with one out of two labels. Another test was performed using a dataset consisting of African elephant dominance networks, and again each network had one label (the national park where the elephant lived) out of two. In both cases, the cut distance was able to recover the underlying labelling with small error.

The cut distance method was also evaluated against other methods to show its superior accuracy. The methods chosen for the comparison are kernel-based methods [34, 70], which we do not consider in this work since they are known to have many drawbacks. Liu, Dong, and Wang decided to use a model selection process based on a 1-Nearest Neighbour classifier, which we already argued may not be a proper choice in some situations. Each network comparison method was tested on the accuracy it can reach by classifying each network as belonging to one of the three previously mentioned typologies of Erdős-Rényi graphs, and the cut distance method reached the best score. No evaluation against other methods already available in literature, like the ones we presented, was provided.

1.6.3 Portrait divergence

Bagrow and Bollt [6] recently introduced a new network distance called *Portrait Divergence*, which is based on the *network portrait* [5], a graph invariant. The idea underlying this method is to compare the distribution of all the shortest paths lengths in the two graphs and thus the measure is purely topological; moreover, it can be generalized to directed and weighted networks.

The network portrait is a matrix $\mathbf{B}$ defined as

$$B_{lk} = \text{the number of nodes who have } k \text{ nodes at distance } l\,,$$

for $0 \leq l \leq d$ and $0 \leq k \leq N-1$, where d is the network diameter and the distances are taken as the shortest path lengths; for weighted networks, path lengths are computed by summing the edge weights along a path, so that a binning strategy is needed to take into account the real-valued length and to proper define the portrait. The network portraits can be computed using, for instance, Breadth-First Search for unweighted graphs and Dijkstra's algorithm for weighted graphs. Some examples of portraits from different types of graphs are shown in Figure 1.7.

The network portrait encodes a lot of topological features of the graph. In particular, each row encodes a distribution. The first row stores the number of nodes as $B_{0k} = N\delta_{k,N}$. The second row stores the degree distribution $P(k)$: $B_{1k} = NP(k)$, since all the neighbours are at distance $l = 1$. The third row stores the distribution of the next-nearest neighbours, and so forth for all the other rows. Other topological features, such as the diameter or the number of edges as well as the number of the shortest paths of length l, can be recovered from the portrait. An important

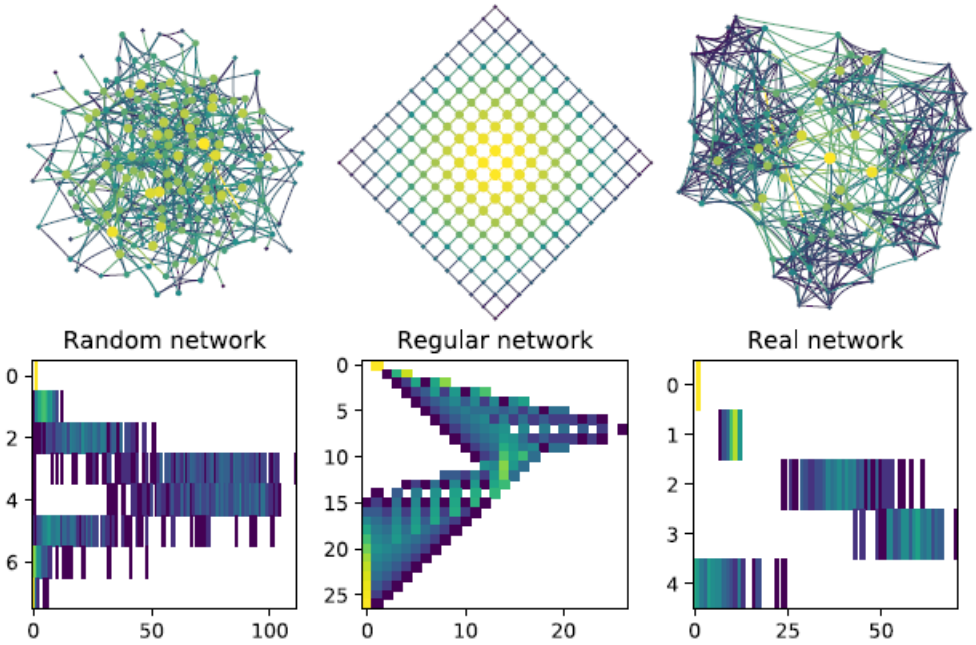


Figure 1.7: Example of network portraits [6]. The random network is an Erdős-Rényi graph while the real network is an American football network.

property of the portrait is that it is a graph invariant, that is, it is always the same for all the isomorphism of that graph. This fact makes the Portrait Divergence a measure that relies only on the structural topology of the graphs for the comparison. Moreover, as the definition of portrait already suggests, Portrait Divergence also is a network distance which does not need correspondence of the nodes to compare graphs.

The Portrait Divergence is defined from the portraits $\mathbf{B}$ and $\mathbf{B}'$ of two graphs as follows. First, the probability $P(k,l)$ (and $Q(k,l)$ for the second graph) of choosing two nodes at distance l and, for one of the two randomly chosen nodes, to have k nodes at that distance l, is defined as follows:

$$P(k,l) = P(k|l)P(l) = \frac{1}{N} B_{lk} \frac{1}{\sum_c n_c^2} \sum_{k'=0}^{N} k' B_{lk'} \,,$$

where n_c is the number of nodes in the connected component c ($\sum_c n_c = N$). Then, the Kullback-Liebler (KL) divergence is computed between the two distributions P and Q. In the end, the Portrait Divergence distance is defined using the Jensen-Shannon divergence, as follows:

$$D_{JS}(\mathcal{G},\mathcal{H}) = \frac{1}{2} KL(P||M) + \frac{1}{2} KL(Q||M) \,,$$

where $M = \frac{1}{2}(P+Q)$ is the mixture distribution of P and Q and $KL(\cdot||\cdot)$ is the KL divergence between the two distributions. With this definition, the Portrait Divergence takes values $0 \leq D_{JS} \leq 1$. The choice of the Jensen-Shannon divergence instead of the KL divergence is motivated by the fact that the former is symmetric and $\sqrt{D_{JS}}$ is a metric. The Portrait Divergence distance has many desirable

properties: as already mentioned, it does not require node correspondence; directed and undirected networks are considered naturally, and the extension to weighted networks is possible (it boils down to compute portraits defining binning intervals to aggregate the real-valued shortest path lengths); it is computationally efficient for small and medium size graphs, since it is quadratic in the number of nodes; it can handle disconnected networks without problems; all scales of the network's structure contribute simultaneously to the output distance.

The performances of the Portrait Divergence distance were evaluated both on synthetic and real networks. In the first case, two types of tests were performed. Firstly, the distances between different realizations of Erdős-Rényi and Barabási-Albert graphs were computed. It turned out that distances between graphs belonging to the same generative models are smaller than distances between graphs coming from different models. The method was then tested on how well it captures network perturbations. Two types of perturbations were performed: a random rewiring, meaning that an existing link is deleted uniformly at random and a new link is inserted uniformly at random, and a degree-preserving rewiring, meaning that a randomly chosen pair of links (i, j) and (u, v) is deleted and the pair (i, u) and (j, v) is added (the edges are chosen such that (i, u) and (j, v) does not already belong to the graph, ensuring the preservation of the degree distribution). To test the behaviour of Portrait Divergence on network perturbations, Bagrow and Bollt generated an instance of an Erdős-Rényi and of a Barabási-Albert graph and then applied iteratively the perturbations on a copy of that graphs, measuring after each perturbation the distance from the original graphs. As expected, it resulted that the random rewiring impacted more on the network's structures leading to larger distances. Moreover, the distances of the Barabási-Albert networks were overall larger than the ones of the Erdős-Rényi network, agreeing with the intuition that an Erdős-Rényi graph is already maximally random so that rewirings impact more on Barabási-Albert graphs.

Portrait Divergence was then used on real networks, specifically on two multilayer biological networks and on one weighted temporal (i.e. evolving) network. The first two datasets consist in a multilayer graph of the genetic and protein interactions in a vegetable species and in a multilayer graph of the nervous system of a nematode. Each layer was treated as a distinct graphs and comparisons between all the layers in each dataset were performed. In this case, the Portrait Divergence distance was able to highlight interesting aspects, which have biological and physical meanings, of the multilayer graphs. The last network analysed was the temporal network of software developer collaboration on a project on *GitHub.com*. It is a weighted network, where the edge weights represent the number of files edited in common by the two linked developers. Portrait Divergence was able to captures patterns in the temporal evolution of the network, in particular revealing a large similarity between two years in which the network growth had a significant slowdown.

1.6.4 Bayes modelling of a population of networks

The last method we present is not properly a network distance measure, though its characteristics can be exploited for the clustering of networks. Durante, Dunson, and Vogelstein [27] proposed a technique to model a population of networks using a Bayesian approach. We will explain in the following how this technique can be used in the task of network comparison.

Durante, Dunson, and Vogelstein [27] proposed a Bayesian nonparametric statistic framework to model a population of networks, which are seen as realizations of a network-valued random variable. The modelling is aimed at inferring the relevant parameters of that random variable: for instance, if we are given a collection of Erdős-Rényi graphs, we would like to say something about the probability p with which the edges were picked while generating the graphs. We now explain in more details the Bayesian model.

First of all, the model considers a collection of undirected and unweighted networks with the same vertex set $\mathcal{V}$ with cardinality N. Each observation $\mathcal{L}(\mathbf{A}_i)$ (i.e. network) is represented using a vector containing the lower triangular part of its adjacency matrix and it is considered to be a realization of the network-valued random variable $\mathcal{L}(\mathcal{A})$, which has binary entries $\mathcal{L}(\mathcal{A})_l \in \{0,1\}$ encoding the presence or absence of an edge between each pair of nodes $l = 1, 2, \dots, N(N-1)/2$, so that the random variable can represent all of the finitely many network configurations $\mathbf{a} \in \mathbb{A} = \{0,1\}^{N(N-1)/2}$. Then, the target of the inference is the probability mass function (pmf) $p_{\mathcal{L}(\mathcal{A})}(\mathbf{a}) = \mathbb{P}(\mathcal{L}(\mathcal{A}) = \mathbf{a})$. This pmf is assigned a mixture model and within each mixture component the edges are conditionally independent Bernoulli random variables given their component-specific edge probabilities, so that

$$p_{\mathcal{L}(\mathcal{A})}(\mathbf{a}) = \mathbb{P}(\mathcal{L}(\mathcal{A}) = \mathbf{a}) = \sum_{h=1}^{H} \nu_h \prod_{l=1}^{N(N-1)/2} \left(\pi_l^{(h)}\right)^{a_l} \left(1 - \pi_l^{(h)}\right)^{1-a_l}, \tag{1.4}$$

for every network configuration $\mathbf{a} \in \mathbb{A}$, where:

- $h = 1, 2, \dots, H$ are the indexes of the mixture components;
- $\nu_h \in (0,1)$ is the probability assigned to each mixture component h;
- $\pi_l^{(h)} \in (0,1)$ is the probability that an edge is observed for the l-th pair of nodes in mixture component h, for every $h = 1, 2, \dots, H$ and for every $l = 1, 2, \dots, N(N-1)/2$.

This is the core of the Bayesian model. However, the modelling continues with the use of a latent space to describe the edge probability vectors $\boldsymbol{\pi}^{(h)}$ with the aim of reducing dimensionality, but we refer directly to [27] for further details. The posterior distributions of the parameters of interest (ν_h and $\boldsymbol{\pi}^{(h)}$ for each $h = 1, 2, \dots, H$) are estimated using a suitable Gibbs Sampler algorithm.

We now describe how this Bayesian model can be used for the clustering of networks. Generating a network from model (1.4) first relies on sampling a component indicator variable $G_i \in \{1, 2, \dots, H\}$ with pmf defined by the mixing probabilities

$\nu_1, \nu_2, \ldots, \nu_H$, which are assigned a Dirichlet process prior. Then, conditionally on the sampled component and given the edge probability vector $\boldsymbol{\pi}^{(h)}$, the whole network is generated sampling its edges from conditionally independent Bernoulli random variables with probabilities $\pi_l^{(h)}$ for each $l = 1, 2, \ldots, N(N-1)/2$. This means that, if we often sample the same component $\bar{h}$, the generated network will be likely to be more similar to a network generated only from the edge probabilities of component $\bar{h}$. This is naturally enhanced by the choice of a Dirichlet process prior for the mixing probabilities $\nu_1, \nu_2, \ldots, \nu_H$: as a natural additional output, the posterior distribution of the indicator variables $G_1, G_2, \ldots, G_H$ provides a clustering among the networks. Moreover, the Dirichlet prior is the starting point for a wide variety of Bayesian methods for clustering data, so that the combination of model (1.4) with some Bayesian clustering methods is a promising direction for future improvements of network comparison methods.

In their work, Durante, Dunson, and Vogelstein used this method to infer the distribution of some network statistics (such as density, clustering coefficient and others) of a set of 100 synthetic networks with 20 nodes and of a set of 42 brain connectivity networks with 68 nodes. In both cases, their method outperformed in accuracy other existing methods in the approximation of the true network statistic distributions. Although this result is not directly related with the task of network comparison, having this additional information can give better comprehension of the results obtained while clustering networks. The main issue, due to the large dimensionality of the model representation, remains the scalability of the algorithm, which is suited only for small graphs.

Chapter 2

Analysis of synthetic networks

In Chapter 1 we gave an overview of the most relevant methods existing in the literature for the task of network comparison. In this Chapter, we present how we chose, used and evaluated some of those methods, to better understand their performances.

The goal of our work is twofold. On one hand, we want to perform a comparison of some of the methods we presented in Chapter 1. We do this mainly for two reasons: firstly, a comprehensive comparison of all the methods we presented is not present in the literature; secondly, we want to better understand how the methods behave in different situations, i.e. when comparing sets of graphs with different sizes, densities or topologies. Moreover, we are strongly interested in analysing and comparing methods which are able to make comparisons between directed and/or weighted networks. On the other hand, we want to use the methods to carry out an exhaustive analysis on two datasets of directed and weighted real-world networks, belonging to the field of economics.

To address these two goals, we proceed with the following steps.

1. We select some of the previously presented methods having in mind that they need to be applied to directed and/or weighted networks. If a method is only applicable to undirected and unweighted networks, we look for a possible extension.

2. We carry out some tests on synthetic networks, to understand the behaviour of each method with different network topologies, sizes and densities. Clustering tests are also carried out to evaluate how well each method can discriminate among different types of network topologies.

3. We finally analyse three real-world network datasets. The first one contains 37 undirected and unweighted transportation networks of European airlines [14, 25], while the other two consist of directed and weighted World Trade Networks of different products. The first dataset is from FAO [26, 31] and contains 364 trade networks of food, agricultural and animal products. The second dataset is extracted from the CEPII-BACI database [15, 65] and collects 1242 World Trade Networks of products belonging to various categories, from agricultural to high technology products.

2.1 Choice of the network distance measures

In our work, we decided to use only network distances that where already present in the literature and whose source codes or executables where freely available. We were also looking for distances which are able to compare directed and/or weighted networks or, if a distance is only defined for undirected and unweighted networks, we also considered distances which could be extended in some way to handle the directed and weighted case. We used these criteria to select, from the distances presented in Chapter 1, the ones which we wanted to analyse and use. We now list again all the previously presented methods, explaining which one we kept and why.

Methods based on global statistics. (Section 1.2) In this class of methods simple network statistics are used to assess similarity or dissimilarity of the two networks. We decided not to consider them, owing to their naiveté and their many issues.

Methods based on community analysis. (Section 1.3) Though the Onnela et al. [62] work was promising and the source code for the MRFs method was available, we preferred not to consider it because the World Trade Networks we want to analyse are known not to have a strong community structure [64].

MI-GRAAL. (Section 1.4.1) We considered this method since it is an alignment-based method, that is, it has the capacity to build a mapping between nodes. This can be an added value for the analysis of all kinds of graphs. Moreover, MI-GRAAL can be customized and made able to compare directed and/or weighted networks. The only potential problem is the actual computational time required to perform comparisons and the maximum size of the graphs that can handle. Whenever we used the MI-GRAAL distance, we set it to consider as nodes similarities the degree, the clustering coefficient and the betweenness centrality. The code of MI-GRAAL is available at http://www0.cs.ucl.ac.uk/staff/natasa/MI-GRAAL/index.html.

Graphlet-based measures. (Section 1.4.2) We were interested in selecting at least one of the alignment-free methods we presented, since they seem the most suitable ones to perform network comparison based only on the topological structure. A comprehensive analysis of the performances of the three methods on synthetic networks can be found in [77] and on real-world networks in [78]. As far as undirected and unweighted networks are concerned, we chose to consider the GCD-11 method for three reasons: it is shown to have the best performances in discriminating synthetic networks of different sizes and densities; it is computationally efficient; it allows a further analysis of the topological structure of the networks by comparing the Graphlet Correlation Matrices. We chose it even if in [78] it is shown to have a slightly worse, but still comparable, performance with respect to RGDF and GDDA in comparing real-world networks coming from different domains. Another reason to choose GCD-11 is related to the comparison of directed networks: in [69] it is shown that DGCD (the directed version of GCD) has the best performances when tested on synthetic networks. Even if DGCD-13

is faster, we decided to choose DGCD-129 as the graphlet method for directed networks because it considers also 4-node graphlets, which encode much more structural characteristics than 3-node graphlets and then are expected to capture finer topological differences between the graphs. The code for GCD-11 is available at http://www0.cs.ucl.ac.uk/staff/natasa/GCD/index.html, while the code for DGCD-129 is available at http://www0.cs.ucl.ac.uk/staff/natasa/DGCD/index.html.

NetDis. (Section 1.4.3) We did not consider this method, since it is shown in [78] that it has many issues and it is also shown to have worse performances, both in computational times and in accuracy, than GCD.

GRAFENE. (Section 1.4.4) We would have liked to consider this method since it is the most recent alignment-free available method based on graphlets, and it is shown to outperform GCD, RGFD and GDDA at least on synthtic networks [30]. Moreover, the method performs Principal Component Analysis and this can provide further informations. The executables are also available at https://www3.nd.edu/~cone/PSN/. The only problem is that the code is implemented to perform comparison between biological networks representing proteins (as the authors did in the paper), so that the executables accept a particular representation, called Contact Map Overlap, of the protein networks as input. We then discarded this method, since the code is strongly domain-dependent and providing as input the Contact Map Overlap representation of networks coming from other domains is meaningless.

Spectral methods. (Section 1.5) Despite the crucial drawbacks highlighted in [76], we decided to use spectral methods due to their straightforward implementation and interpretability. We define the spectral distance between two graphs as the Euclidean distance between the spectra of their representation matrices. Since in [76] the analysis of the performances in discriminating network classes using different matrix representation was poor, we decided to consider various versions of the spectral distance by considering different matrix representations of the graphs. In particular, we chose to use the adjacency matrix, the Laplacian matrix, and the Symmetric Normalized Laplacian. The first two representation matrices are suitable for the undirected case, both weighted and unweighted, since in this situation they are symmetric and therefore the eigenvalues are real. The Symmetric Normalized Laplacian can instead be used with undirected networks, both weighted and unweighted. Note that it can be applied also to directed networks, both weighted and unweighted, but in this case the definition of Symmetric Normalized Laplacian (see Equation (A.1)) considers the network as undirected, thus discarding the additional information given by the edge directions. No source code was available with [76], but it was straightforward to implement these distances on our own.

DeltaCon. (Section 1.6.1) We considered this method since it was shown to have many desirable properties for a network distance. An implementation that is able to handle directed graphs was also available, so that we decided to use and evaluate it. The source code can be found at http://web.eecs.umich.edu/~dkoutra/.

Cut distance. (Section 1.6.2) We would have liked to consider this method since it is able to naturally handle all kinds of networks, even directed and weighted, but neither the source code nor an executable is available. Moreover, we had big concerns about the computational efficiency of the method even when used on small graphs.

Portrait Divergence. (Section 1.6.3) We selected this method, because it is a very recent and able to deal with all types of networks. The source code is available at https://github.com/bagrow/portrait-divergence.

Bayes' modeling of a networks' population. (Section 1.6.4) We decided not to consider this method because it is not actually a network distance, since the clustering of network that it provides is only an additional output. Moreover, the Bayesian clustering model used in the method is the simplest possible. Nonetheless, we think that in the future this method can be a valuable tool for network comparison if provided with a more powerful Bayesian clustering model.

In addition to all these distances, we decided to consider also the simplest possible distances one can think between two networks, that is, matrix norms between the adjacency matrices of the graphs. This choice is motivated from the fact that, to the best of our knowledge, a norm between the adjacency matrices was never considered in the literature to be a possible way to compare networks. We then wanted to consider this possibility to evaluate if this simple distance is actually too simple and it is not suitable for network comparison. Thus, we considered four new network distances that are usual metrics among matrices. Given two networks $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$ with adjacency matrices $A_1 = [a^1_{ij}]$ and $A_2 = [a^2_{ij}]$ respectively and identical node sets $\mathcal{V} = \mathcal{V}_1 = \mathcal{V}_2$ (if $\mathcal{V}_1 \neq \mathcal{V}_2$, take $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ and pad with zeros the corresponding adjacency matrices), we define:

- **Euclidean distance:** it is the Euclidean norm between the adjacency matrices of the two networks:

$$d_{EUC}(\mathcal{G}_1, \mathcal{G}_2) = \sqrt{\sum_{i,j \in \mathcal{V}} \left(a^1_{ij} - a^2_{ij}\right)^2}\,.$$

- **Manhattan distance:** it is the Manhattan norm between the adjacency matrices of the two networks:

$$d_{MAN}(\mathcal{G}_1, \mathcal{G}_2) = \sum_{i,j \in \mathcal{V}} |a^1_{ij} - a^2_{ij}|\,.$$

- **Canberra distance:** it is the Canberra norm between the adjacency matrices of the two networks:

$$d_{CAN}(\mathcal{G}_1, \mathcal{G}_2) = \sum_{i,j \in \mathcal{V}} \frac{|a^1_{ij} - a^2_{ij}|}{|a^1_{ij}| + |a^2_{ij}|}\,,$$

  where we set $|a^1_{ij}| + |a^2_{ij}| = 1$ if $a^1_{ij} = a^2_{ij} = 0$.

- **Jaccard distance:** it is the Jaccard distance between the adjacency matrices of the two networks:

$$d_{JAC}(\mathcal{G}_1, \mathcal{G}_2) = 1 - J(A_1, A_2) = 1 - \frac{|A_1 \cap A_2|}{|A_1 \cup A_2|}\,, \tag{2.1}$$

where $J(A_1, A_2)$ is the Jaccard similarity. The intersection and the union in Equation (2.1) are meant elementwise. An alternative definition can be given by setting:

  - $p = |\left\{a_{ij}^1 = 1 \wedge a_{ij}^2 = 1\right\}|$ for $i, j \in \mathcal{V}$;
  - $q = |\left\{a_{ij}^1 = 1 \wedge a_{ij}^2 = 0\right\}|$ for $i, j \in \mathcal{V}$;
  - $r = |\left\{a_{ij}^1 = 0 \wedge a_{ij}^2 = 1\right\}|$ for $i, j \in \mathcal{V}$;
  - $s = |\left\{a_{ij}^1 = 0 \wedge a_{ij}^2 = 0\right\}|$ for $i, j \in \mathcal{V}$.

  Then, the Jaccard distance can be defined as

$$d_{JAC}(\mathcal{G}_1, \mathcal{G}_2) = \frac{q + r}{p + q + r}\,.$$

  A generalization of the Jaccard distance can be used to handle weighted networks, both directed and undirected, by considering the Weighted Jaccard similarity [17, 44]:

$$J_W(A_1, A_2) = \begin{cases} \dfrac{\sum_{i,j \in \mathcal{V}} \min(a_{ij}^1, a_{ij}^2)}{\sum_{i,j \in \mathcal{V}} \max(a_{ij}^1, a_{ij}^2)} & \text{if } \sum_{i,j \in \mathcal{V}} \max(a_{ij}^1, a_{ij}^2) > 0 \\ 1 & \text{if } \sum_{i,j \in \mathcal{V}} \max\left(a_{ij}^1, a_{ij}^2\right) = 0 \end{cases}$$

  Then, the Weighted Jaccard distance is defined as

$$d_{WJAC}(\mathcal{G}_1, \mathcal{G}_2) = 1 - J_W(A_1, A_2)\,.$$

Note that all these distances can be used with any kind of network, directed and undirected, weighted and unweighted (in the weighted case it is enough to consider, instead of the adjacency matrix, the weight matrix). Note also that, in the unweighted cases, since the elements of the adjacency matrices are binary, the Manhattan and the Canberra distances output the same result for the same pair of networks; the same happens with the Euclidean distance, which outputs the same result under square root. This means that, in the unweighted case, the three distances are actually the same, so that just one can be considered in the analyses.

The network distances we selected can be grouped into two different categories: distances which require a correspondence between the nodes of the two graphs to be compared, and distances which are independent on that correspondence. The formers are all those distances whose definition is based on the adjacency matrix or on some kind of node similarity, so that the graphs are required to have the same, or at least overlapping, node set; the latter are those which are based on some

Table 2.1: The classification of all the distances selected for our work

	Distances which require node correspondence	Distances independent on node correspondence
Undirected and Unweighted	Euclidean distance Manhattan distance Canberra distance Jaccard distance DeltaCon	Spectral Adjacency distance Spectral Laplacian distance Spectral SNL distance MI-GRAAL GCD-11 Portrait Divergence
Undirected and Weighted	Euclidean distance Manhattan distance Canberra distance Weighted Jaccard distance	Spectral Adjacency distance Spectral Laplacian distance Spectral SNL distance MI-GRAAL (extended) Portrait Divergence
Directed and Unweighted	Euclidean distance Manhattan distance Canberra distance Jaccard distance DeltaCon	MI-GRAAL (extended) DGCD-129 Portrait Divergence
Directed and Weighted	Euclidean distance Manhattan distance Canberra distance Weighted Jaccard distance	MI-GRAAL (extended) Portrait Divergence

topological feature which is not related to nodes labelling. To give an example, think for instance about the comparison of two networks in which the nodes represent countries: if we want to use the Euclidean distance, we must know which nodes have the same label in the two graphs to properly compare country by country, while if we want to use for instance GCD-11, we do not need to know the labels, since graphlets are geometrical structures independent from nodes labels. We want to stress the importance of this grouping, since the information given by the two types of distances has different interpretation. In the first case, the output distance will also contain information about the nodes importance in the networks: two graphs are similar if the same nodes have similar roles (where "roles" is related to the quantity on which the distance is defined) in both graphs. In the second case, the output distance will a be pure estimation of the topological similarity of the two graphs, completely disregarding the roles of the nodes.

In Table 2.1 we group all the distances we selected by type of graph on which they can be applied and by dividing them in these two categories, distances with requires node correspondence and distances which are independent on it.

After having selected the distances to use, we want to asses their behaviour on benchmark networks. We then set up three experimental frameworks that we describe in the following Sections, providing the related results.

2.2 Successive perturbation test

In this type of test, we performed successive perturbations starting from an original graph and we measured, at some fixed number of iterations, the distance of the obtained graph from the original one. This is aimed at checking if the distances behave "well", in the sense that the measured distance converges to zero when very similar graphs are compared (i.e., after few perturbations), that increases with the number of perturbations and that does not fluctuate too much (i.e., the method is reliable even if the graphs are perturbed). Moreover, we expect that the measured distance saturates to some threshold after having performed a large number of perturbations, because the perturbed network has become fully randomized and remains such for any further perturbation.

The types of perturbations that we perform on undirected and unweighted graphs are the following:

- **Removal:** an existing edge is picked uniformly at random and deleted from the graph.
- **Addition:** a new edge (between two nodes which are not already linked) is added uniformly at random in the graph.
- **Random switching:** an existing edge is picked uniformly at random from the graph and it is removed; after that, a new edge is added uniformly at random between two nodes which are not already linked.
- **Degree-preserving switching:** two edges are picked uniformly at random and swapped in the following way: if we picked the edges (i, j) and (u, v), we delete these edges and we insert the new edges (i, v) and (j, u). The original edges (i, j) and (u, v) can be selected only if the new edges (i, v) and (j, u) that will be inserted are not already present in the edge set, so that the degrees of the nodes do not change after the rewiring.

We applied the same types of perturbations to directed and unweighted networks, obviously taking care of the edge directionality, especially in the degree-preserving switching perturbation. For this kind of networks, we also considered one more type of perturbation:

- **Change of direction:** an existing edge is picked uniformly at random and its direction is reversed.

This perturbation is expected to affect only the network distances that can compare directed networks.

2.2.1 Specifications for the undirected and unweighted case

We wanted to investigate the effects of different network topologies and densities on the distances. We chose to keep the size of the original networks fixed at 1 000 nodes, not too large to increase computational time too much, but at the same time large enough to let the different network topologies arise. We considered densities of 0.01 and 0.05 for each one of three network models, namely Erdős-Rényi, Barabási-Albert and LFR with mixing coefficient $\mu = 0.2$, i.e. with a strong community structure (see Appendix A.3). Thus we have 6 different original networks. On each one of them, we performed 1 000 times the 4 perturbations for the undirected and unweighted case. Each perturbation was repeated 10 times to take into account the randomness in the choice of the edges to be added, deleted or swapped; in other words, it is like if we are generating 10 different "histories" of perturbations for each original graph. We chose 16 measurement points to measure the distance of the resulting networks after k perturbations from the original graphs. These 16 measurement points are chosen to be equally spaced in the logarithmic scale of the integer-valued interval $[0, 1\,000]$, so that we end up with 5 measurements for each order of magnitude of the number of perturbations, plus the final measurement at perturbation number 1 000. The measurements are obviously repeated for each of the 10 repetitions, so that we end up with a 16×10 matrix for each distance, whose rows are the distances at the measurement points and whose columns gather the different histories. In this experimental setting, we took into account the randomness in the choice of the edges to be added, deleted or rewired, but we did not take into account the randomness in the choice of the initial networks to be perturbed. This aspect should obviously be considered if one wants to perform a robust analysis, for instance by generating 10 initial networks for each network model and combination of parameters and then by taking the average of the resulting distances at the same measurement points. In this work, we reduced the analysis to 1 network for each model to reduce the computational requirements.

The distances that we considered for this test are all the distances in the first row of Table 2.1 except for the Canberra and the Manhattan distance: they behave exactly the same as the Euclidean distance apart from the square root. Moreover, we did not considered the MI-GRAAL distance when we analysed the graphs with 0.05 density, due to its high computational times (see Section 2.4).

The original graphs for the three undirected network models are generated with the parameters shown in Table 2.2. A few remarks: for the Barabási-Albert graphs, it was not possible to choose the exact number of resulting edges, so that we tuned the number of edges to add in each algorithm's step to get as close as possible to the required densities. The LFR networks were generated with the code from [51, 52]; also in this case, the code is not able to produce a network with the exact number of edges required, so that we run the code multiple times and we took the first network that had a number of edges close to the one required. How much "close" is specified by the *Edge tolerance* parameter reported in Table 2.2c, so that the number of edges of the generated network belongs to the interval $[\textit{Required edges} \pm \textit{Edge tolerance}]$.

Table 2.2: Parameters used to generate the networks (both undirected and directed) with 1 000 nodes used in the perturbation tests

	Undirected		**Directed**	
Density	**0.01**	**0.05**	**0.01**	**0.05**
Nodes	1 000	1 000	1 000	1 000
Edges	4 995	24 975	9 990	49 950

(a) Parameters for Erdős-Rényi graphs

	Undirected		**Directed**	
Density	**≈0.01**	**≈0.05**	**≈0.01**	**≈0.05**
Nodes	1 000	1 000	1 000	1 000
Edges added in each step	5	25	10	51
Resulting edges	4 985	24 675	9 945	49 674

(b) Parameters for Barabási-Albert graphs

	Undirected		**Directed**	
Density	**≈0.01**	**≈0.05**	**≈0.01**	**≈0.05**
Nodes	1 000	1 000	1 000	1 000
Exponent of degree distrib. (γ)	3	3	3	3
Exponent of community size distrib. (β)	1	1	1	1
Mean degree	10	50	10	50
Maximum degree	100	250	450	800
Mixing parameter (μ)	0.2	0.2	0.2	0.2
Minimum degree in community	5	5	5	5
Required edges	4 995	24 975	9 990	49 950
Edge tolerance	20	25	50	100

(c) Parameters for LFR graphs. For directed graphs, mean and maximum degree are intended as mean and maximum in-degree.

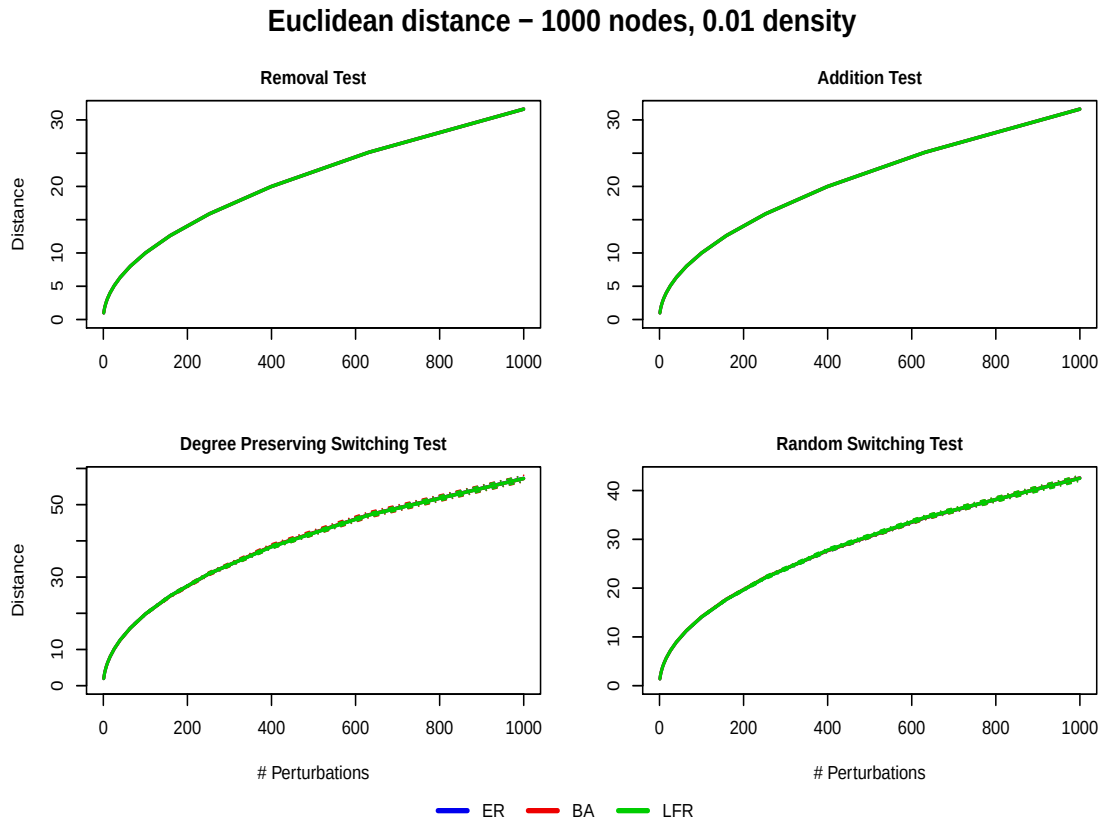
2.2.2 Results

We show the results in Figure 2.1 for the networks with 0.01 density and in Figure 2.2 for the networks with 0.05 density. Both Figures are organized in multiple plots: each plot shows the results related to a single distance and it is in turn divided in four panels, one for each type of perturbation. In each panel we gathered together the results for the three network topologies, to better compare them. The bold lines represent the average of the measured distances over the ten repetitions of each perturbation, whereas the dash-dotted lines represent the average measured distances plus and minus three times the standard deviation over the same ten repetition, so that almost 90% of the measured distances fall in that point-wise confidence band. Note that often the confidence band is so narrow to be practically invisible.

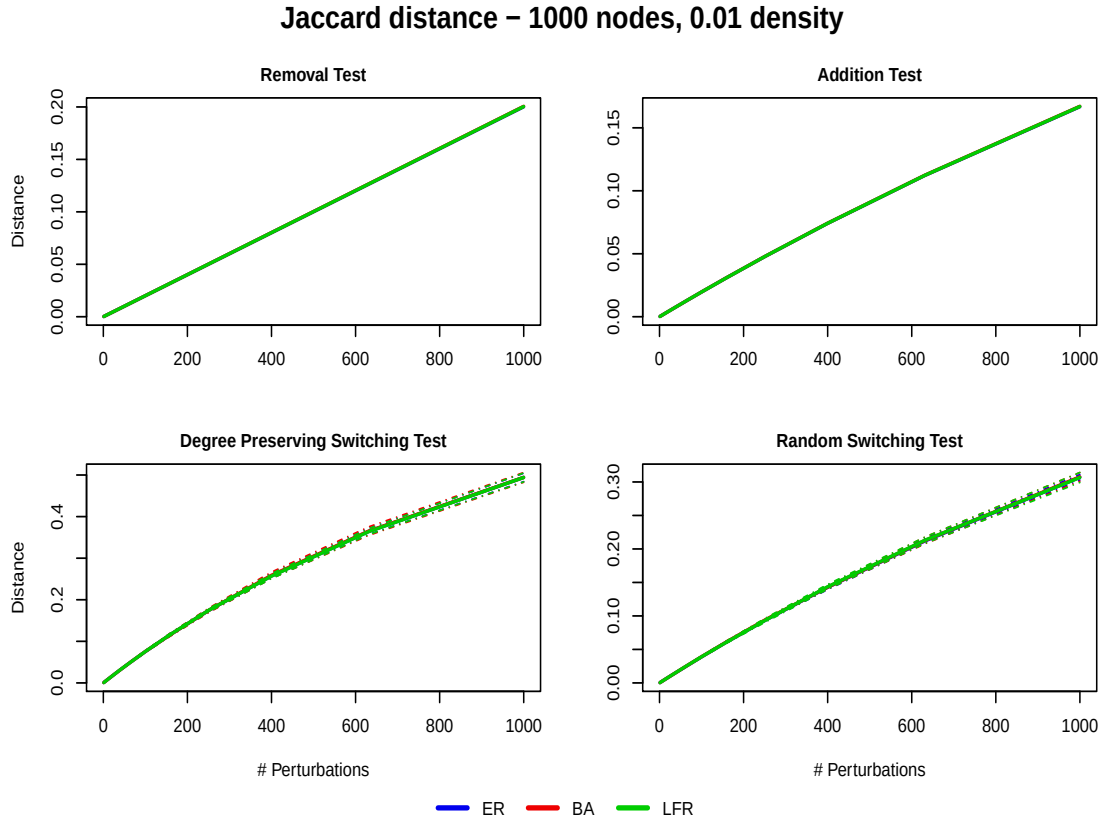
Results for networks with 0.01 density

We first analyse the results obtained on the synthetic networks with 1 000 nodes and 0.01 density (Figure 2.1). First of all, we notice that all the distances we considered behave "well", in the sense described at the beginning of Section 2.2: they all tend to zero when few perturbations are considered, they increase as the number of perturbations increases and they have tight confidence band, with the exceptions of GCD-11 and MI-GRAAL, which fluctuate more. Moreover, in most cases the distances present a saturation or a trend to saturate after many perturbations, as expected; this is especially evident for the degree-preserving and random switching tests, while this behaviour is not always present in the addition and removal tests. These results are promising since they prove that the methods we are considering can be used to do proper comparisons.

Another evidence that can be noticed by looking at the plots is the different behaviour of the two class of distances, those which require node correspondence and those that are independent on that. In particular, considering the first group, for each kind of perturbation the measured distances present the same trend, whit almost equal values for each network topology (see Figures 2.1a to 2.1c). Instead, in the second group, we notice a different trend across the different types of perturbations and of network topologies. This is an evidence that confirms that distances which require node correspondence are not aware of the network topology, but only of the nodes role, importance or connectivity: for those distances, a perturbation acts in the same way in all the considered network topologies. For instance, the Euclidean distance simply measures the square root of the number of edges that have been added, removed or switched, disregarding the topology in which these changes happen. Taking a more careful look, however, if we consider up to about the first 100 perturbations, we notice that also the network distances independent on node correspondence have similar measured distances for each perturbation and for each network model (except for the Spectral distances in the degree-preserving and random switching tests and for the MI-GRAAL distance in all the four perturbations, see Figures 2.1d to 2.1g).

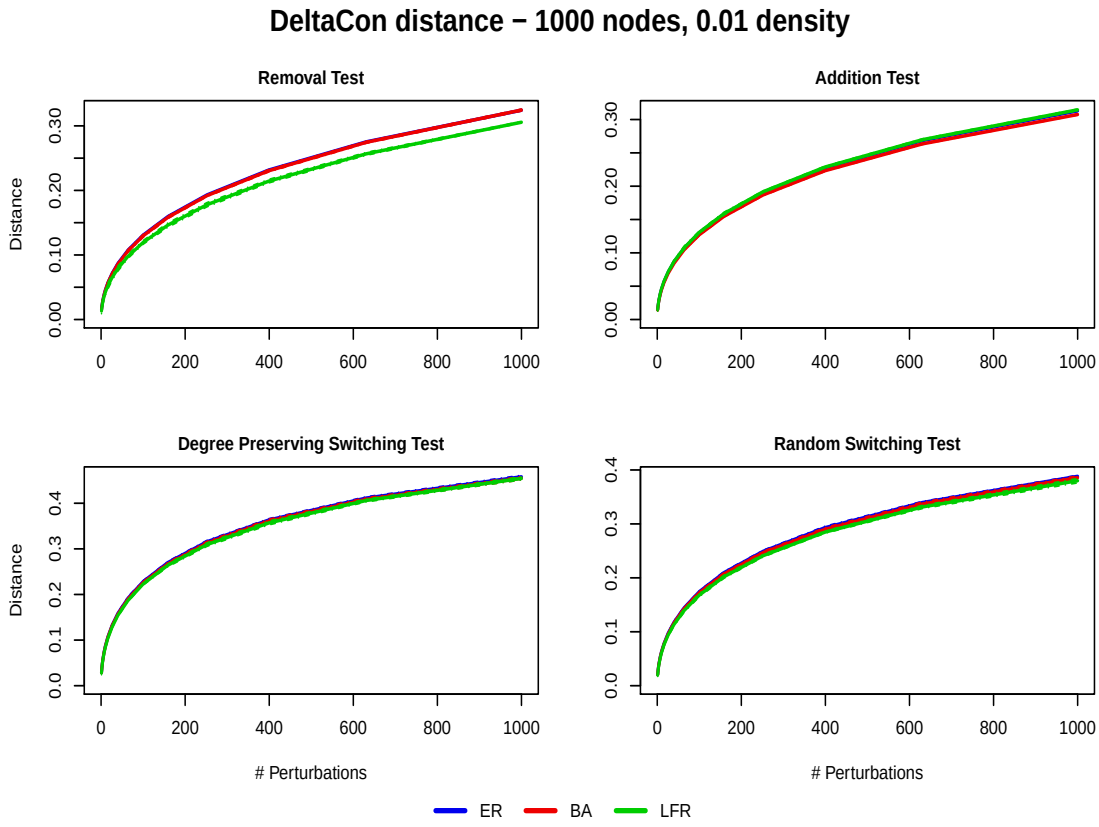


(a)

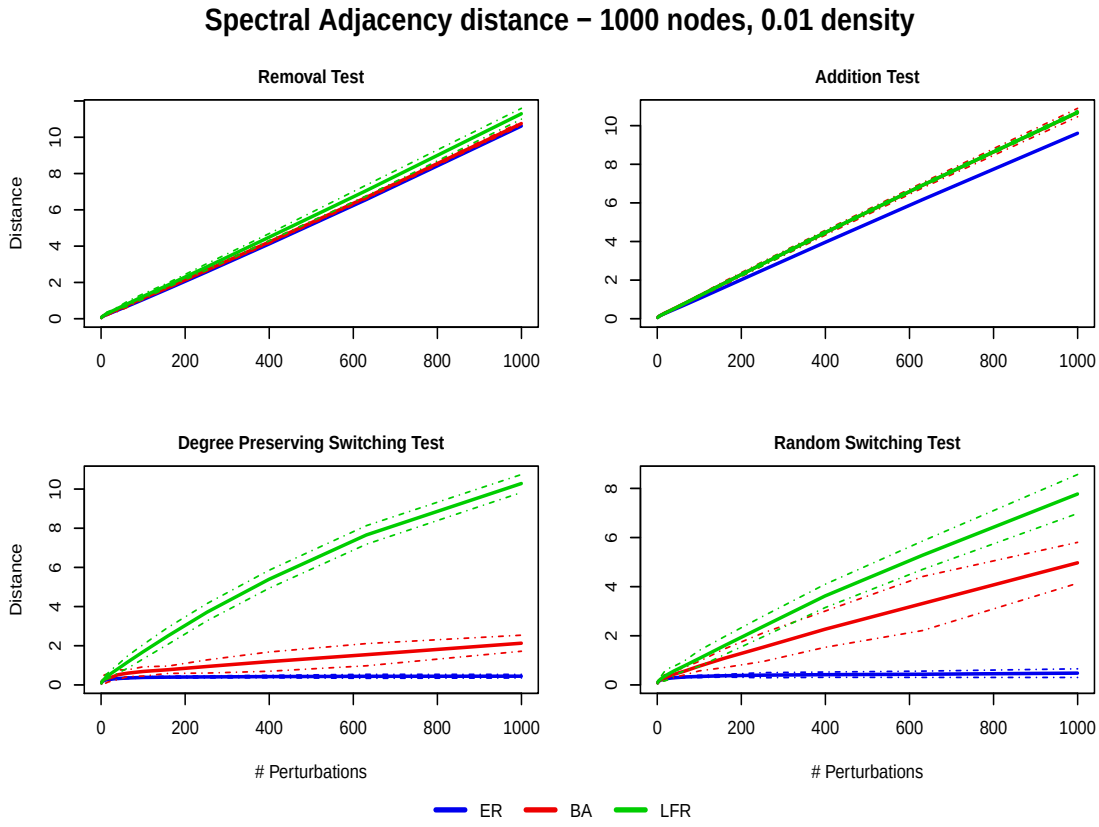


(b)

Figure 2.1: Results of the successive perturbations test on undirected and unweighted synthetic networks with 0.01 density.

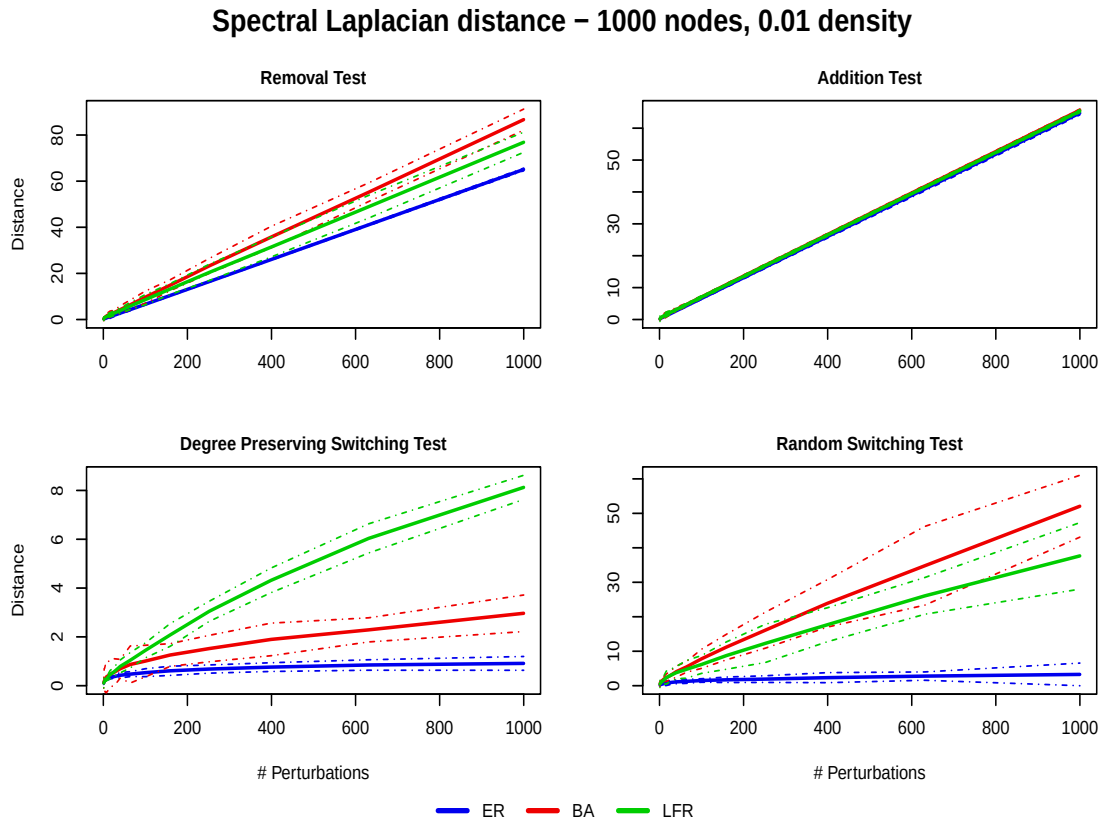


(c)

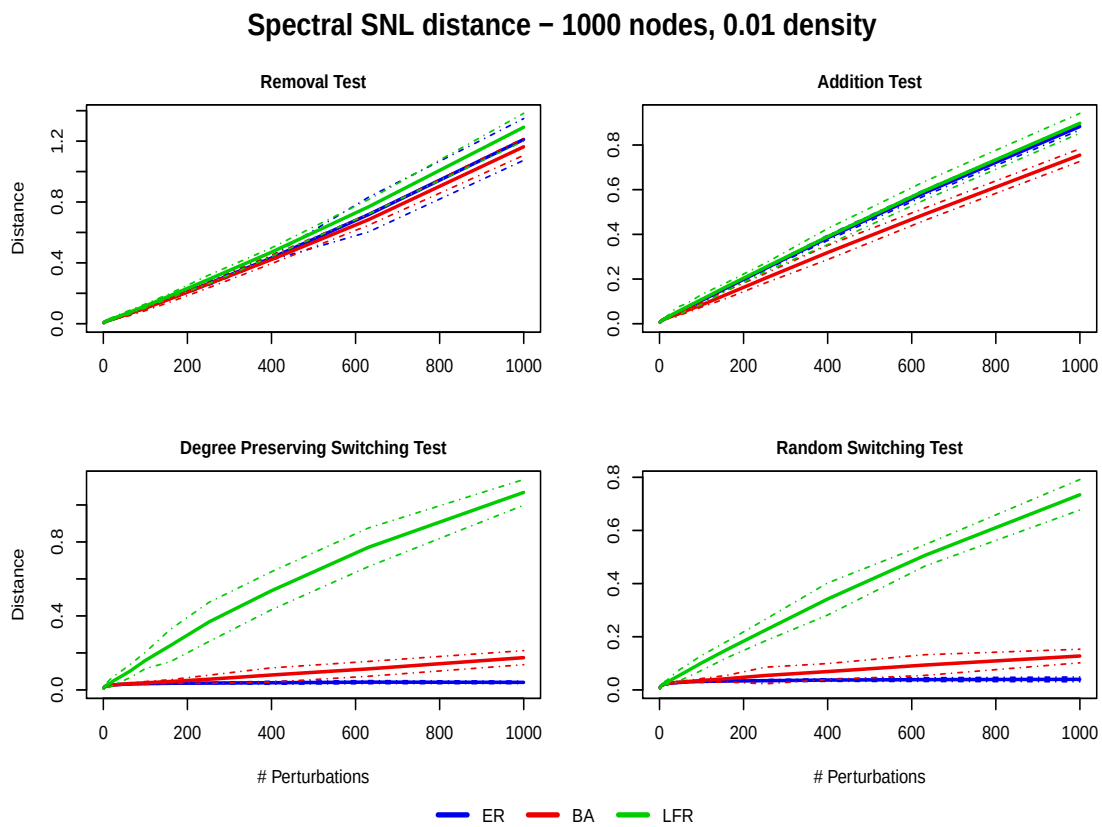


(d)

Figure 2.1 (cont.): Results of the successive perturbations test on undirected and unweighted synthetic networks with 0.01 density

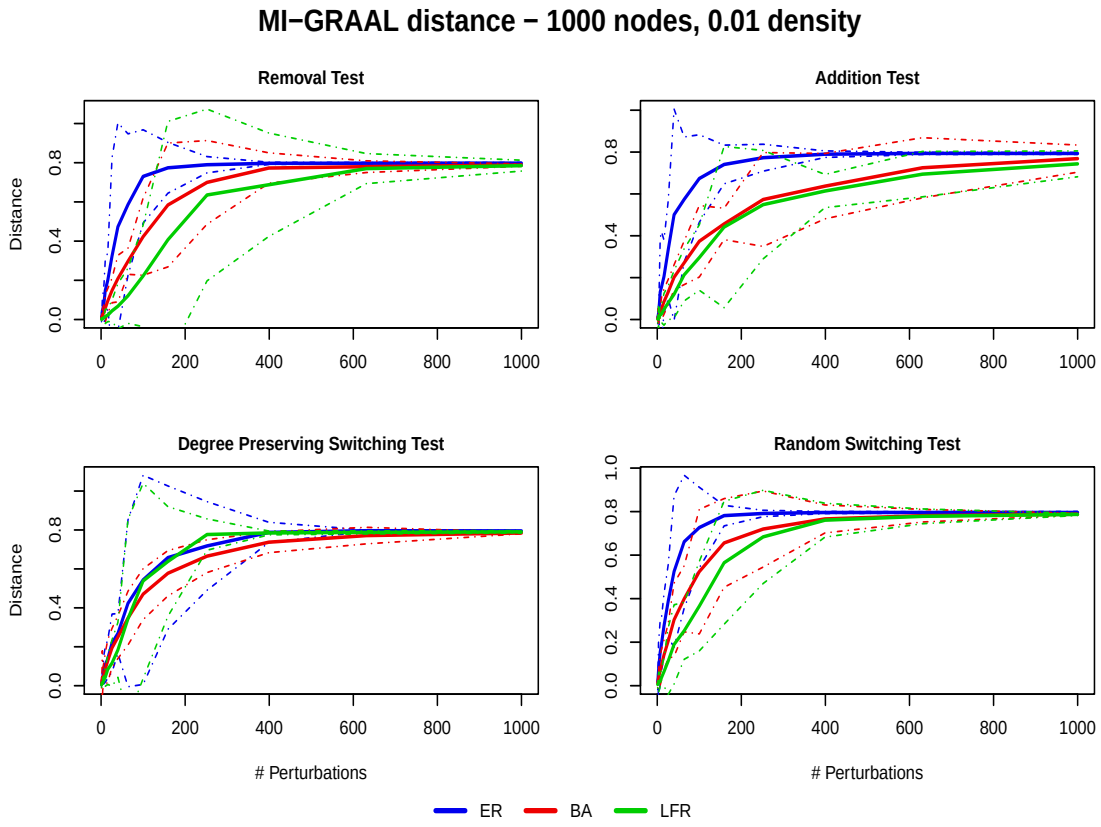


(e)

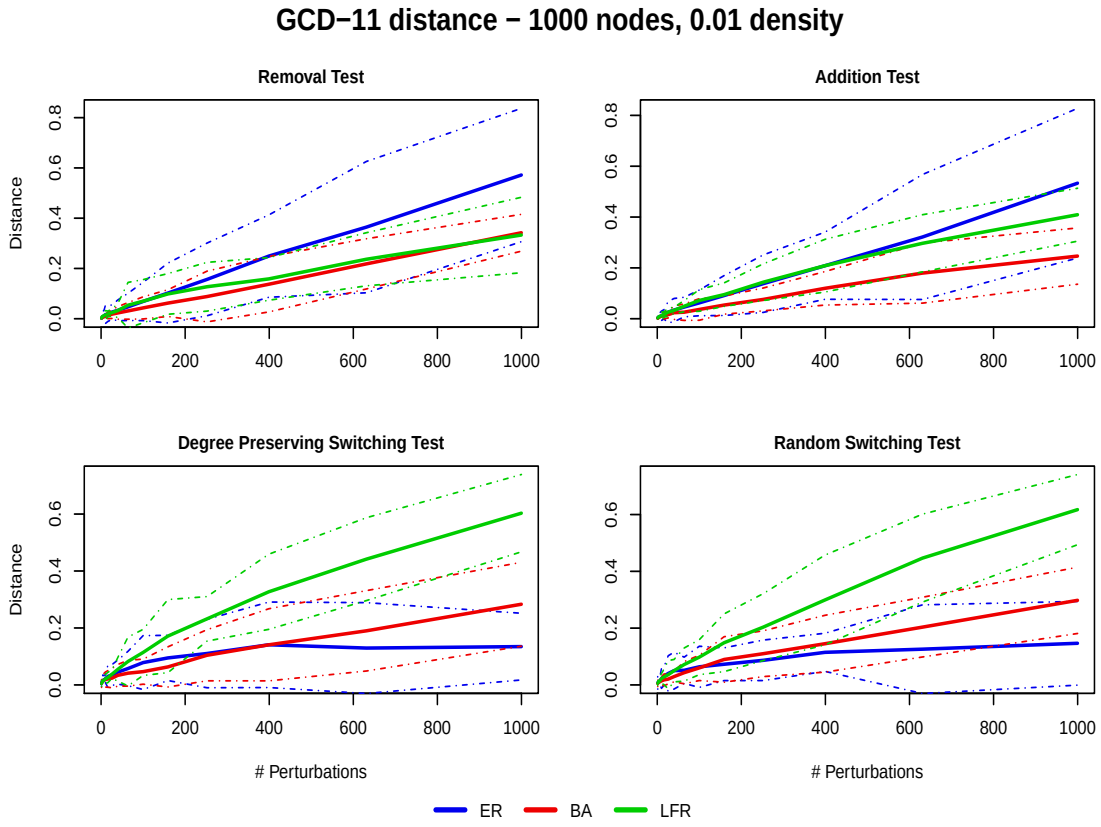


(f)

Figure 2.1 (cont.): Results of the successive perturbations test on undirected and unweighted synthetic networks with 0.01 density.

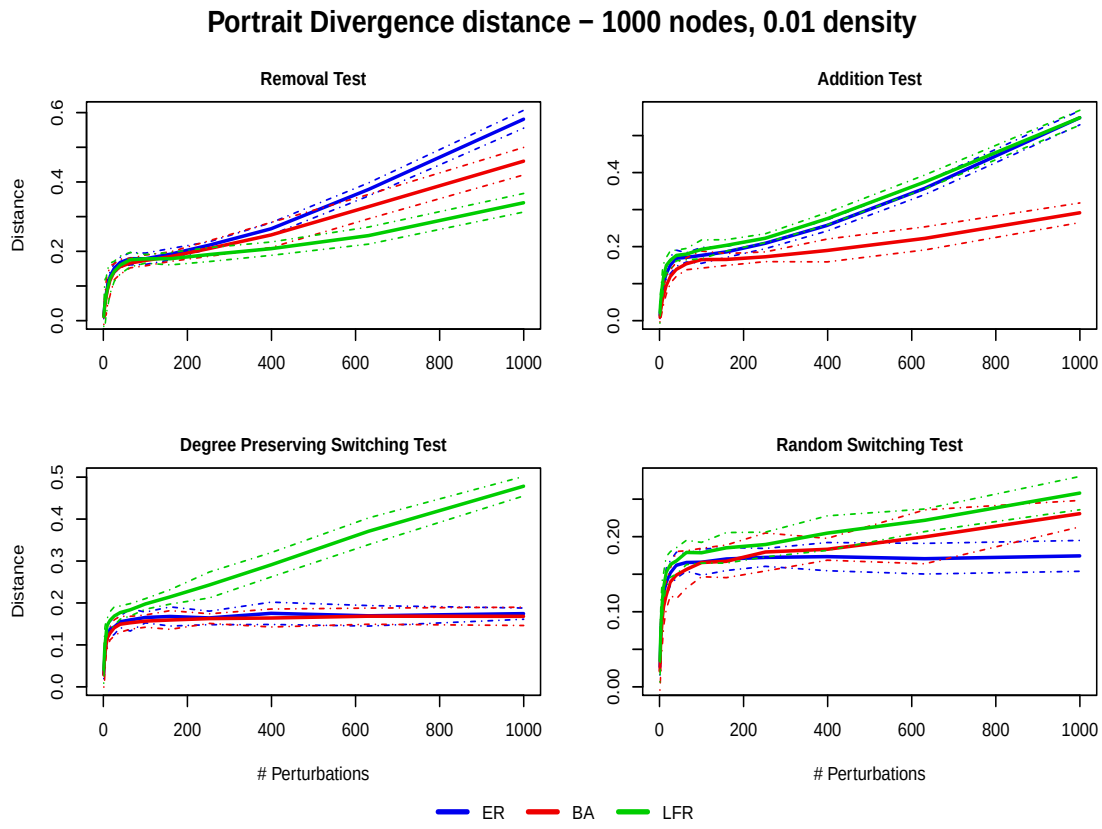


(g)



(h)

Figure 2.1 (cont.): Results of the successive perturbations test on undirected and unweighted synthetic networks with 0.01 density.



(i)

Figure 2.1 (cont.): Results of the successive perturbations test on undirected and unweighted synthetic networks with 0.01 density.

We now take a closer look to the distances which are independent on the node correspondence. For what concerns the three Spectral distances (Figures 2.1d to 2.1f), there is a different behaviour in the different kind of tests. In the addition and removal tests, the trend is linear and very similar for all the three topologies. Instead, in the degree preserving and random switching tests we see different behaviours for the different topologies. In particular, except for the random switching test relative to the Spectral Laplacian distance, the LFR curve always stays above the ER and the BA curve. This behaviour catches the higher fragility of LFR networks to random rewiring of edges with respect to the other topologies, meaning that the community structure in a LFR network is broken faster than the scale-free structure of Barabási-Albert graphs or the randomness of Erdős-Rényi graphs when edges are randomly rewired. This behaviour is expected, since, if the number of communities is large enough, degree preserving switching of edges more easily will create connection between two different communities, while the scale-free structure of a Barabási-Albert graph is preserved. Similarly, random switching of edges more easily will delete intra-communities edges and will create inter-community connections; with this perturbation also the scale-free structure is affected because hubs more easily will lose connections. On the other hand, Erdős-Rényi topology is the least affected by random rewiring of both types: an Erdős-Rényi graph is already random, so that random rewiring will produce of course different graphs, but structurally equivalent to the original one and thus, after some perturbations, all the perturbed graphs will have similar distances from the original one. Indeed, we can see that after around 200 perturbation the distance of the perturbed Erdős-Rényi graph from the original one does not change any more.

The same behaviour of the LFR curve, which stays above the ER and the BA one in the degree preserving and in the random switching tests, can be found also in the GCD-11 (Figure 2.1h) and in the Portrait Divergence (Figure 2.1i) distances. In particular, as we already mentioned, GCD-11 seems to be the network distance which has the higher variability, and this can be an issue for it. Instead, Portrait Divergence presents an interesting behaviour. It is the only distance which has a step in the first few perturbations, then either it goes immediately to saturation or it starts to increase again. The first behaviour is evident in the two switching tests, while the second in the addition and in the removal tests. In the degree preserving switching test, instead, we observe that the ER and the BA curve stay constant after the initial step. Again, the Erdős-Rényi topology under random rewiring produces equivalent graphs that are almost equally distant from the original one, but the interesting fact is that the same happens for the scale-free structure: we argue that after some degree preserving rewirings the shortest paths distribution is almost the same for all the perturbed graphs, so that the Portrait Divergence distance of the perturbed graphs from the original one keeps to be very similar.

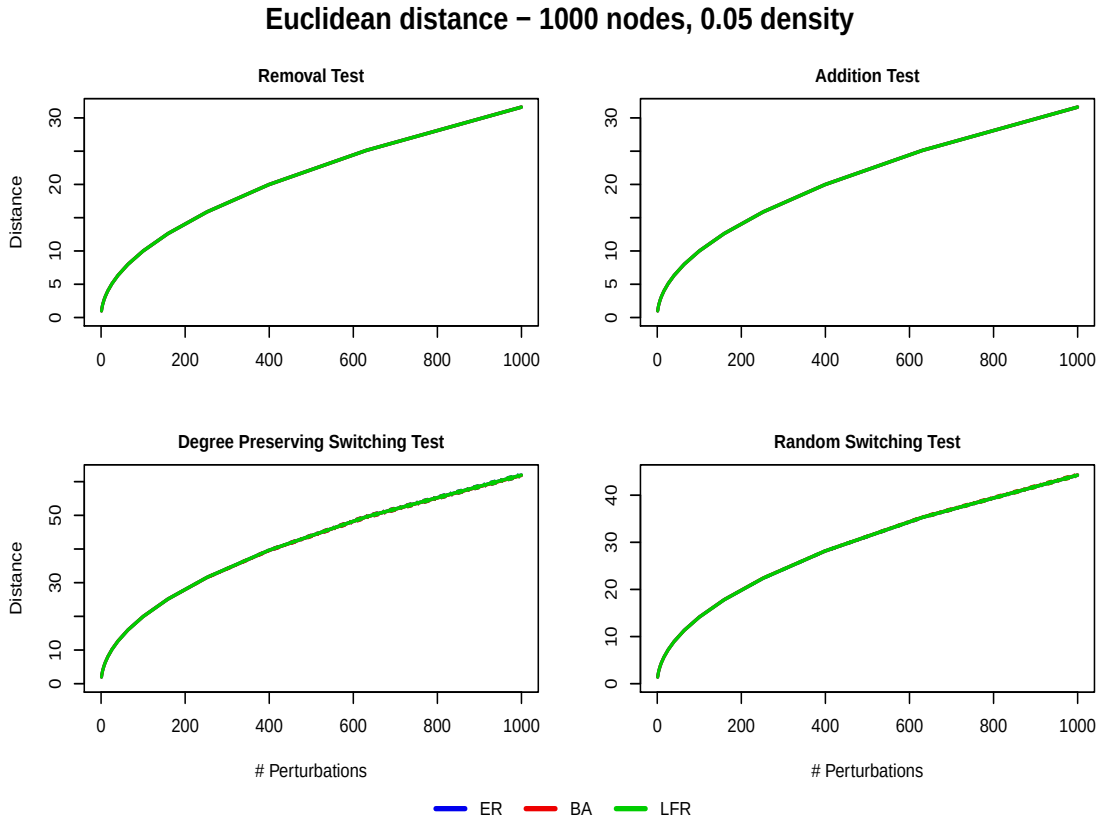
Finally, for the MI-GRAAL distance (Figure 2.1g) we observe a different behaviour from all the other distances. First of all, as already mentioned, it has large variability with respect to the choice of the perturbed edge. We can observe an extreme case in the removal test or in the degree-preserving switching test, where, for some values of the number of perturbations, the confidence band of the LFR

curve covers almost all the interval [0,1], which is the distance range. This means that a single perturbation can lead to a graph apparently equal to the original one or to a totally different one, depending on the edge chosen to be perturbed. This also happens for the ER curve in the addition test, and this is obviously a great issue for this method, while the BA always have the narrowest confidence band. Moreover, the curve which stays above the other two is not the LFR curve, which instead stays always below, but the ER curve. This could be due to the particular functioning of the MI-GRAAL distance, which builds a mapping between the most similar nodes of the compared networks. In the LFR and BA networks the nodes keep high similarity even after many perturbations: think for instance about the hubs of a BA graph, so that MI-GRAAL always maps hubs of the perturbed network with hubs of the original one. The same does not happen to the ER graphs, whose node similarities are much more affected by only a few perturbations: they can create small differences in the degree distribution, the clustering coefficient or in the betweenness centrality which can be enough to completely change the mapping. This happens for all the tests which do not preserve the degree distribution, while in the degree-preserving switching test we observe that all the curves stay closer to each other. The last interesting characteristic of MI-GRAAL is that in all tests the three curves reach their plateau at the same threshold.

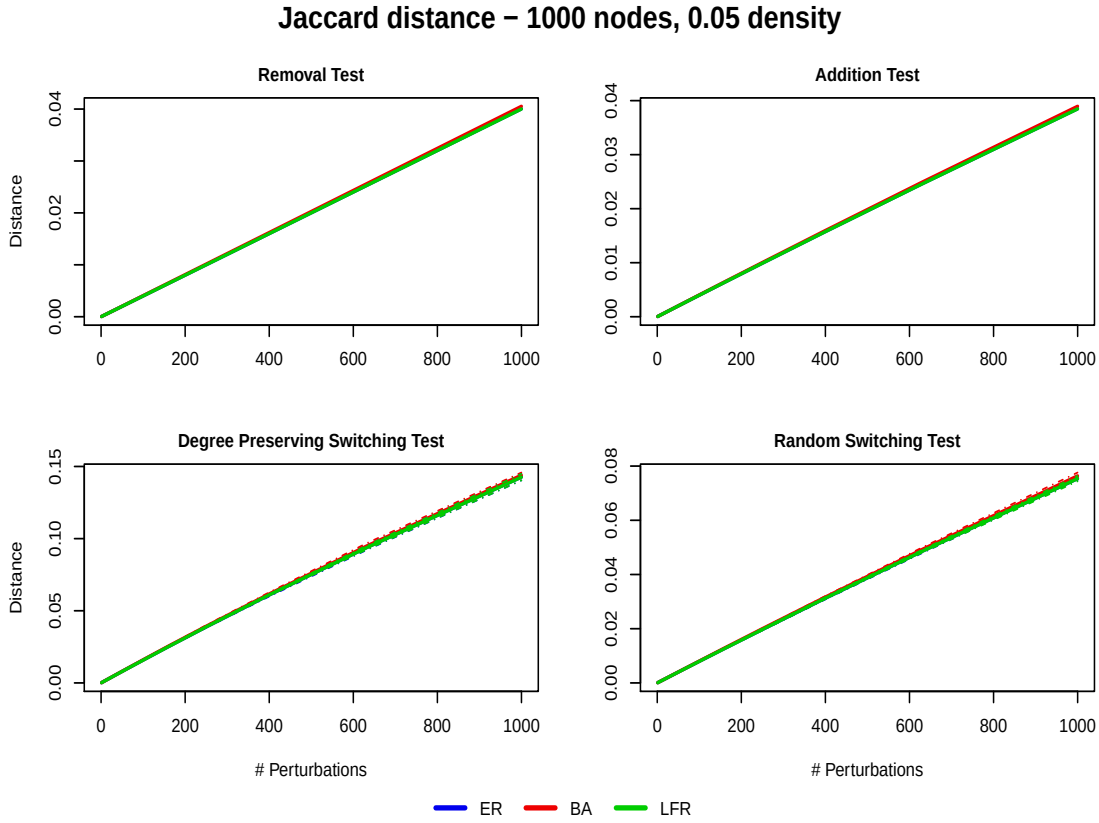
Results for networks with 0.05 density

The plots, shown in Figure 2.2 and related to the undirected and unweighted networks with 1 000 nodes and 0.05 density, essentially show the same behaviours that we mentioned for the networks with 0.01 density. We recognize the different behaviour of the distances which requires node correspondence, which again are not aware of the network topologies, with respect to those independent on that. For what concerns the latter methods, we see again that the LFR curve stays above the ER and the BA ones, with the only exception of the Spectral Laplacian distance (Figure 2.2e) in the random switching test, as in the 0.01 density case (Figure 2.1e). GCD-11 (Figure 2.2g) presents again the highest variability among the distances which are independent on node correspondence. Portrait Divergence (Figure 2.2h) is the only method that shows some qualitative differences with respect to the 0.01 density case. In particular, we see that the initial step is not as sharp as before; another difference is that almost all curves, in the four tests, go immediately to saturation after the initial growth, with the only exception of the ER curve in the addition and removal tests and of the LFR curve in the degree-preserving switching test. About the differences with respect to the 0.01 density case, we notice that in almost all methods, except to the Euclidean and the Spectral Laplacian distances, there is a decrease in the value of the measured distance for the same fixed number of perturbations. This shift reflects the presence of more edges in the graphs, so that graphs are more robust to perturbations.

We observed that, in the perturbation tests on undirected and unweighted graphs, the behaviour of the methods is not much influenced by the different densities of the original graphs. Notice, however, that this is not an exhaustive analysis, since we only analysed two values of edge density.

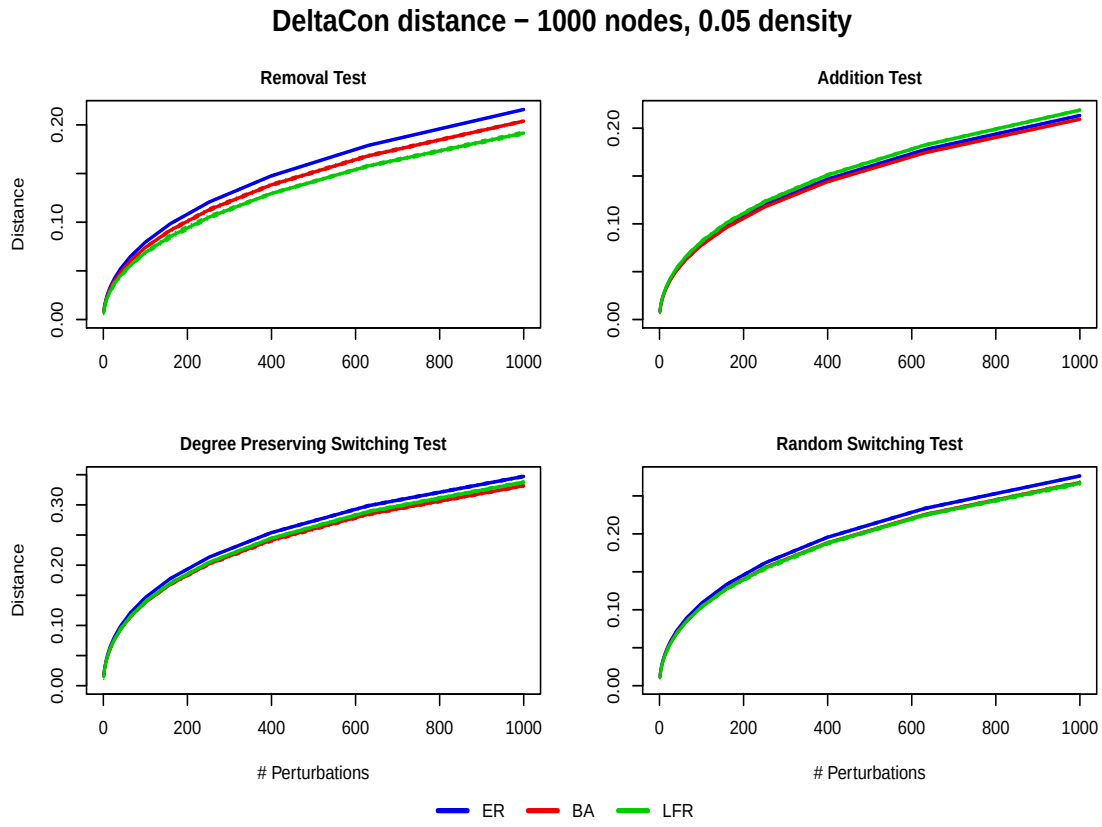


(a)

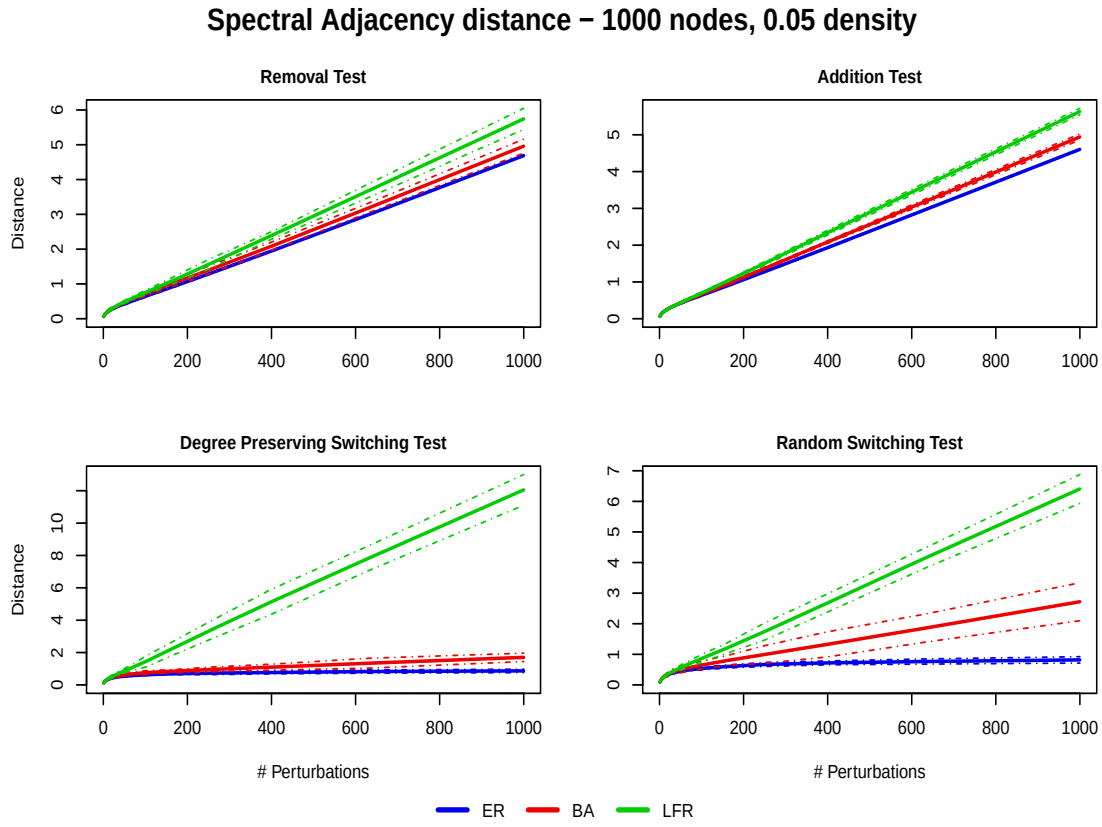


(b)

Figure 2.2: Results of the successive perturbations test on undirected and unweighted synthetic networks with 0.05 density.

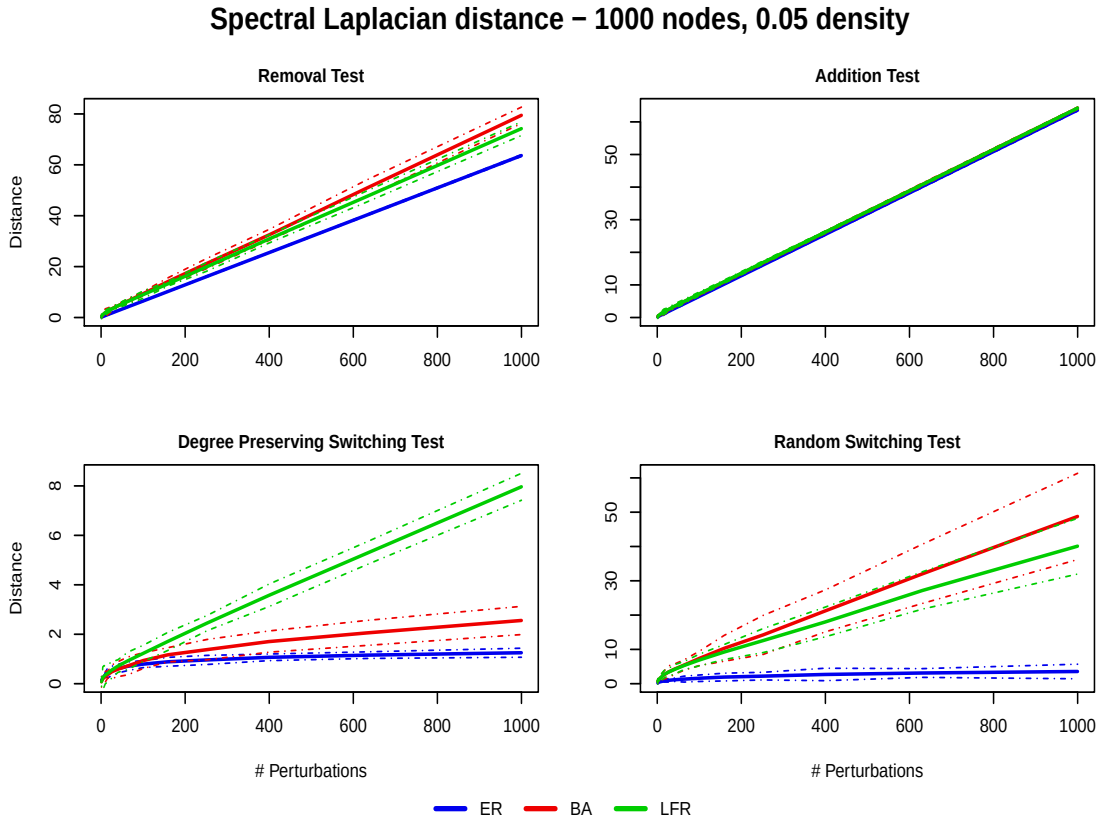


(c)

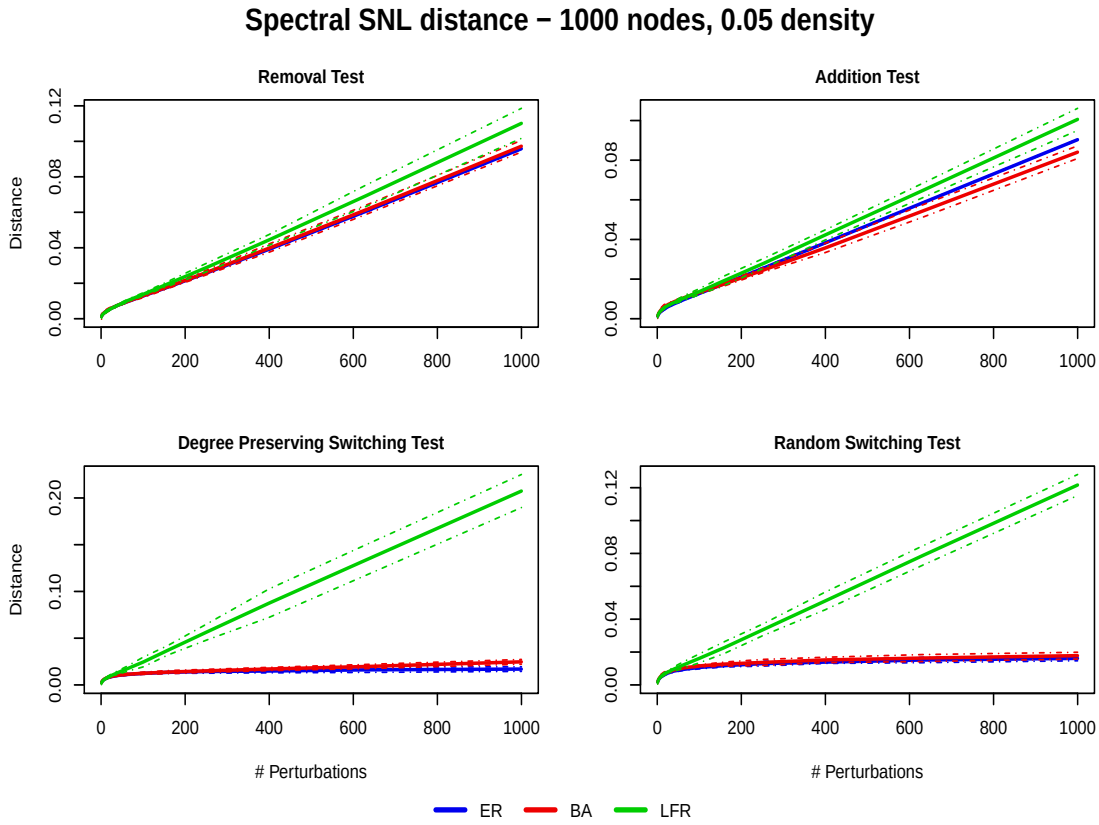


(d)

Figure 2.2 (cont.): Results of the successive perturbations test on undirected and unweighted synthetic networks with 0.05 density.

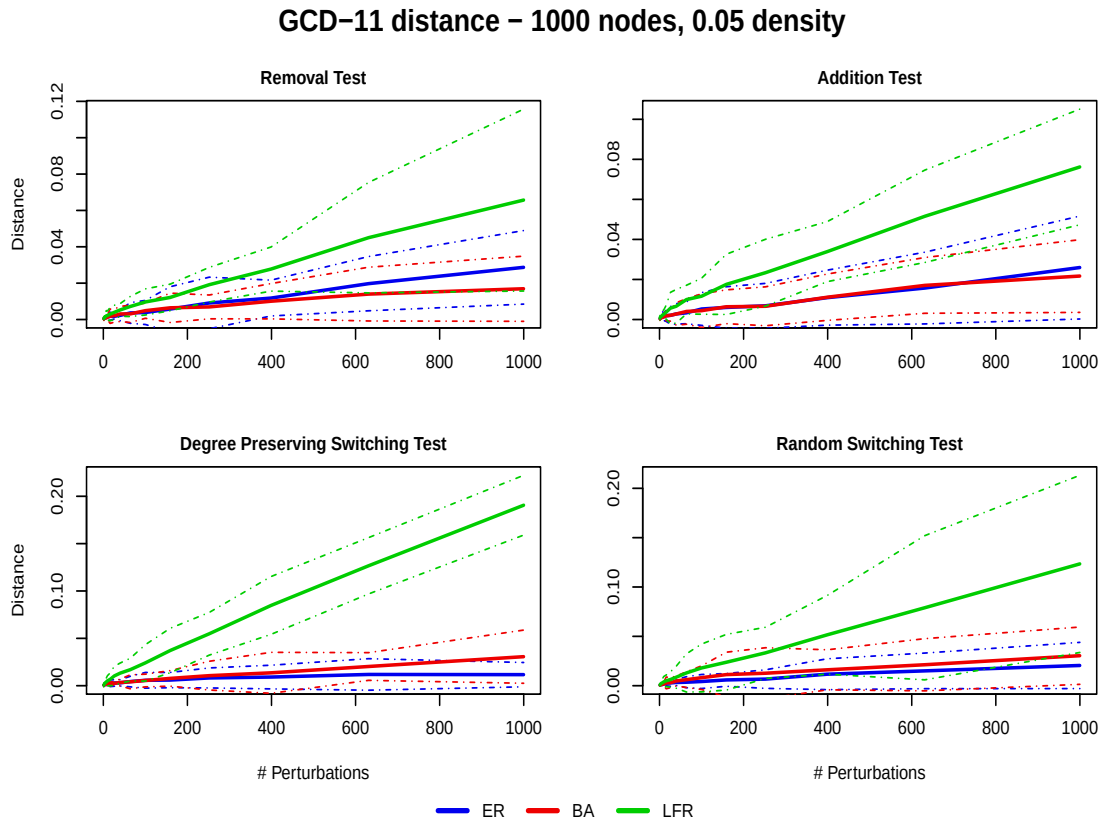


(e)

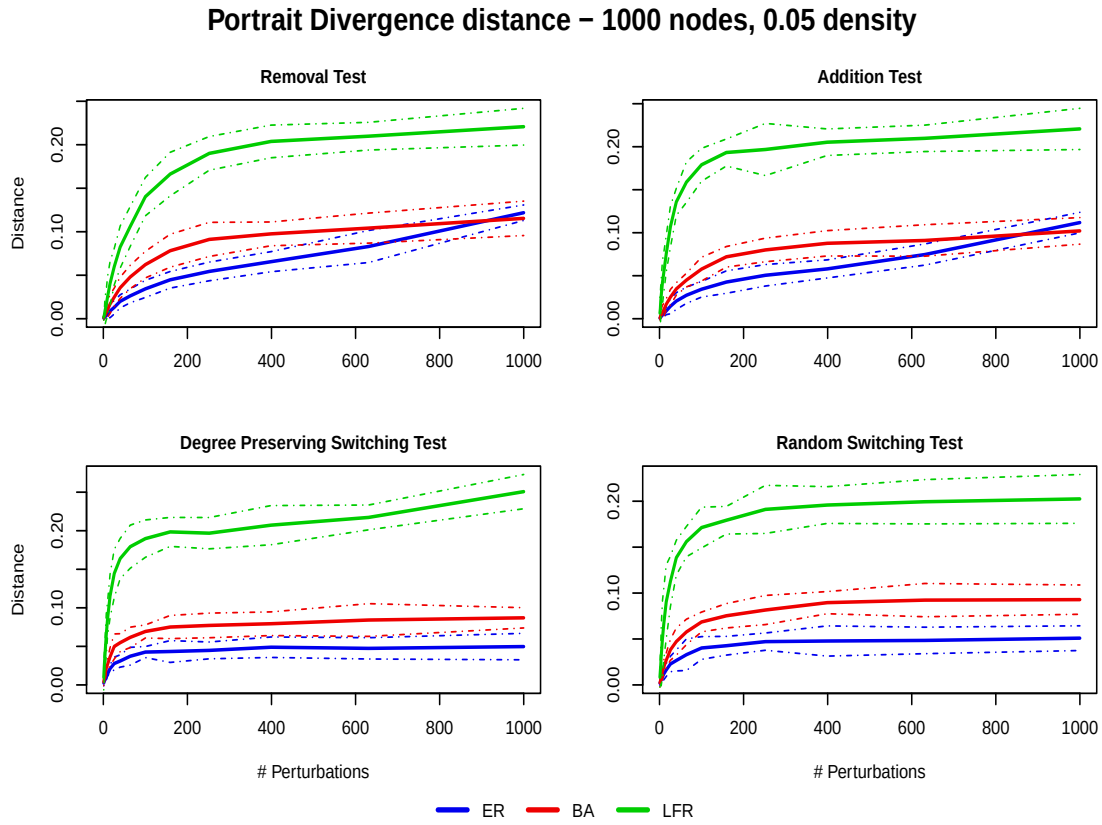


(f)

Figure 2.2 (cont.): Results of the successive perturbations test on undirected and unweighted synthetic networks with 0.05 density.



(g)



(h)

Figure 2.2 (cont.): Results of the successive perturbations test on undirected and unweighted synthetic networks with 0.05 density.

2.2.3 Specifications for the directed and unweighted case

The setting for the perturbation test on directed and unweighted networks is almost the same as for the undirected and unweighted case. In particular, we considered exactly the same networks models, with the same size and the same densities as before, only generating directed networks (see below). We considered the same number of perturbations, the same number of repetitions ("histories") for each perturbation, the same measurement points and the same perturbations of the previous case along with the change of direction perturbation. For this last perturbation, we also computed the undirected versions of the perturbed graphs and we compared it to the undirected versions of the original graphs to check whether the distances able to handle directed graphs actually exploit the information given by the edge directions. We used GCD-15, which takes into account all the 4-node undirected graphlets, as the undirected counterpart of the DGCD-129 distance.

We considered the distances suitable for the directed and unweighted case, which are reported in the third row of Table 2.1. Since we are again in the unweighted case, we did not consider the Manhattan and Canberra distances. Moreover, we did not considered any of the possible extended versions of MI-GRAAL, due to its too high computational cost (see Section 2.4). Then, we carried out the analysis with five distances.

The parameters used to generate the original networks are reported in Table 2.2. The directionality of the edges in the different network models is chosen as follows:

- **Erdős-Rényi:** once an edge is picked, its directionality is drawn at random with probability 0.5;
- **Barabási-Albert:** at each time step, a new vertex with a fixed number of out-links is added, each link pointing to an already existing node with probability proportional to the in-degree of that node. This results in a Barabási-Albert network in which the hubs have high in-degree.
- **LFR:** see Appendix A.3.3.

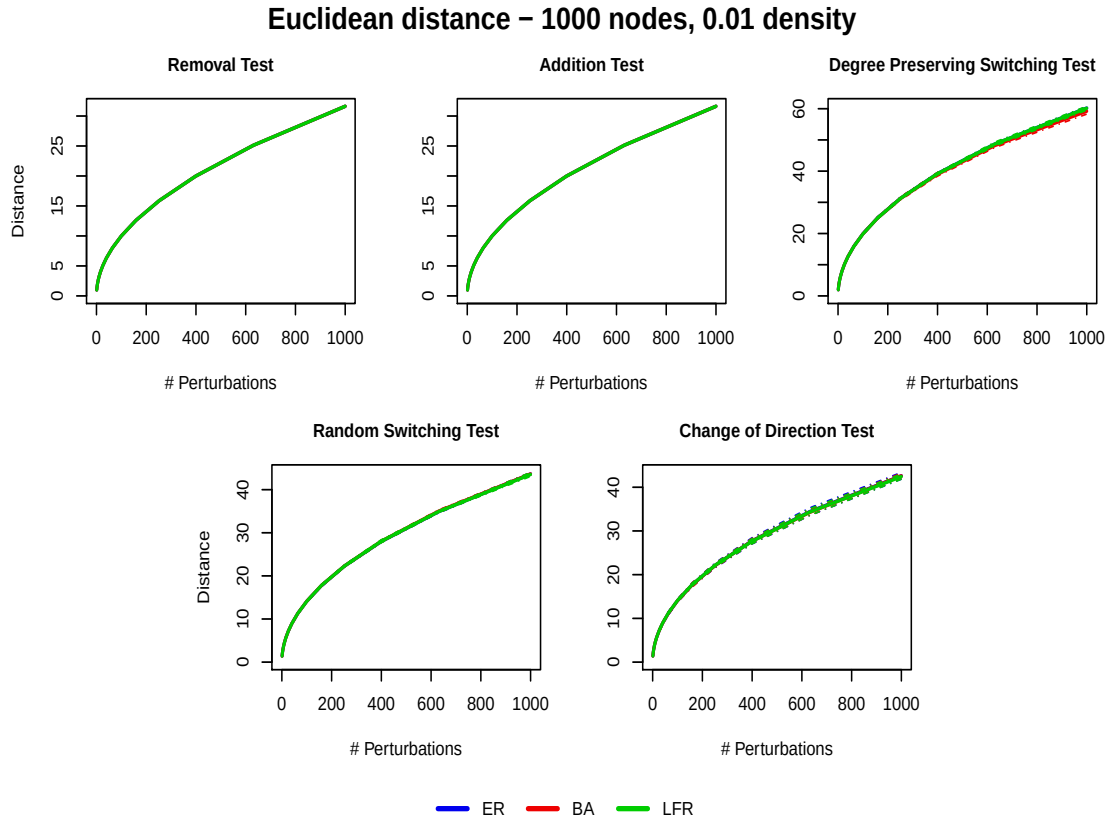
In this work we will also use network distances to compare directed and weighted graphs, but we did not carry out any test on synthetic networks using them. Although this is obviously needed to have an exhaustive analysis, it is difficult to design suitable experiments. The main difficulties are the definition of proper weighted benchmark networks and their perturbations.

2.2.4 Results

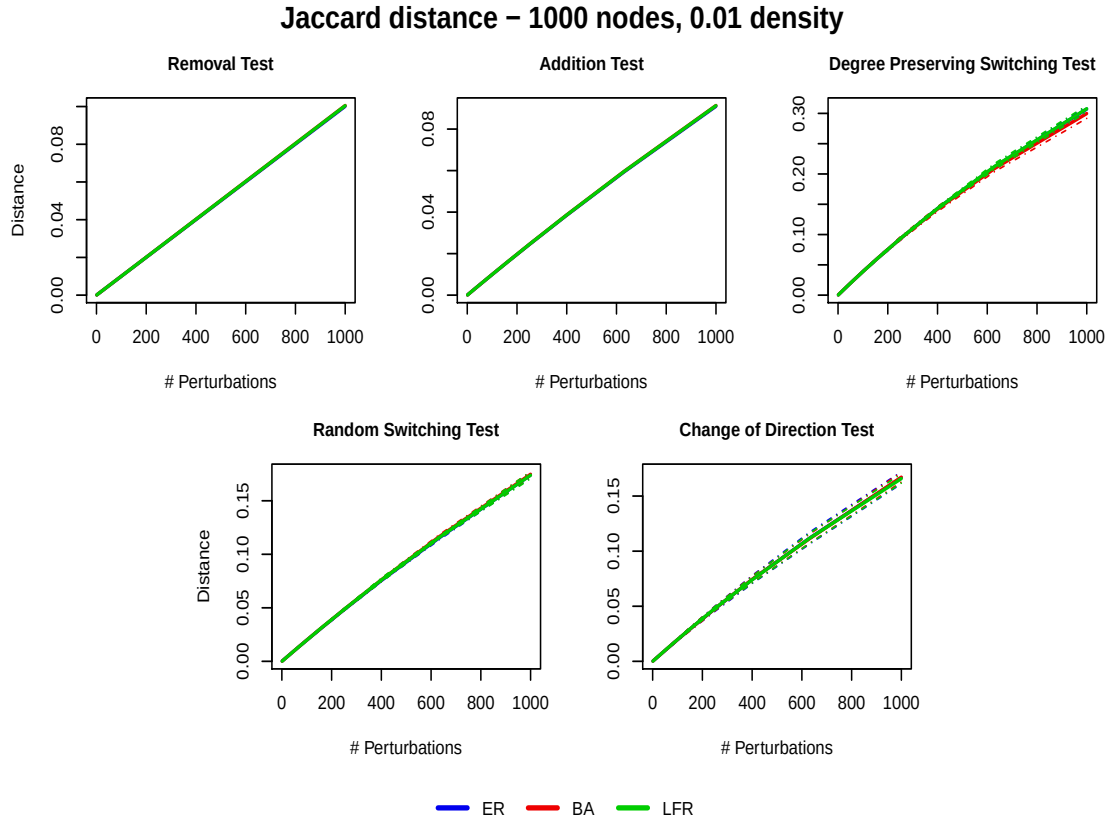
We show the results for networks with 0.01 edge density in Figure 2.3 and for networks with 0.05 density in Figure 2.4. The Figures are organised in the same way as those related to the perturbation tests on undirected and unweighted networks.

Results for networks with 0.01 density

We observe that all the distances which require node correspondence behave "well" in this context too, in the sense discussed at the beginning of Section 2.2.

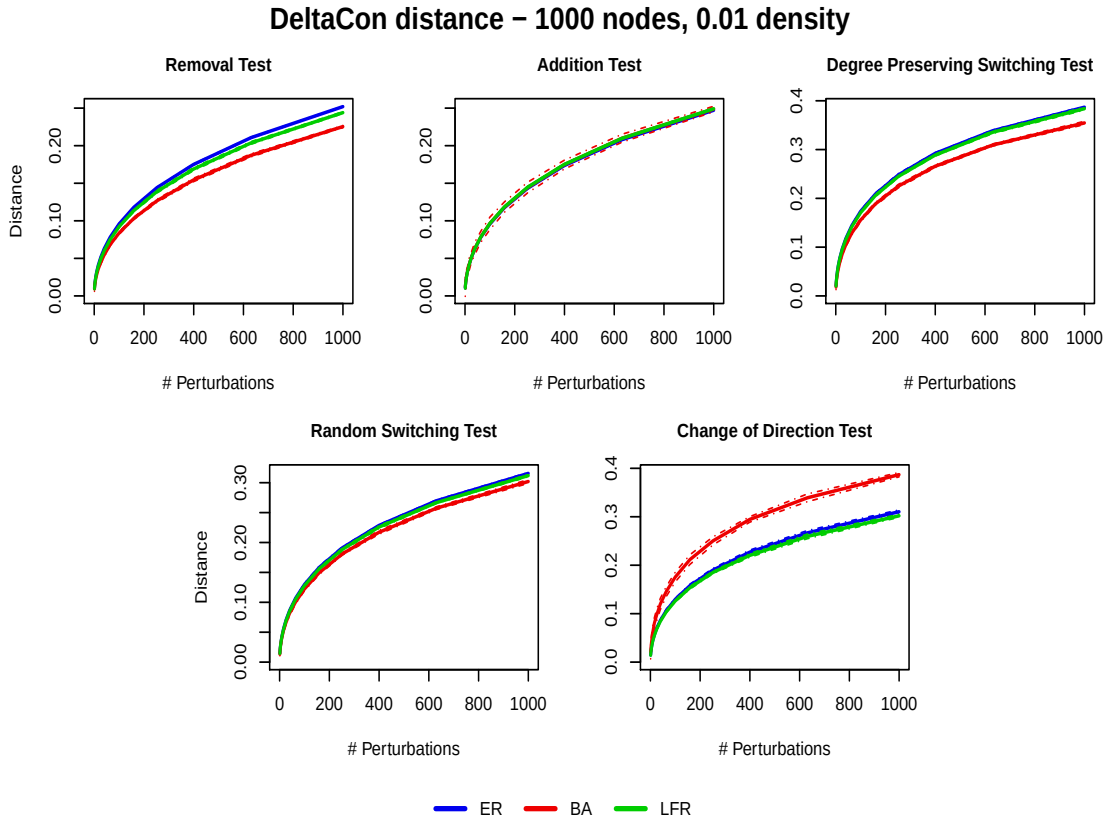


(a)

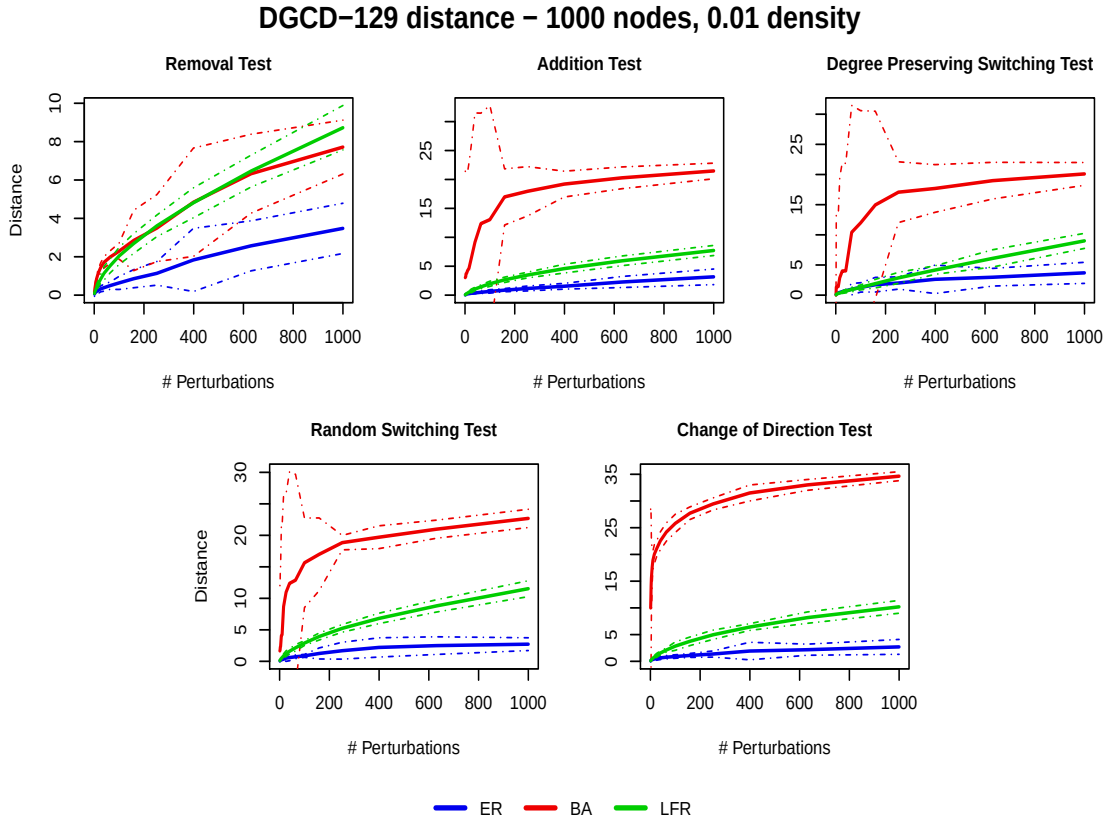


(b)

Figure 2.3: Results of the successive perturbations test on directed and unweighted synthetic networks with 0.01 density.

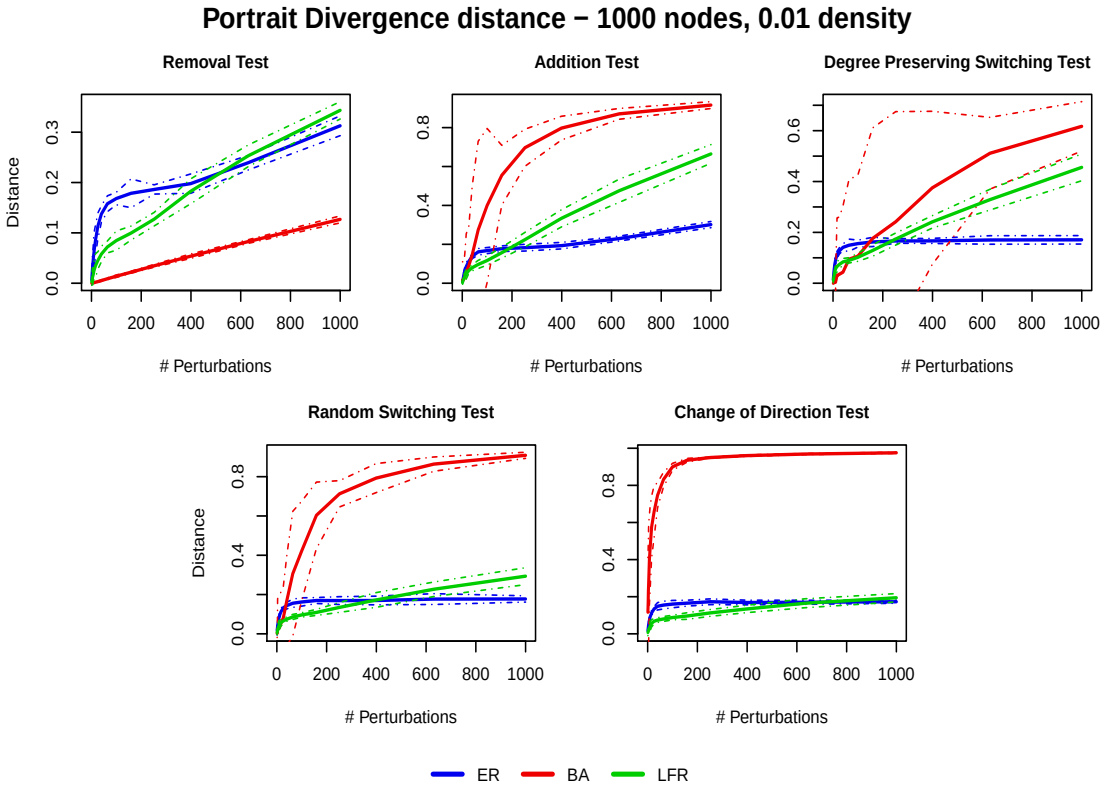


(c)

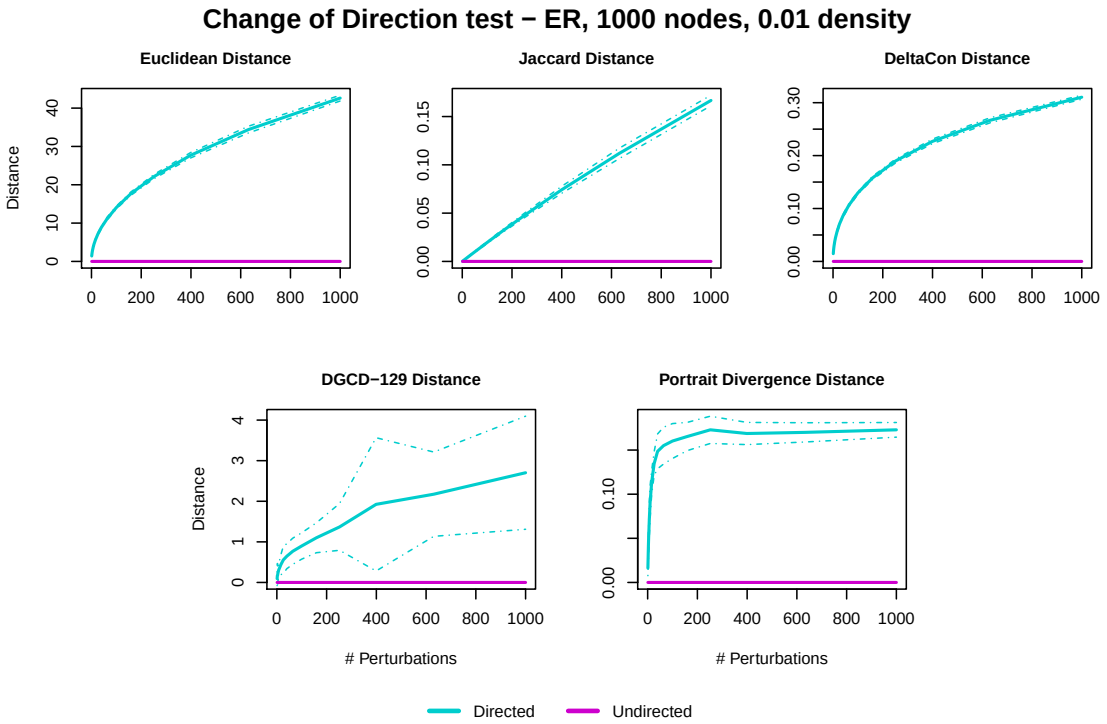


(d)

Figure 2.3 (cont.): Results of the successive perturbations test on directed and unweighted synthetic networks with 0.01 density.



(e)



(f) Change of direction test on directed/undirected and unweighted synthetic networks with 0.01 density.

Figure 2.3 (cont.): Results of the successive perturbations test on directed and unweighted synthetic networks with 0.01 density.

The DGCD-129 and the Portrait Divergence distances instead have issues in some tests for what concerns the BA topology. Indeed, the DGCD-129 produces a large distance even when only one perturbation is performed (this happens in the addition, random switching and change of direction tests); the same behaviour characterizes the Portrait Divergence distance in the change of direction test. The BA topology also shows a high variability in the first 200 perturbations using DGCD-129, while for the Portrait Divergence distance the variability remains high for all the perturbations only in the degree-preserving switching test. This issues, that are mainly related to the BA curve, are due to the particular structure of the directed Barabási-Albert graph. Indeed, by construction it has the same out-degree for all the nodes but different in-degree, with the hubs having the largest in-degrees. This results in a strongly organised structure and few changes in the connections or in the edge directions are enough to produce dramatic changes in the number and kind of graphlets observed and in the shortest path distribution. We did not consider neither a BA topology with hubs having large out-degrees nor a BA topology where edges have randomized directions, but they should be taken into account to provide an exhaustive analysis of these distances.

We then observe that, as in the undirected and unweighted case, we have a different behaviour between distances which require node correspondence and distances independent on that. Considering the first group (Figures 2.3a to 2.3c), all distances show a very similar trend, with almost equal values for each network topology and for each kind of test. The DeltaCon distance is the only one where, in some tests, the BA curve has slightly different values from the other curves. This is again a prove that this class of distances is not aware of the network topology of the graphs.

The metrics independent on node correspondence (Figures 2.3d and 2.3e), instead, produce different patterns for different topologies and tests. Overall, we can notice that now, unlike in the perturbation tests on undirected and unweighted networks, the BA curve always stays above the others, in all the tests except the removal one. This behaviour is explained with the same motivation as before: having a strong organized structure, few perturbations are enough to produce large changes in the characteristics of the BA graph. In the removal tests of the Portrait Divergence distance the BA curve stays below the others, while in the removal tests of DGCD-129 it is almost equal to the LFR curve. This is due to the fact that the removal tests is the one that least affects the strongly organized structure of the BA topology. About the Portrait Divergence distance, we note that interestingly in the removal test the BA curve does not present the typical initial step, but instead has a linear trend. Moreover, note in the change of direction test the fast convergence of the BA curve to the maximum value (that is 1), denoting once more how much the BA structure is affected by only few changes of the edge directions.

Finally, we show in Figure 2.3f the curves related to ER networks computed in the change of direction test for each distance, comparing them with the curves computed by their undirected counterparts. As expected, the latter are identically zero. We only show the ER curves since also the BA and the LFR curves present an identical behaviour. The result clearly shows that all the considered distances exploit the additional information given by the edge directions and are thus suitable tools to analyse directed networks.

Results for networks with 0.05 density

The results for the networks with 0.05 edge density are shown in Figure 2.4. We observe for all the metrics and all the tests almost the same behaviour and issues already described for the networks with 0.01 density, both for the distances which require node correspondence and for the distances independent on that. We only notice that, in all the tests except the removal one, the BA curve of the DGCD-129 distance has a pronounced step in the first few perturbations, unlike in the 0.01 edge density case. Moreover, we also observe, mainly in the Jaccard and in the DeltaCon distances, a decrease in the value of the measured distances for the same fixed number of perturbations. As in the undirected and unweighted case, this shift reflects the presence of more edges in the graphs. We show as example in Figure 2.4f the curves related to the ER networks computed on the change of direction test for each method, comparing them with the curves computed by their undirected versions. Also in this case the latter are identically zero, and the BA and LFR curves present the same behaviour.

Therefore, as in the undirected and unweighted case, the behaviour of the methods is not much influenced by the change in the density of the original graphs. Notice, however, that also in this case we analysed only two values of edge density, so that this is not an exhaustive analysis.

2.3 Clustering networks

To test the effectiveness of each method to recognize and group together different types of networks, we performed a clustering test using some graphs generated from various network models. The more a network distance assigns small distances to pairs of networks coming from the same family and large distances to pairs of networks coming from different families, the better it performs in the clustering task.

2.3.1 Description

To perform the clustering tests, one for the undirected and unweighted case and one for the directed and weighted case, we chose to use again the same three families of networks we used in the successive perturbation tests, namely Erdős-Rényi, Barabási-Albert and LFR (with mixing coefficient $\mu = 0.2$) graphs. For each network family, we generated 5 networks for each of the following values of number of nodes and edge densities: 1 000 and 2 000 nodes and 0.01 and 0.05 edge densities. This results in the generation of 20 networks for each family, for a total of 60 networks analysed. We report in Table 2.3 the parameters of the graphs with 2 000 nodes (both undirected and directed) used in the clustering tests, while the parameters used to generate the graphs with 1 000 nodes are those already contained in Table 2.2. We computed all the pairwise distances in this set of graphs and we ended up with a 60×60 distance matrix for each one of the network distances used.

The clustering tests are carried out in the same way both for the undirected/unweighted and the directed/unweighted case. Only the distances used change:

Table 2.3: Parameters used to generate the networks (both undirected and directed) with 2 000 nodes used in the clustering tests

	Undirected		**Directed**	
Nodes	2 000	2 000	2 000	2 000
Density	0.01	0.05	0.01	0.05
Edges	19 990	99 950	39 980	199 900

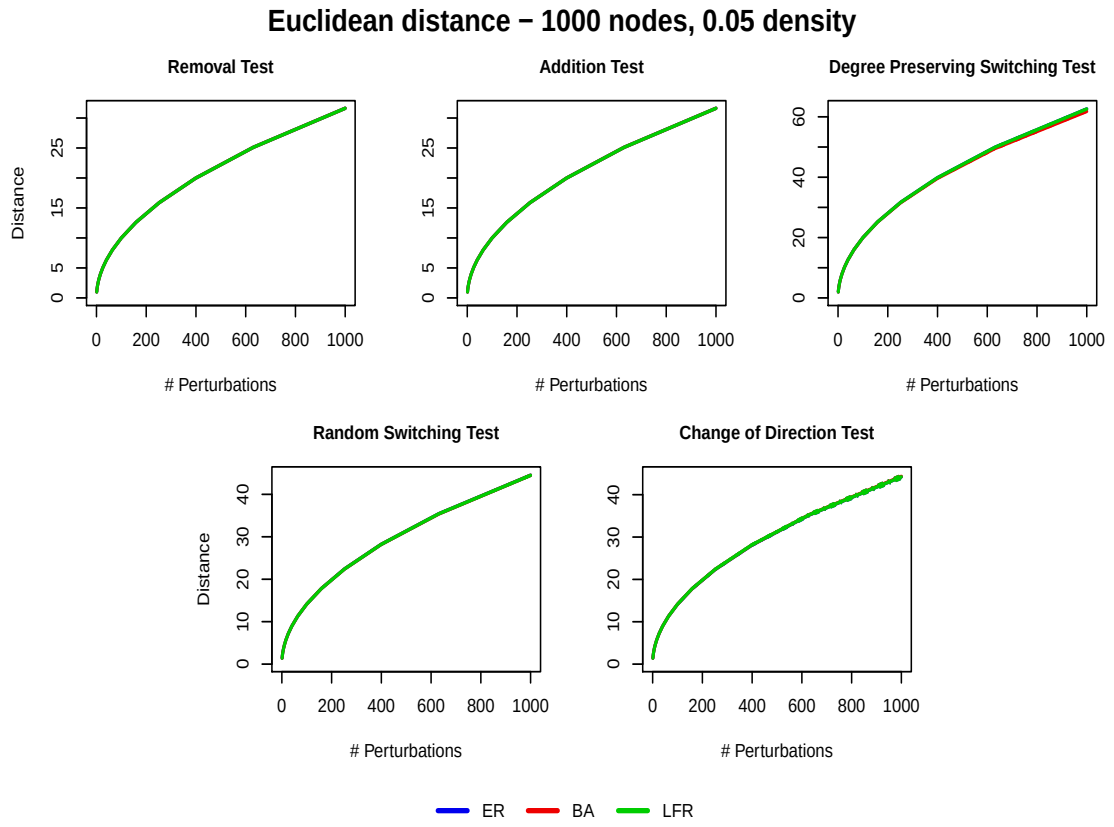
(a) Parameters for Erdős-Rényi graphs

	Undirected		**Directed**	
Nodes	2 000	2 000	2 000	2 000
Density	≈0.01	≈0.05	≈0.01	≈0.05
Edges added in each step	10	51	20	103
Resulting edges	19 945	100 674	39 790	200 644

(b) Parameters for Barabási-Albert graphs

	Undirected		**Directed**	
Nodes	2 000	2 000	2 000	2 000
Density	≈0.01	≈0.05	≈0.01	≈0.05
Exponent of degree distrib. (γ)	3	3	3	3
Exponent of community size distrib. (β)	1	1	1	1
Mean degree	20	100	20	100
Maximum degree	900	1450	1100	1600
Mixing parameter (μ)	0.2	0.2	0.2	0.2
Minimum degree in community	5	5	5	5
Required edges	19 990	99 950	39 980	199 900
Edge tolerance	50	100	100	200

(c) Parameters for LFR graphs. For directed graphs, mean and maximum degree are intended as mean and maximum in-degree.

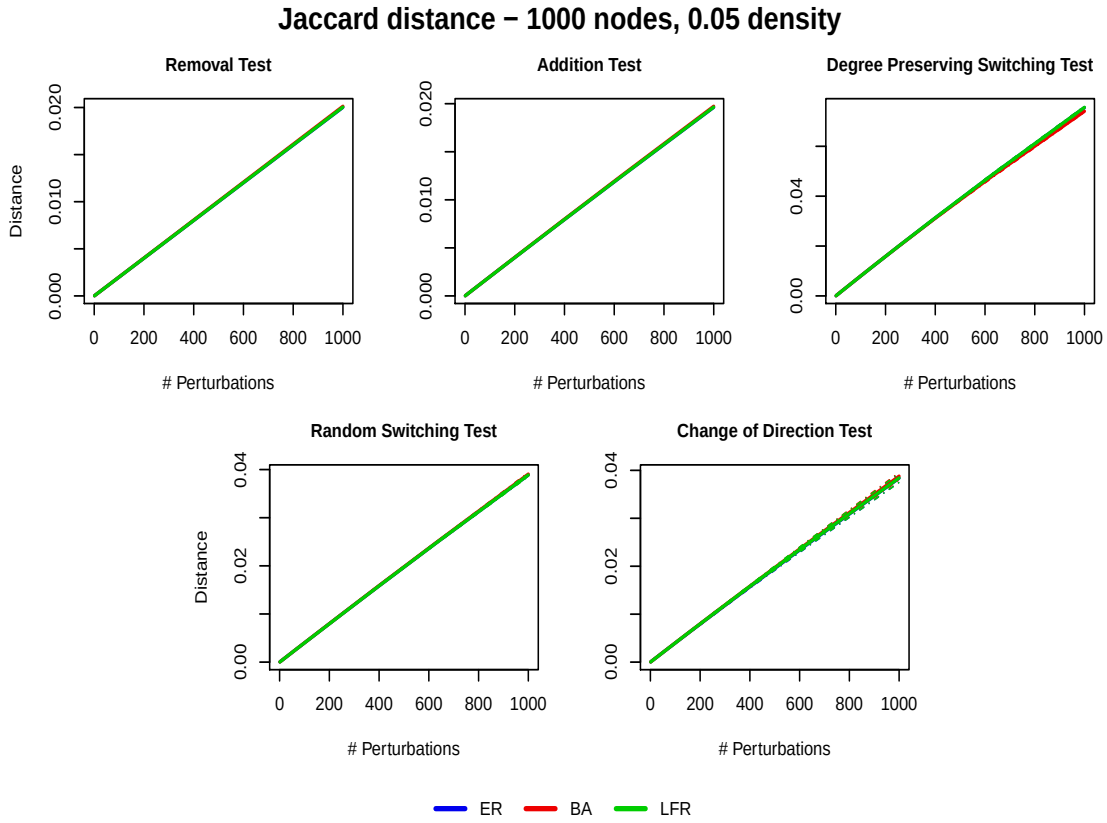


(a)

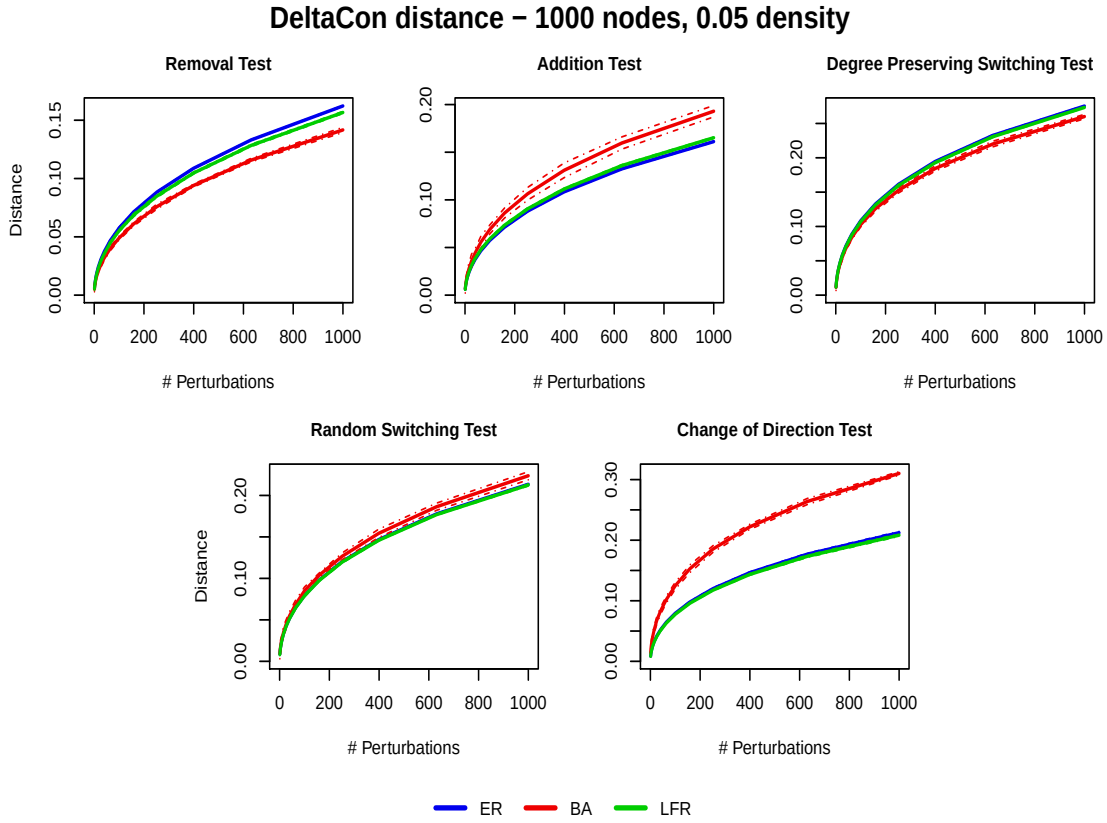
Figure 2.4: Results of the successive perturbations test on directed and unweighted synthetic networks with 0.05 density.

referring to Table 2.1, we used the methods in first row, second column for the undirected/unweighted case and the methods in the third row, second column for the directed/unweighted case (excluding MI-GRAAL, which is computationally too heavy: an estimation of the computational time needed, on the basis of the results obtained in Section 2.4, is of about 42 days). We only used distances which are independent on node correspondence because, unlike in the perturbation tests (in which we always modify the same graph, so that node correspondence is kept), here we generated separately the networks for each family and in this situation a node correspondence cannot be established. This means that, if used in this situation, distances which require node correspondence will give totally meaningless results.

To properly evaluate the performances of the various methods in the task of clustering networks, we use a Precision-Recall analysis framework, as explained in Appendix B. In this context, Precision-Recall analysis is a robust and a reliable tool to evaluate the methods' performances because it is based only on the distance matrix produced by the methods and it does not introduce any additional parameter, as the use of dendrograms would do. Indeed, the use of dendrograms to asses the clustering results requires to choose an additional parameter to build the trees, i.e. the linkage method. The result is that the dendrogram structure can change based

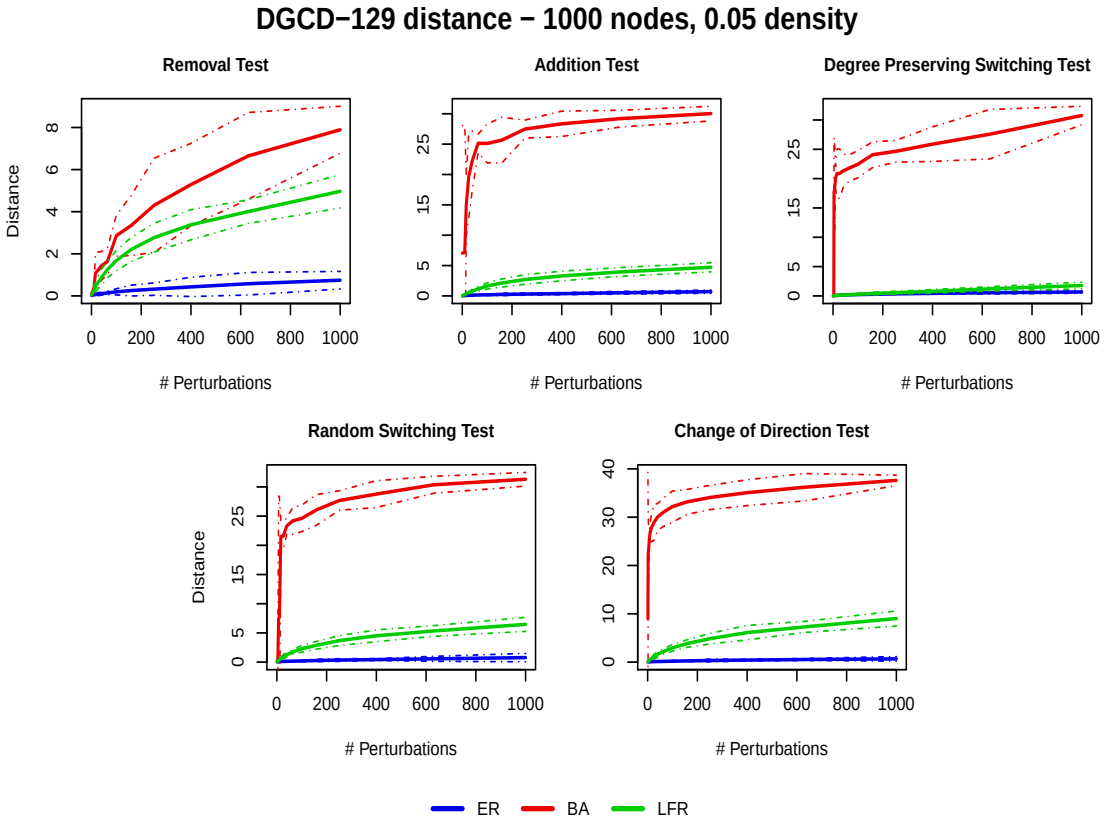


(b)

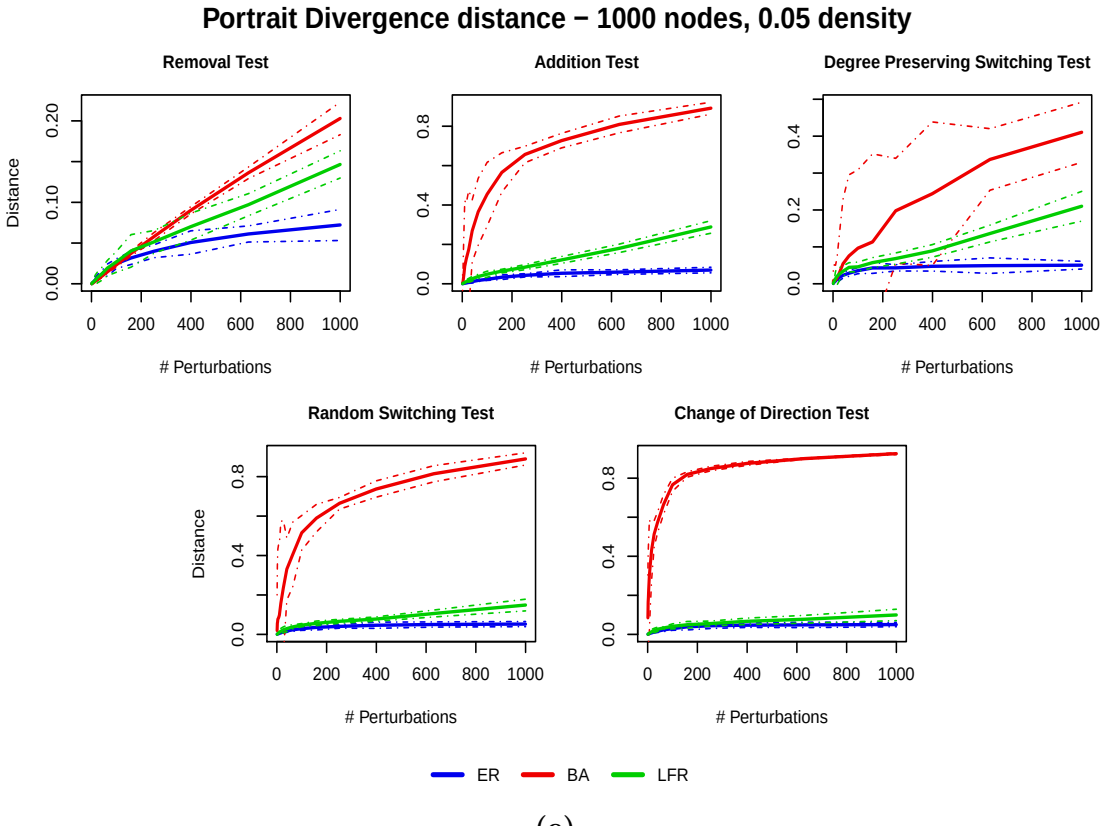


(c)

Figure 2.4 (cont.): Results of the successive perturbations test on directed and unweighted synthetic networks with 0.05 density.

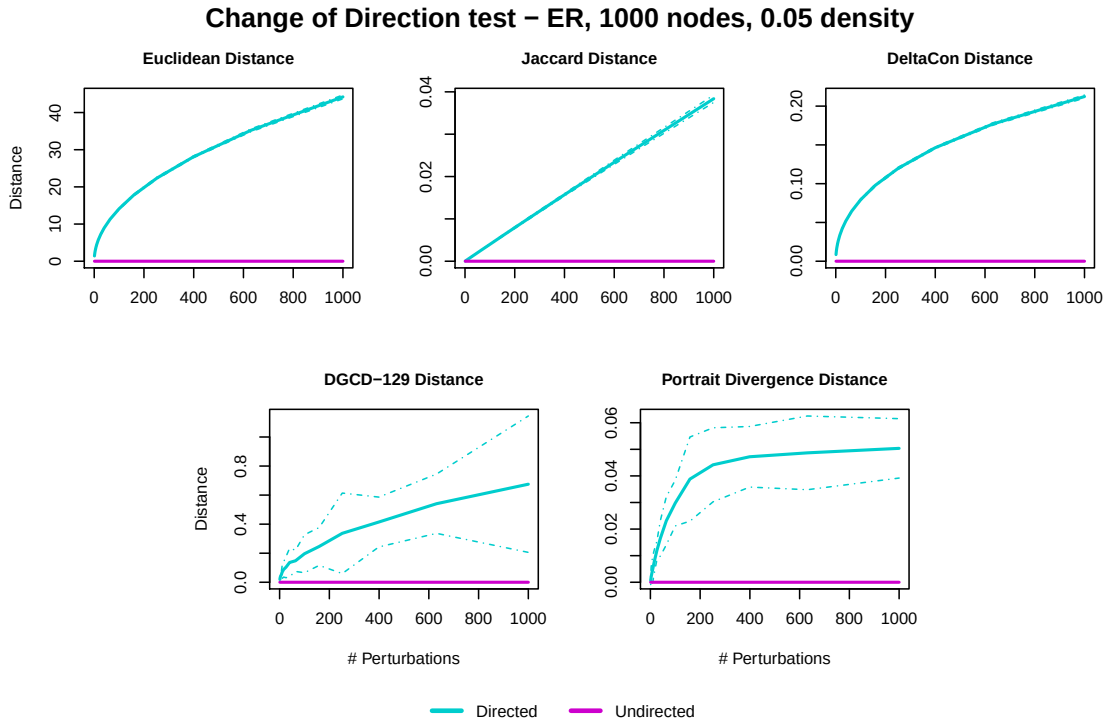


(d)



(e)

Figure 2.4 (cont.): Results of the successive perturbations test on directed and unweighted synthetic networks with 0.05 density.



(f) Change of direction test on directed/undirected and unweighted synthetic networks with 0.05 density.

Figure 2.4 (cont.): Results of the successive perturbations test on directed and unweighted synthetic networks with 0.05 density.

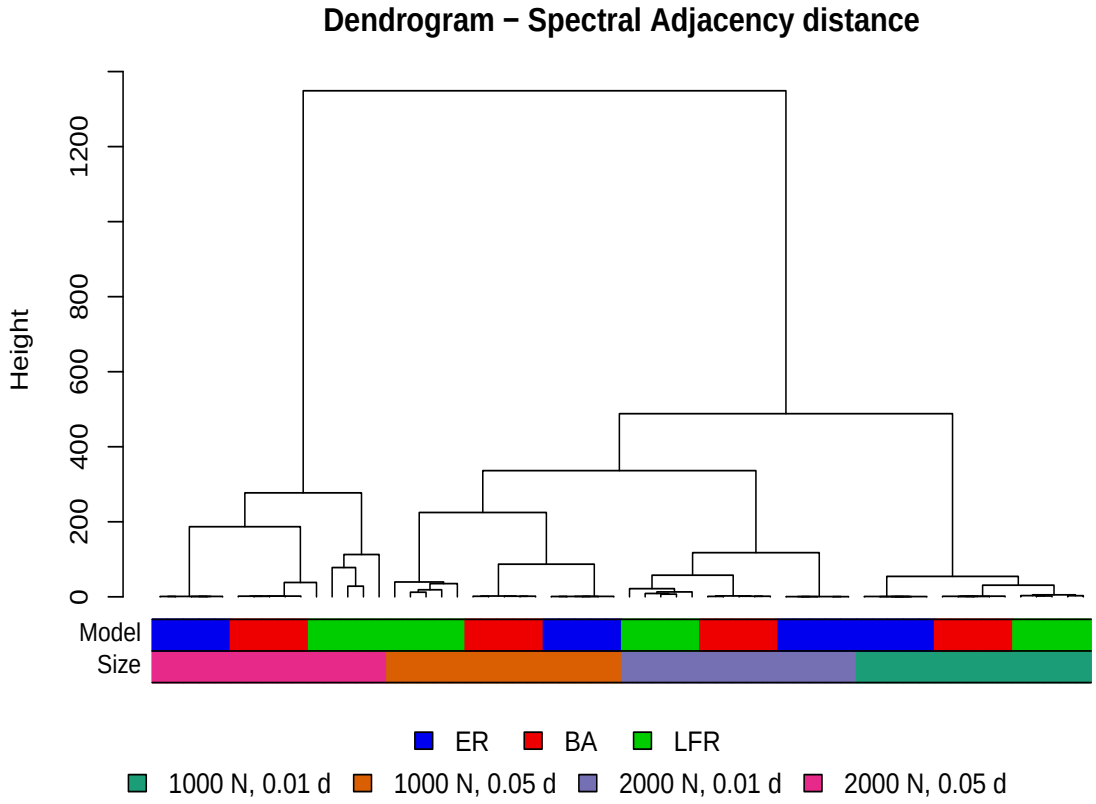
on the linkage, and consequently the interpretation of the results may vary. In the following, we will anyway use dendrograms (built with *ward.D2* linkage [60]) since they are an effective and clear way to display results, always combining this tool with Precision-Recall analysis or providing the Cophenetic Correlation Coefficient [39, 71] to asses the quality of the dendrograms (that is, how much their structure reflects the underlying distance matrix).

2.3.2 Results

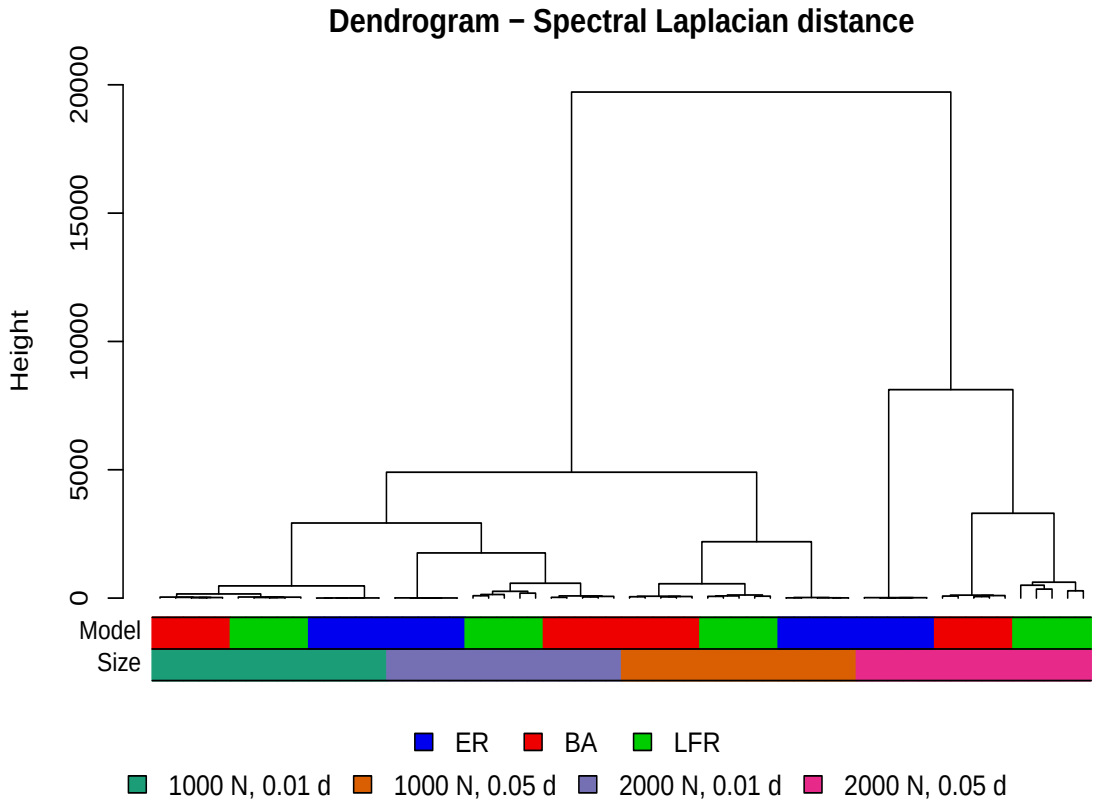
Undirected and unweighted case

We present in Figure 2.5 the dendrograms that we obtained from the distance matrices obtained from the network distances. We ordered the leaves of the dendrograms with the Optimal Leaf Ordering (OLO) algorithm [10], in order to minimize the distance between adjacent leaves. To better understand and describe the results, we also coloured the leaves of the dendrograms with respect to two criteria: the first colouring denotes the division of the networks into the three families of network models, while the second colouring denotes the division with respect to size and density with which the networks were generated.

By looking at the coloured bars under the leaves, we notice that all the distances correctly group together networks which are generated from the same network

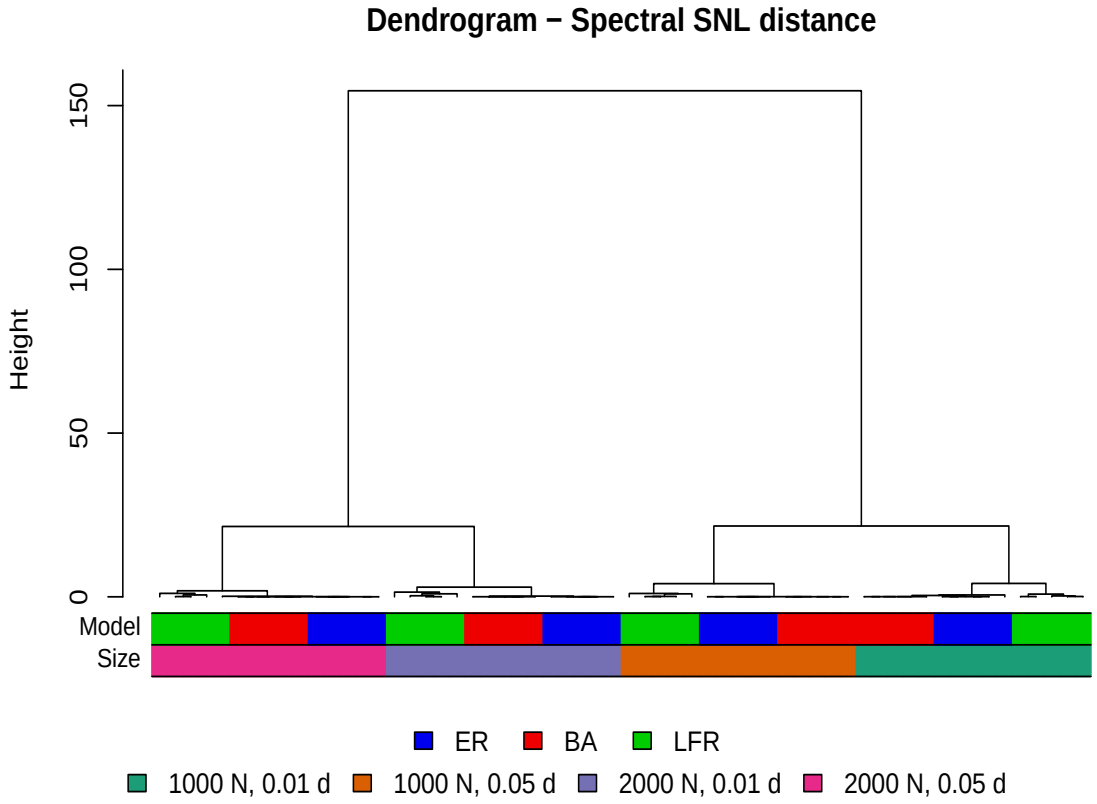


(a)

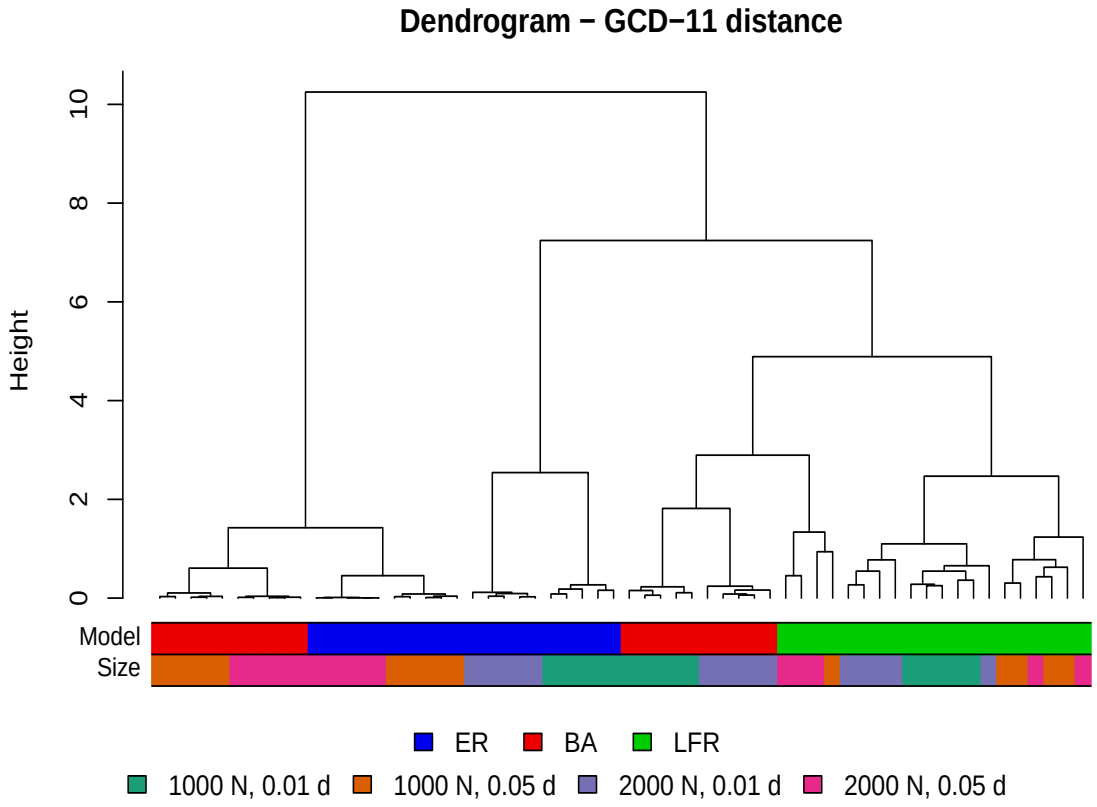


(b)

Figure 2.5: Dendrograms obtained from the clustering test on undirected and unweighted synthetic networks. In the legend, N stands for number of nodes, while d stands for edge density.

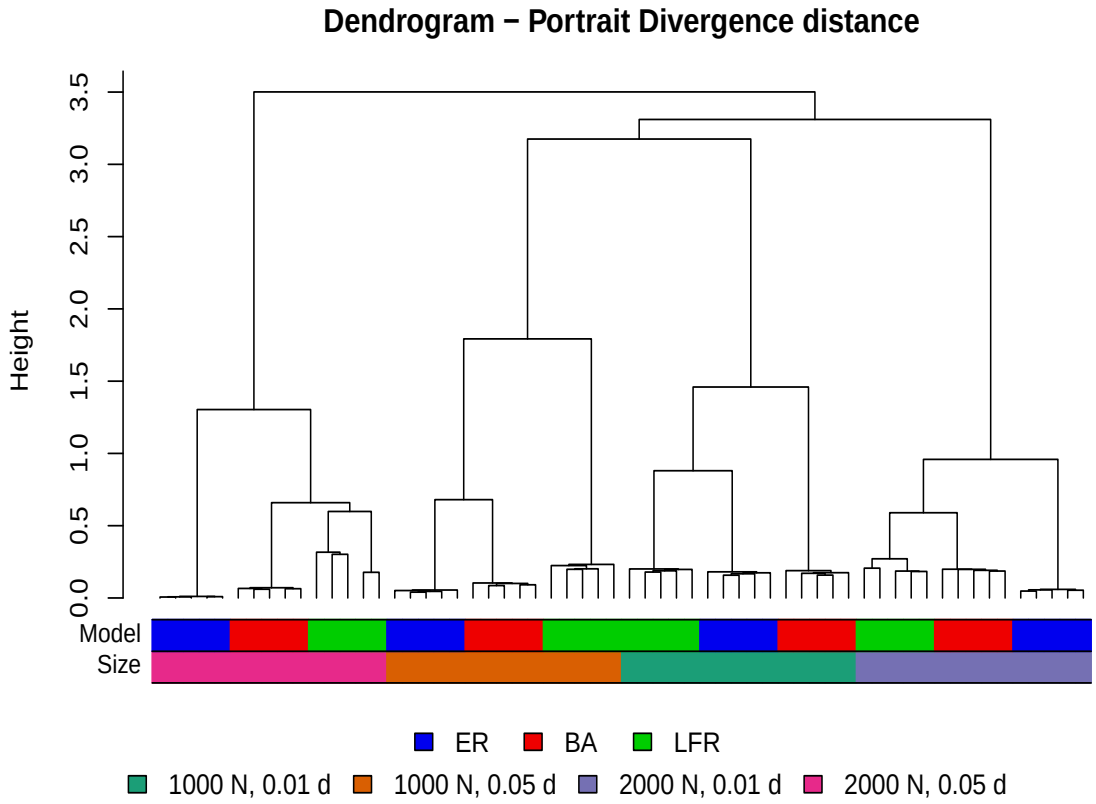


(c)



(d)

Figure 2.5 (cont.): Dendrograms obtained from the clustering test on undirected and unweighted synthetic networks.



(e)

Figure 2.5 (cont.): Dendrograms obtained from the clustering test on undirected and unweighted synthetic networks.

model and have the same size and density, but the GCD-11 distance is the only one that achieves a better clustering by grouping together networks coming from the same network family but with different sizes and densities. Indeed, GCD-11 groups together all the LFR networks, along with the BA graphs with low densities, and identify two other clear clusters: one containing only the ER graphs with low densities, and the other containing the BA and ER graphs with high densities. Instead, by looking at the second row of the coloured bars, all the other distances group the networks based on the size and the density they have, and not based on the network family they belong to. In other words, if we consider two networks with different sizes and densities, the large distance given by these metrics is more likely due to the different size and density rather than to an actual difference in the network topology. Thus, these distances are strongly dependent upon network size and density and they do not provide reliable results in such situations. Only GCD-11 is able to group networks actually basing on their topology structures, showing only a dependence on edge density. These results clearly show that, for the undirected and unweighted case, GCD-11 is the only network distance independent on node correspondence which proves reliable in the clustering task when heterogeneous networks are considered. This is confirmed also by the comparison of the Precision-Recall curves shown in Figure 2.6, where the PR curve of the GCD-11 distance always stays above the other curves and has the highest AUPR. The values of AUPR for all the distances are reported in Table 2.4a; note that the Spectral SNL distance is the worst performing method, with a performance which is only slightly better than a random classifier, and that the other two Spectral distances and Portrait Divergence have comparable performances.

A deeper analysis is needed to understand whether the analysed distances can anyway perform well in discriminating the various network families when networks with same size and density are compared. For each one of the four possible combinations of size and density, we reduced the distance matrices to consider the networks with same size and density and we computed again the Precision-Recall curves and the AUPR for each method. The results are shown in Table 2.4b, showing that all methods performs much better in situations where the graphs to be compared have the same size and density, achieving much higher AUPR than a random classifier and often providing a perfect classification. Overall, the worst performing distance is Spectral SNL, which never achieves a perfect classification and always have the lowest values of AUP; this is due to the fact that the method never distinguishes between ER and BA graphs. We also have an evidence of the dependence on density of GCD-11, since its AUPR values show a large decrease when denser networks are considered.

All the distances considered in the undirected and unweighted clustering test are then suitable for the clustering task and they are able to distinguish among different topologies, if all the input networks have comparable sizes and densities; when this condition is not met, only GCD-11 should be use to produce reliable results about the topological similarity of the networks.

Table 2.4: AUPR of the methods in the clustering test on undirected/unweighted networks

(a) AUPR values when all networks are considered

Distance	**AUPR**
Spectral Adjacency	0.4578784
Spectral Laplacian	0.4714461
Spectral SNL	0.3909493
GCD-11	0.6881952
Portrait Divergence	0.4549765
Random classifier	0.3220339

(b) AUPR values when only networks with same size and density are considered. The AUPR of a random classifier is 0.2857143 in all situations.

Distance	**AUPR**
Spectral Adjacency	1
Spectral Laplacian	1
Spectral SNL	0.8059879
GCD-11	1
Portrait Divergence	1

1000 nodes and 0.01 density

Distance	**AUPR**
Spectral Adjacency	0.9847082
Spectral Laplacian	1
Spectral SNL	0.8327631
GCD-11	0.8541507
Portrait Divergence	1

1000 nodes and 0.05 density

Distance	**AUPR**
Spectral Adjacency	0.9960190
Spectral Laplacian	0.9692216
Spectral SNL	0.8344244
GCD-11	1
Portrait Divergence	0.9951681

2000 nodes and 0.01 density

Distance	**AUPR**
Spectral Adjacency	0.8542287
Spectral Laplacian	1
Spectral SNL	0.7817300
GCD-11	0.8162835
Portrait Divergence	0.8604364

2000 nodes and 0.05 density

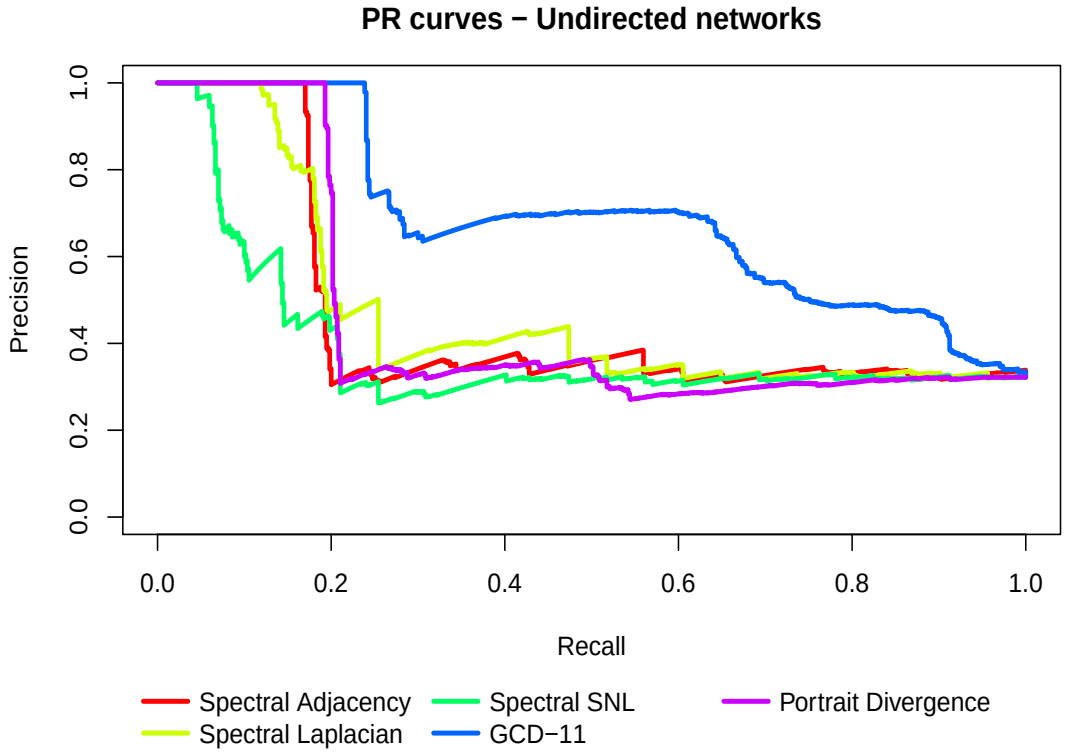
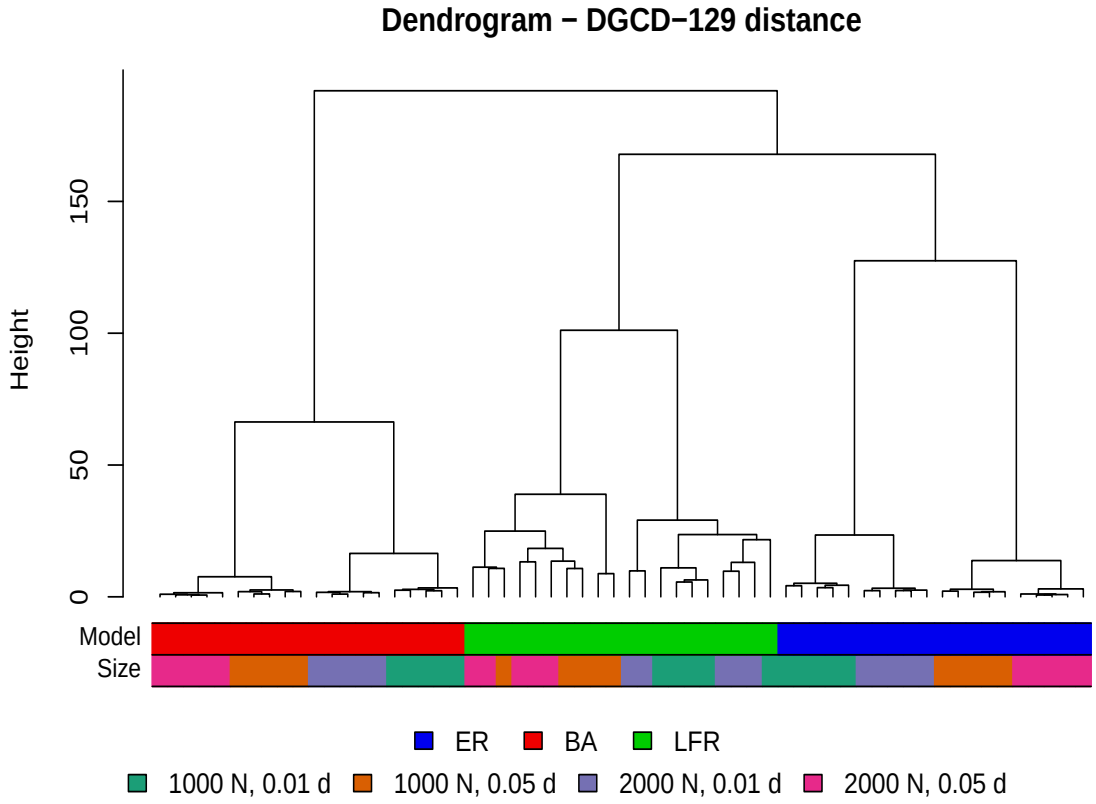


Figure 2.6: Precision-Recall curves of the methods used in the clustering test on undirected and unweighted synthetic networks.

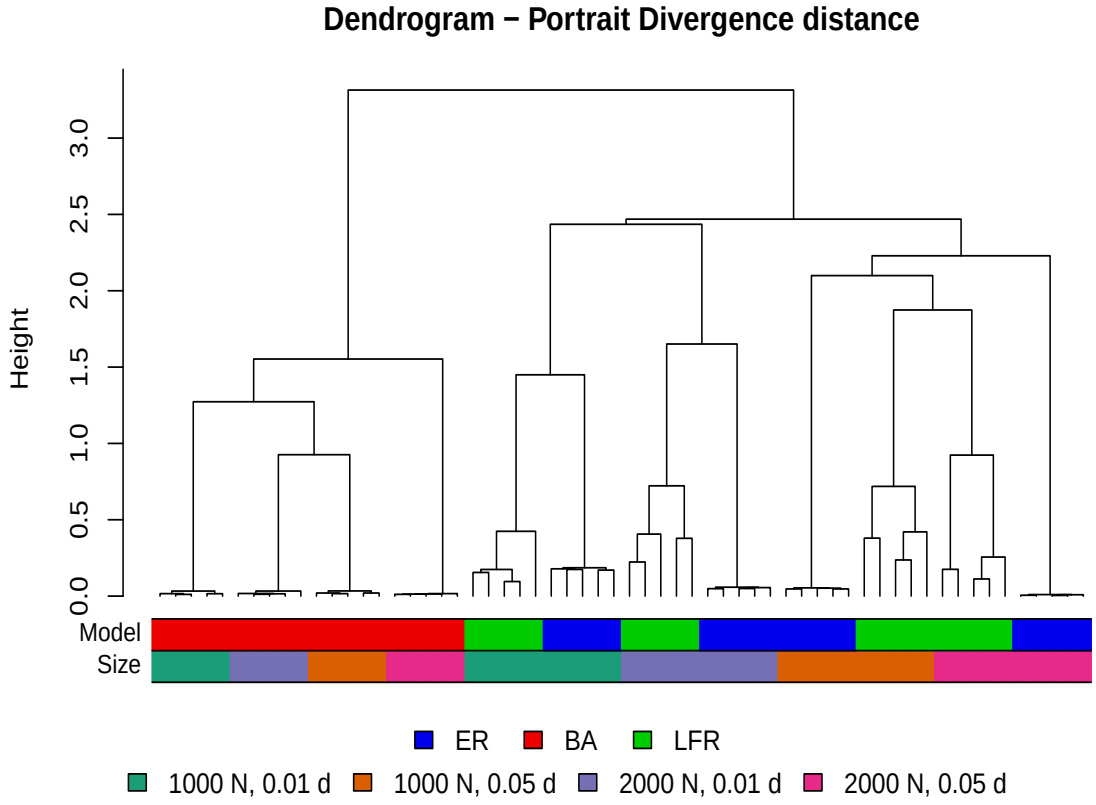
Directed and unweighted case

We present the results in the same way as we did for the undirected and unweighted case, by showing the dendrograms obtained from the distance matrices. Again, we coloured the leaves according to the network families the networks belongs to and according to their sizes and densities (Figure 2.7).

We see two different behaviours. In Figure 2.7a, we see how DGCD-129 does not depend on network sizes and densities and it is able to perfectly recover the division into the three original clusters. Moreover, inside the BA and ER families, it is also able to correctly group together networks whit same sizes and densities. In Figure 2.7b, the Portrait Divergence distance behaves better than in the undirected and unweighted case, since it is able to recover in one cluster all the BA graphs. It groups all the ER and the LFR graphs in the other cluster, which can be seen as the union of other two very close clusters, one containing all the graphs with 0.01 edge density, and the other containing all the graphs with 0.05 edge density. This behaviour shows that in the directed and unweighted case the Portrait divergence distance has better performances than in the undirected and unweighted case and it is only partially dependent on network densities. As confirmed by the Precision-Recall curves (Figure 2.8), DGCD-129 clearly outperforms Portrait Divergence in recognizing the different network topologies.



(a)



(b)

Figure 2.7: Dendrograms obtained from the clustering test on directed and unweighted synthetic networks. In the legend, N stands for number of nodes, while d stands for edge density.

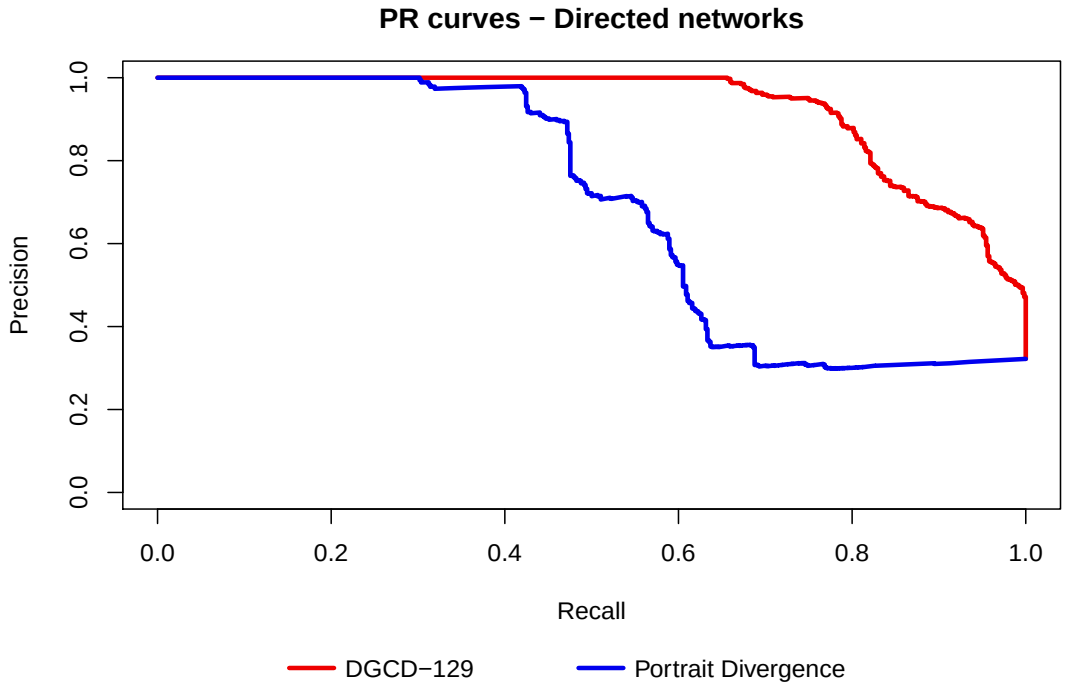


Figure 2.8: Precision-Recall curves of the methods used in the clustering test on directed and unweighted synthetic networks.

Table 2.5: AUPR of the methods in the clustering test on directed/unweighted networks

(a) AUPR values when all networks are considered

Distance	AUPR
DGCD-129	0.9276987
Portrait Divergence	0.6845735
Random classifier	0.3220339

(b) AUPR values when only networks with same size and density are considered. The AUPR of a random classifier is 0.2857143 in all situations.

Distance	AUPR
DGCD-129	1
Portrait Divergence	1

1000 nodes and 0.01 density

Distance	AUPR
DGCD-129	1
Portrait Divergence	1

1000 nodes and 0.05 density

Distance	AUPR
DGCD-129	1
Portrait Divergence	0.9951681

2000 nodes and 0.01 density

Distance	AUPR
DGCD-129	1
Portrait Divergence	1

2000 nodes and 0.05 density

We report in Table 2.5b the AUPR values computed for the distances when considering only the networks with same sizes and densities. As already observed, in any situation the methods perform much better and are able to perfectly distinguish among the three different topologies.

The distances that we considered for the directed and unweighted case are then suitable for the clustering task and for recognizing different network topologies in any situations; in particular, when graphs with different sizes and densities are considered, DGCD-129 outperforms Portrait Divergence.

2.4 Testing execution times

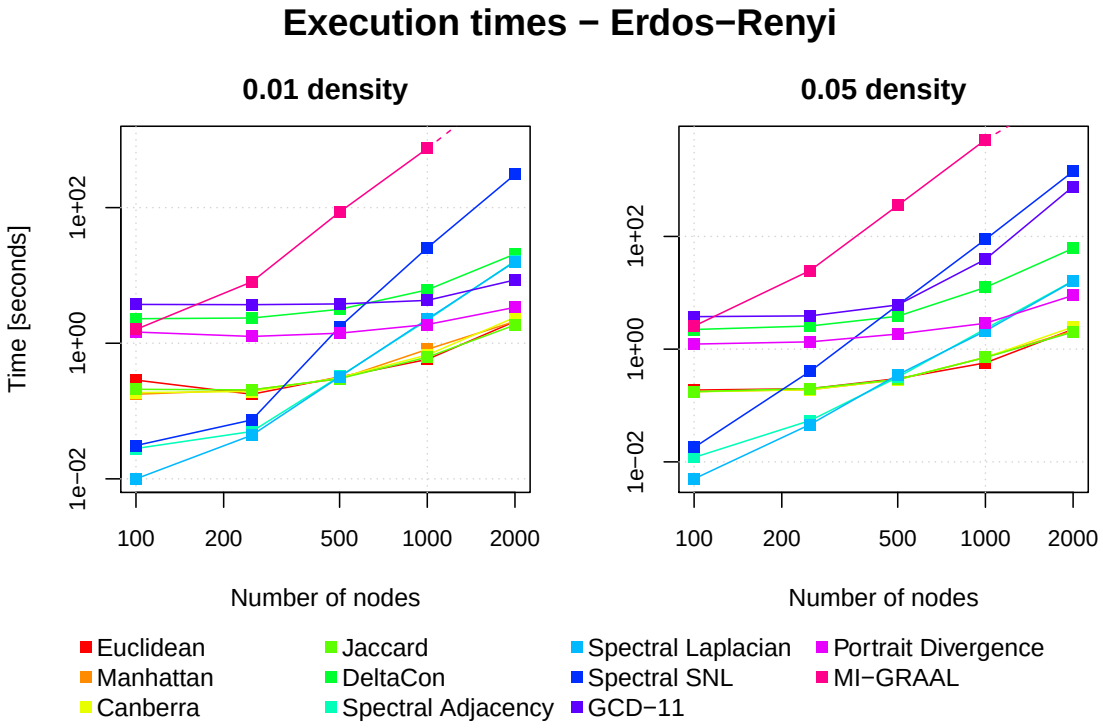
Tests on synthetic networks are carried out to evaluate the execution times needed by the various methods, to understand their scalability and their usability in the task of comparing a large set of networks.

2.4.1 Description

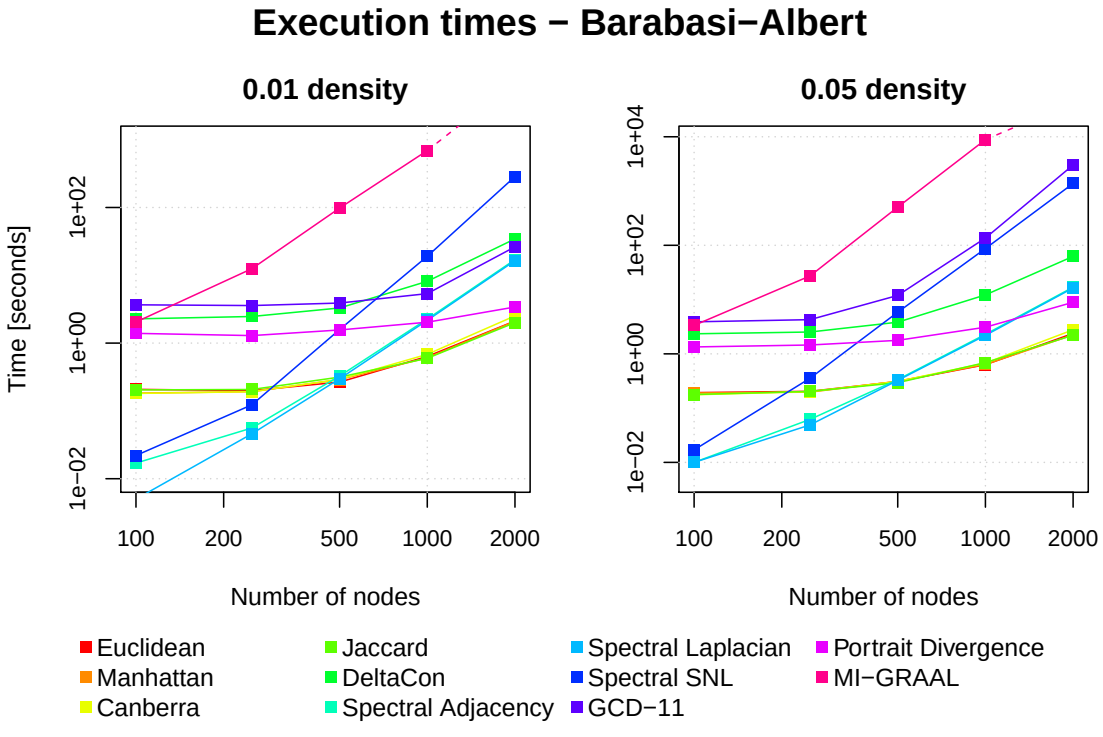
We carried out the analysis of the computational efficiency using two different network topologies, namely Erdős-Rényi and Barabási-Albert graphs (see Appendix A.3). For each one of them, we considered five different sizes, i.e. 100, 250, 500, 1 000 and 2 000 nodes, and two different densities, i.e. 0.01 and 0.05. We then have 10 different combination of parameters for each network model. For each combination of parameters, we generated 10 pairs of graphs and we computed the distances between them using the methods of the undirected and unweighted case (see Table 2.1); finally, we computed the average time needed to calculate the 10 distances. Note that the execution times include the time needed to compute the network characteristics which is used for the comparison other than the execution time of the comparison itself; in some cases it is also included some extra time needed to write to and read from files the network characteristics. Moreover, the time needed to perform a pairwise comparison between more than two network scales less than linearly with the input size, since the code is optimized to compute once for each network the network characteristic needed for the comparison and then to compute all the pairwise distances, so that the same network characteristic is not computed multiple times.

2.4.2 Results

The results of the analysis are shown in Figure 2.9. In each panel of the two figures, we display the average performances for increasing size and fixed density. Note that, for the MI-GRAAL distance when graphs of 2 000 nodes were considered, we did not let the execution finish since each single comparison took more than 12 hours. Then, we used 12 hours as the average performance value in this case, and we represented this (best case) estimation with a dashed line for the last part of the MI-GRAAL curves.



(a)



(b)

Figure 2.9: Average execution times for the comparison of two networks.

The qualitative results on the two different topologies are almost identical, so that we discuss them without distinguishing between the two. We see two different behaviours: some methods show a linear trend, meaning that they have polynomial dependence on the number of nodes N, while other methods show a more than linear trend, meaning that their dependence on N is more than polynomial. The first group consists of the Spectral distances and apparently of MI-GRAAL (the last point of its curve is a best case estimation, as we already mentioned), while the second group gathers all the other distances. Moreover, we notice that the four naïve distances, namely the Euclidean, the Canberra, the Manhattan and the Jaccard distances, have almost equal computational times, and the same happens for the Spectral Adjacency and the Spectral Laplacian distances. The Spectral SNL distance, instead, takes larger values and it is strongly affected by the increase in the density: indeed, the computation of the Symmetric Normalized Laplacian requires a loop over all the edges. The network density strongly affects some methods (Spectral SNL, GCD-11, Portrait Divergence, DeltaCon and MI-GRAAL), but does not produce relevant changes in the others (the other two Spectral distances and the naïve distances). This different behaviour is due to the different network characteristics that each method has to compute to perform the comparison: for instance, the computation of the adjacency matrix for the naïve distances is not affected by the network density, while the computation of the graphlets for GCD-11 becomes more and more demanding as the number of edges increases.

Although this is not an exhaustive analysis of the computational times, since we considered only graphs with no more than thousands nodes, we can draw some conclusions about which method is best suited in which situation, with respect to the computational time. The Spectral Adjacency and the Spectral Laplacian distance are the fastest methods for small graphs (till 500 nodes), while the Euclidean, Manhattan, Canberra and Jaccard distances are the fastest ones for medium graphs (from 500 to 2 000 nodes); none of them is influenced by network density. Conversely, MI-GRAAL is the slowest methods (except for very small graphs), requiring hours of computation for medium sized graphs on the machine we used; this makes it impractical to use also for the comparison of large sets of networks. All the methods that show a more than linear dependence on N, especially GCD-11, should be avoided for the comparison of graphs with more than thousands of nodes, of too dense graphs or of a large number of networks; nevertheless, the execution times required for the comparison of sparse graphs with few thousands of nodes are still acceptable. The Spectral SNL distance should be avoided as well when comparing networks with a large number of edges, even if its dependence on N is polynomial. Moreover, even if our analysis does not highlight this behaviour, all the Spectral distances should be avoided when networks with more than thousands of nodes are considered, due to the high computational cost of computing eigenvalues of large matrices. To conclude, space issues may affect mainly the methods based on the computation of the matrices representing the graphs (adjacency matrix and Laplacian); nonetheless, this problem is easily overcome by using sparse matrix representation.

Chapter 3

Analysis of real-world networks

In this Chapter, we illustrate the application of the methods studied in **??** to real-world networks. More precisely, we selected three datasets that can be interpreted as multilayer networks [45]. The choice of multilayer networks is motivated from the fact that they naturally are sets of graphs that can be compared: each layer can be seen as a separate network defined over the same node set. This is a desirable property for our analysis, because it enables the use of the distances which require node correspondence. Then, we can evaluate how the two classes of methods, that is, distances which require node correspondence and distances which are independent on node correspondence, behave in the same task of clustering the layers and if they provide different insights. We present separately the three dataset we used along with the results of the analysis.

3.1 European Air Transportation network

3.1.1 Description of the dataset

The first multilayer network we analyse is the European Air Transportation network (EATN) updated at 2011, which was already considered in [14]. We downloaded a preprocessed version of the dataset from [25]. This network has 450 nodes, each one representing an European airport, along with an identifier and the geographical coordinates. We actually found two nodes which have zero coordinates and no reasonable identifier, so that we deleted it from the dataset. The actual size of the node set is thus 448. The total number of edges is 3 588, distributed over 37 layers, each one representing a different airline (we list in Table 3.1 the 37 airlines). All the layers are undirected and unweighted.

To analyse this dataset we used the distances suitable to the undirected and unweighted case: Euclidean, Jaccard, DeltaCon, the three Spectral distances, MI-GRAAL, GCD-11 and Portrait Divergence. We decided to use MI-GRAAL for this analysis to have an idea of its performances in the clustering task; the computational time here is not a problem since the single layers, taken without isolated nodes, are very small and they can be compared in reasonable time. Note that the first three distances we listed require node correspondence between the graphs. As we already mentioned, unlike in the clustering tests with synthetic networks, here we do have node correspondence, due to the multilayer nature of the original

Table 3.1: List of the 37 airlines in the dataset

1	Lufthansa	20	LOT Polish Airlines
2	Ryanair	21	Vueling
3	Easyjet	22	Air Nostrum
4	British Airways	23	Air Lingus
5	Turkish Airlines	24	Germanwings
6	Air Berlin	25	Panagra Airways
7	Air France	26	Netjets
8	Scandinavian Airlines	27	Transavia Holland
9	KLM	28	Niki
10	Alitalia	29	SunExpress
11	Swiss International Airlines	30	Aegean Airlines
12	Iberia	31	Czech Airlines
13	Norwegian Air Shuttle	32	European Air Transport
14	Austrian Airlines	33	Malev Hungarian Airlines
15	Flybe	34	Air Baltic
16	Wizz Air	35	Wideroe
17	TAP Portugal	36	TNT Airways
18	Brussels Airlines	37	Olympic Air
19	Finnair		

dataset. The aim of this analysis is to check on real data the behaviours we found in Section 2.3.2 about the distances independent on node correspondence and to evaluate the performances of the distances which require node correspondence.

We carried out a clustering analysis on this dataset. Firstly, we computed all the pairwise distances between all the layers using each of the aforementioned methods, obtaining nine 37×37 distance matrices. Then, we built dendrograms using agglomerative hierarchical clustering with *ward.D2* linkage [60]; we also reordered the leaves using the Optimal Leaf Ordering (OLO) algorithm [10], in order to minimize the dissimilarity between neighbour leaves. We also computed the Cophenetic Correlation Coefficient [39, 71] of such clusterings to asses their quality, and we produced heatmaps of the distance matrices to better understand the effectiveness of each network distance to identify clear and separated clusters.

3.1.2 Results

We show the resulting dendrograms in Figure 3.1. The first thing that can be noticed is that the two classes of distances produce different outputs: indeed, the Euclidean, Jaccard and DeltaCon distances do not produce any clustering of the various airlines, while the distances independent on node correspondence obtain some type of separation. Anyway, the only analysis of the dendrograms can be misleading; for this reason, we provide in Table 3.2 the Cophenetic Coefficient of the clusterings, and in Figure 3.2 the heatmaps of the distance matrices produced by each method, ordered with the same layer ordering as in the dendrograms.

We see from Table 3.2 that the lowest Cophenetic Coefficients are those of the

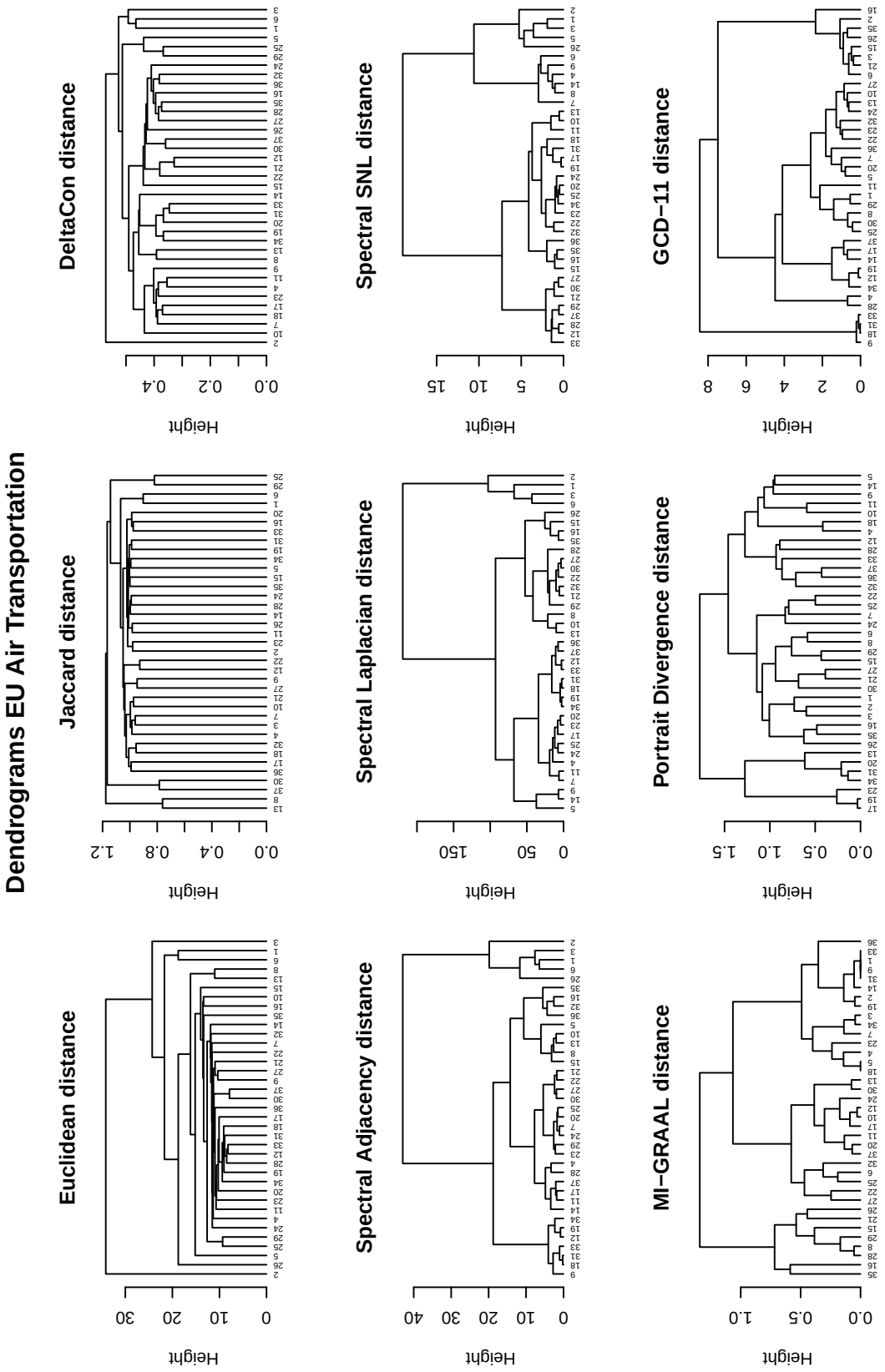


Figure 3.1: Dendrograms obtained from the clustering analysis of the EATN dataset.

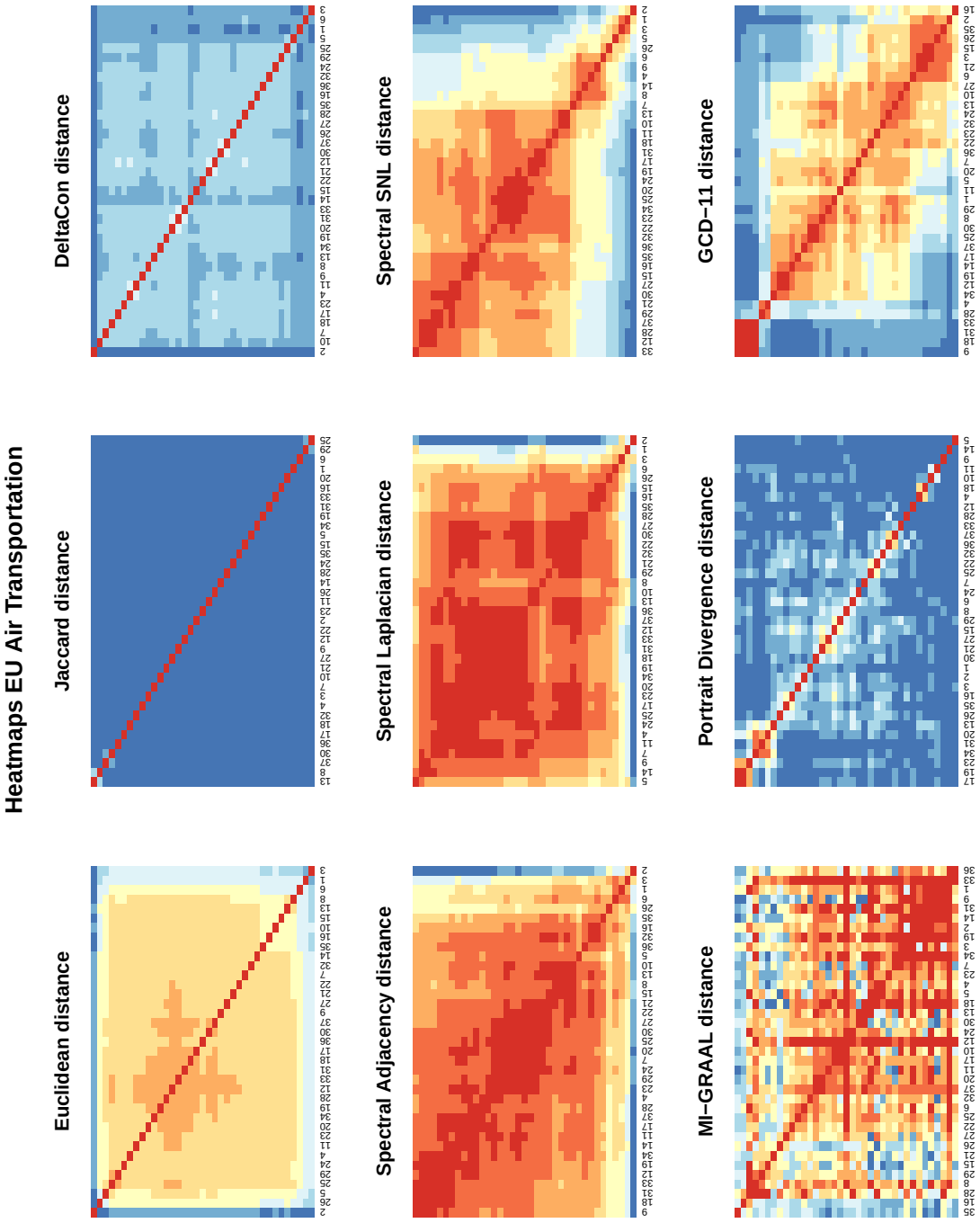


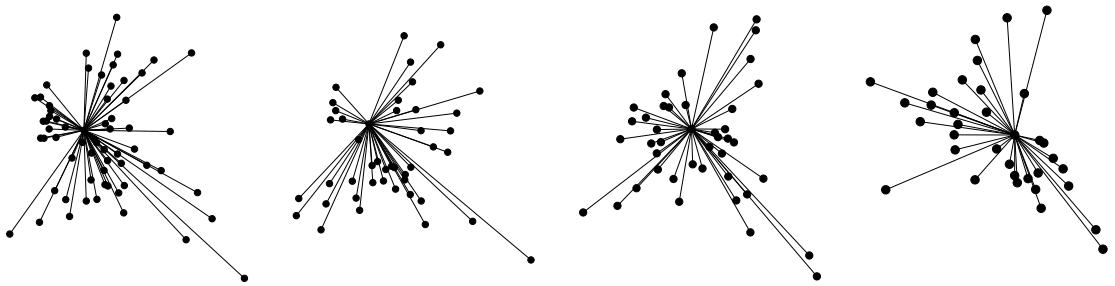
Figure 3.2: Heatmaps of the distance matrices produced by the different methods. The colour in position (i, j) represents the distance between networks i and j: cool colours (blue) mean large distances, while warm colours (red) mean small distances.

Table 3.2: Cophenetic Coefficients of the EU Air Transportation dendrograms.

Euclidean	0.9769143	Spectral SNL	0.8359300
Jaccard	0.3425482	MI-GRAAL	0.4971917
DeltaCon	0.7209673	Portrait Divergence	0.6000926
Spectral Adjacency	0.7995584	GCD-11	0.8220971
Spectral Laplacian	0.8106683		

Jaccard and the MI-GRAAL distances: this means that the clusterings shown in the dendrograms do not reflect the structure of the corresponding distance matrices and we should not rely on conclusions based on the dendrograms. On the other hand, all the other network distances have sufficiently large Cophenetic Coefficient, so that the dendrograms more accurately represent the corresponding distance matrices. The same conclusions can be drawn by looking at the heatmaps in Figure 3.2: the Jaccard distance matrix is almost completely composed of equal values, so that it is impossible to identify clusters, and the MI-GRAAL distance matrix with the ordering given by the dendrogram still remains scarcely interpretable, so that the three clusters that one identifies in the dendrogram have a really poor quality.

The analysis of the ordered distance matrices is also interesting for another reason: no distance matrix produces a clear clustering of the different airlines, with a clear separation between the groups, with the only exception of the GCD-11 distance. The Euclidean and the DeltaCon distances identify one single cluster including all the networks, except the layer corresponding to Ryanair, which is identified by both distances as the most distant layer with respect to all the others. This result agrees with intuition, since the Ryanair layer is the one which has the largest number of nodes and edges: comparing it with smaller layers always results in large distances. This also implies that distances which require node correspondence are strongly affected by size and density of the graphs that are being compared: two graphs with different sizes or densities are likely to be more distant than two graphs with similar characteristics. About the Jaccard distance, we already pointed out that all the pairwise distances have almost equal values, and no cluster is identified. The three Spectral distances behave in a very similar way. From the dendrograms we can identify two main clusters, a small one containing airlines 1, 2, 3, 6 (but not for the Spectral SNL, which instead includes airline 5) and 26 (but not for the Spectral Laplacian distance), and a larger one, containing all the other airlines. The small cluster groups together the airlines with the highest number of nodes and connections (> 75 nodes, > 110 edges). This behaviour reflects the strong dependence of the three spectral distances on size and density of the graphs, as highlighted in the synthetic tests carried out in Section 2.3.2. The heatmap of the Spectral SNL distance shows a clearer separation among the groups and identifies a third cluster, which anyway is composed of airlines apparently with no common characteristic. The heatmaps of the MI-GRAAL distance does not show any meaningful clustering; it is able to catch three (airlines 9, 31 and 33) out of four star graphs present in the dataset and to group them together at zero distance (i.e., it found a perfect matching), but it is not clear at all why it missed



(a) KLM **(b)** Brussels Airlines **(c)** Czech Airlines **(d)** Malev Hungarian Airlines

Figure 3.3: The four star graphs present in the EATN dataset.

the fourth star graph. The Portrait Divergence distance identifies only two small clusters and a third one, with all the remaining layers. The two small clusters group airlines 17, 19, 23 and 13, 20, 31, 34, and the only feature in common within each group is the maximum degree. Looking for other examples of this behaviour, we only find that airlines 4 and 18 have the same maximum degree and are paired at small distance, but this does not happen for other values of maximum degree. Looking at the heatmap, we note that all the remaining layers have large pairwise distances, so that the corresponding cluster is not significant. Moreover, this is a counterintuitive result. All the layers of this dataset are small and with a particular structure, so that their shortest path distributions are expected to be similar; thus, one would expect to obtain small, and not large, pairwise distances between the layers. For instance, the two airlines 18 and 31 have both a star structure and they differ only for two nodes in the size, but their Portrait Divergence distance (which ranges in [0,1]) is 0.9107. The last method to be analysed is GCD-11. Its heatmap confirms the subdivision in three clusters shown by the dendrogram. In particular, one cluster contains airlines 9, 18, 31 and 33, which are the four pure star layers present in the dataset (they are shown in Figure 3.3). The second cluster contains airlines 2, 3, 6, 15, 16, 21, 26 and 35: it gathers seven out of eight airlines whose layers have high clustering coefficient (> 0.15). The third cluster groups all the remaining layers. Moreover, airlines 4 and 28 are also paired at very small distance, and they both have zero clustering coefficient without being a star graph. These results clearly shows that GCD-11 is effective in discriminating networks based on purely topological features.

Previously we found that the distances which need correspondence between nodes find a unique cluster including all the layers, but nonetheless their results give some interesting insights if analysed from a different point of view. Taking a more careful look, we notice that some airlines based in the same nations or regions are paired together: taking for instance the Euclidean distance, it is the case of airlines 1 and 6 (Germany), 8 and 13 (Scandinavia), 30 and 37 (Greece), 9 and 27 (Netherlands). The same happens with the other two distances which requires node correspondence. The Jaccard distance pairs together the aforementioned

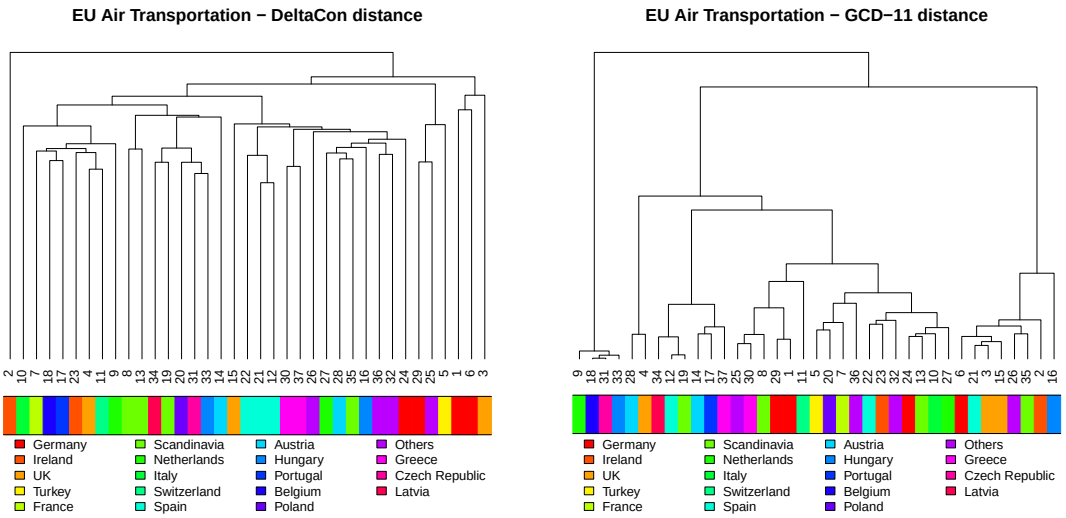


Figure 3.4: DeltaCon and GCD-11 dendrograms with the leaves coloured with respect to the nations the airlines are based in. The DeltaCon distance, which requires node correspondence, is able to group many airlines that are based in the same nation.

airlines and in addition airlines 3 and 4 (UK), 12 and 22 (Spain), 2 and 23 (Ireland), 16 and 33 (Hungary). The DeltaCon distance groups together airline 8 and 13 (Scandinavia), 12, 21 and 22 (3 out of 4 based in Spain), 30 and 37 (Greece), 1 and 6 (Germany), 24 and 29 (the other two German airlines). This kind of grouping is not recovered by any of the distances independent on node correspondence; we show as examples the cases of the DeltaCon and the GCD-11 distances in Figure 3.4. These results are possible due to the fact that networks distances which require node correspondence use similarities between nodes to compute the distance. Airlines based in the same nation share the same national airports; moreover, often the two paired airlines are the national company and a low-cost company of the same nation, so that they offer different journey and price conditions on the same routes. These two facts make the nodes of two airlines based in the same nation very similar and thus reduce the distance between them. Then, the clustering provided by these kind of distances, even if its quality is not optimal since we have many nations and few airlines per nation, can anyway be evaluated as good, since it succeeds in pairing together (i.e., at the smallest distance possible) layers whose nodes have high similarities.

3.2 FAO Trade network

3.2.1 Description of the dataset

The next case study is about the trade network of food, agricultural and animal products from the FAO database. The dataset was first used in [26] and we downloaded it from [25]. We also integrated the information about the products using the official FAO website [31]. The dataset represents the worldwide food

Table 3.3: The HS 2-digit codes classification

Description	HS Codes
Animal and Animal Products	01 - 05
Vegetable Products	06 - 15
Foodstuffs	16 - 24
Mineral Products	25 - 27
Chemicals	28 - 38
Plastics and Rubbers	39 - 40
Raw Hides, Skins and Leathers	41 - 43
Wood and Wood Products	44 - 49
Textiles	50 - 63
Footwear and Headgear	64 - 67
Stone and Glass	68 - 71
Metals	72 - 83
Machinery and Electrical	84 - 85
Transportations	86 - 89
Miscellaneous	90 - 97

import/export network (updated on 2010). It is composed by 364 layers, each one representing a different product, which share 214 nodes, representing countries. The total number of connections is 318 346. The multilayer network is directed and weighted, where the weights represent the export value, in thousands of USD, of a product from a country to another one.

The additional information we added from [31] is the Harmonized System (HS) codes [73] updated at 2007, to recover a classification for each product. In particular, we refer to the 15 sections at the top of the classification hierarchy, shown in Table 3.3. Each category contains several products; a finer classification can be made by extending the HS code to four or even six digits, with the first two digits still denoting the section the product belongs to. In particular, the products considered in this FAO dataset are identified by the HS 6-digits codes and they belong to seven out of the fifteen categories: *Animals and Animal Products*, *Vegetable Products*, *Foodstuffs*, *Chemicals*, *Plastics and Rubbers*, *Raw Hides, Skins and Leathers* and *Textiles*.

We performed some preliminary operations on the dataset. Due to the different sources of our data, we found two mismatched products in the two datasets, namely *Pigeons, other birds* and *Dregs from brewing, distillation*, so that we excluded them from the dataset. Moreover, besides the directed and weighted original data, for our analysis we also a considered directed and unweighted version of the different layers. This choice was taken to consider increasing complexity scenarios, both to provide an exhaustive description of the behaviour of the network distances also in directed and unweighted situations, and to check whether the methods gain in performances and sensitivity when considering the original directed and weighted data.

The simplification to the directed and unweighted case was made by computing two indexes. The first one is the Revealed Comparative Advantage (RCA, also

known as Balassa index) [7], which, for a fixed product, measures the relative export performance of a country, defined as the share of its own export of the considered good divided by the share of the total world export of the considered good. Denoting by $E = [e_{cp}]$ the export trade matrix where each entry e_{cp} denotes the amount of product p that country c exports to other countries, the RCA of country c on product p can be expressed as

$$RCA_{cp} = \frac{\frac{e_{cp}}{\sum_{p'} e_{cp'}}}{\frac{\sum_{c'} e_{c'p}}{\sum_{c'p'} e_{c'p'}}}. \tag{3.1}$$

In other words, the RCA compares how much a country exports a product p with respect to its total export and how much all the countries export the same product p with respect to the global world export. If RCA_{cp} is larger than 1, then country c can be considered as a large exporter of good p. We simplified our directed and weighted layers as follows: after having computed the RCA for all countries and all products in our dataset, for each layer p we kept only the nodes (i.e., countries c) with RCA_{cp} larger than 1 (i.e., the countries that are large exporters of that product) and all the nodes reached by their out-links. We then obtained a simplified multilayer network, composed of directed and unweighted layers, which describe the export relationships between countries. This simplified version of the original dataset will be referred to as the "Export" dataset in the following.

We adopted an analogous procedure to simplify the original dataset, but focusing on import relationships. An analogous index related to import can be defined just considering the import trade matrix $I = [i_{cp}]$ instead of the export trade matrix in Equation (3.1). In this case, the index compares how much a country imports a product p with respect to its total import and how much all the countries import the same product p with respect to the global world import, so that the index measures how much a country depends on a product. If the index is larger than 1, then country c is considered a large importer of that product. We can simplify the original data in an analogous way by considering for each layer only the nodes that are large importers and all the nodes that are origin of their in-links. Again, we end up with a simplified multilayer network of directed and unweighted layers, which describe the import relationships between countries. This simplified version of the original dataset will be referred to as the "Import" dataset in the following.

The original directed and weighted multilayer network will be referred to as the "directed and weighted" or the "DW" dataset in the following.

The last preprocessing step consisted in removing the layers with very few nodes or highly disconnected, to avoid considering too small graphs in the comparisons thus preventing the methods to incur numerical problems. For each multilayer network (Export, Import and DW), we excluded the layers whose largest weakly connected component contained less than 10 nodes. We then obtained an Export dataset containing 347 layers, an Import dataset containing 352 layers, and a DW dataset containing 353 layers.

Table 3.4: Ranges of size and density in the three FAO datasets

	Export		**Import**		**DW**	
	Size	Density	Size	Density	Size	Density
Min	11	0.0156170	14	0.01673451	20	0.01522956
Max	151	0.1742424	193	0.13610106	207	0.17043433

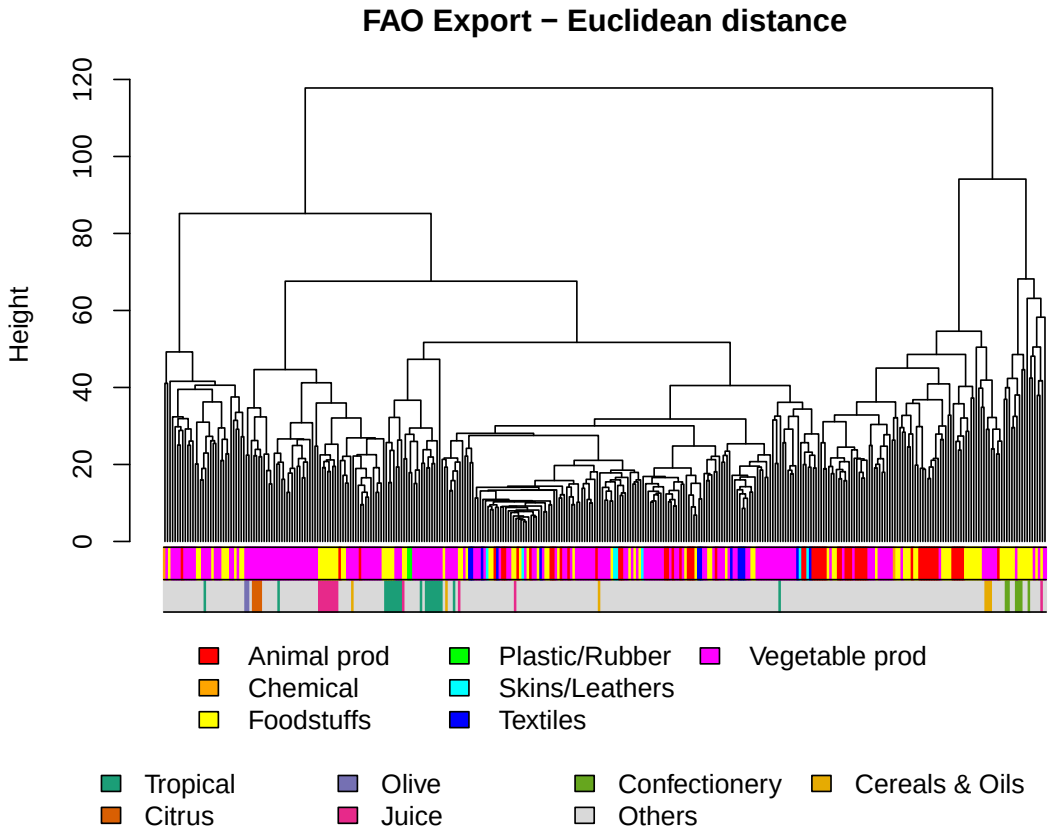
We carried out a clustering analysis on the three datasets. For the Import and Export datasets, we used the distances reported in the third row of Table 2.1, with the exception of the Canberra and Manhattan distances (it is an unweighted case, so they behave the same as the Euclidean distance) and of MI-GRAAL (due to its high computational times). For the DW dataset, we used the distances in the fourth row of Table 2.1, with the exception of MI-GRAAL (for the same reason). Note that the sizes and the densities of the layers in the three datasets have large variations (Table 3.4), so that for the Import and Export case we expect that only the DGCD-129 distance will produce reliable results among the distances independent on node correspondence, while in the DW case we have not *a priori* information.

The clustering analysis is aimed at highlighting economical aspects that can be derived from the similarity of the trade networks. For instance, it is interesting to investigate whether the clustering of networks highlights supply chains, i.e., whether the methods group at small distances layers of raw materials and layers of the final goods produced with those raw materials. Another interesting economical aspect to be analysed on these datasets is whether layers of specific products which are produced and traded by the same countries are recognized to be similar. Methods which requires node correspondence are the favourite to investigate this question. Moreover, we expect that none of the clustering produced will accurately group products belonging to the same HS category, since each category contains very different products, which may obviously be traded by different countries or may have different trade patterns.

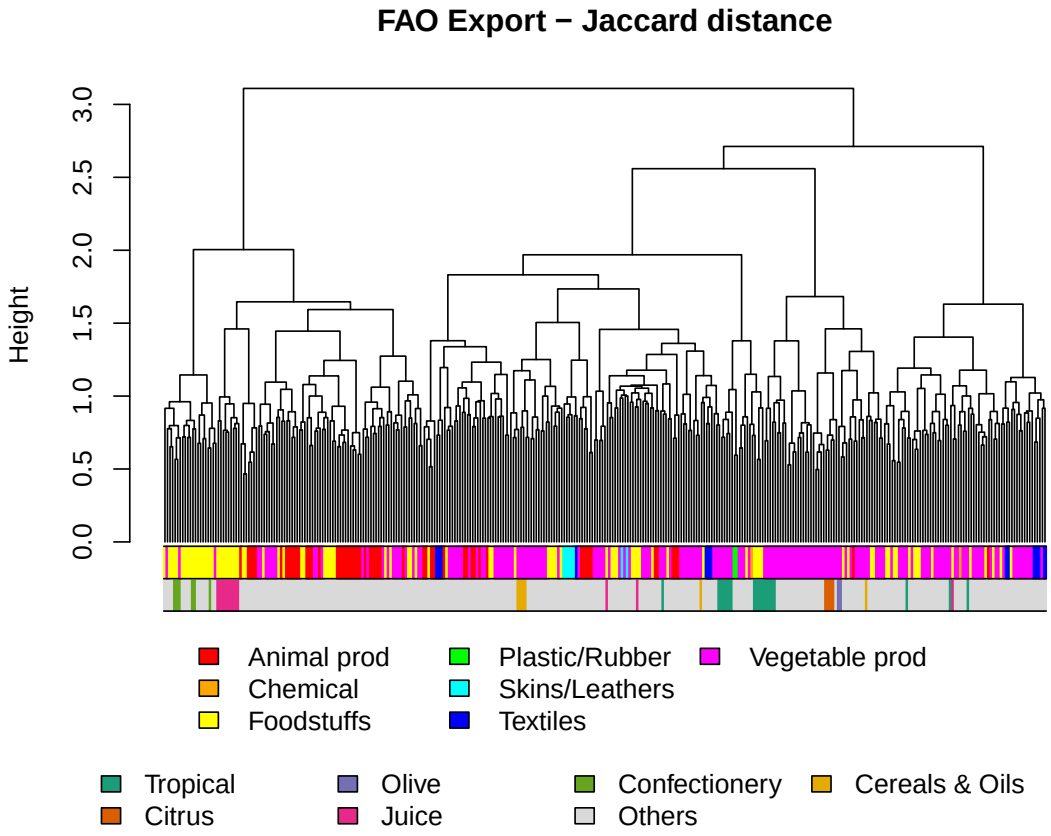
3.2.2 Results on Export and Import datasets

First of all, we present the results that we obtained from the clustering analysis of the Export and Import datasets. We show in Figures 3.5 and 3.7 the resulting dendrograms; under each dendrogram we put a coloured bar with two rows: in the first row, colours denote the membership of a leaf to the corresponding HS section, while in the second row we highlighted some groups of products that we will discuss in the following. We provide in Figures 3.6 and 3.8 the heatmaps of the distance matrices produced by each method, ordered with the same layer ordering as in the dendrograms. Finally, we also provide the Cophenetic Correlation Coefficients of such dendrograms in Table 3.5.

Differently from the European Air Transportation case study, here all the methods present a similar behaviour. Looking at the dendrograms, most of the methods identifies two clusters both in the Export and in the Import case. Also

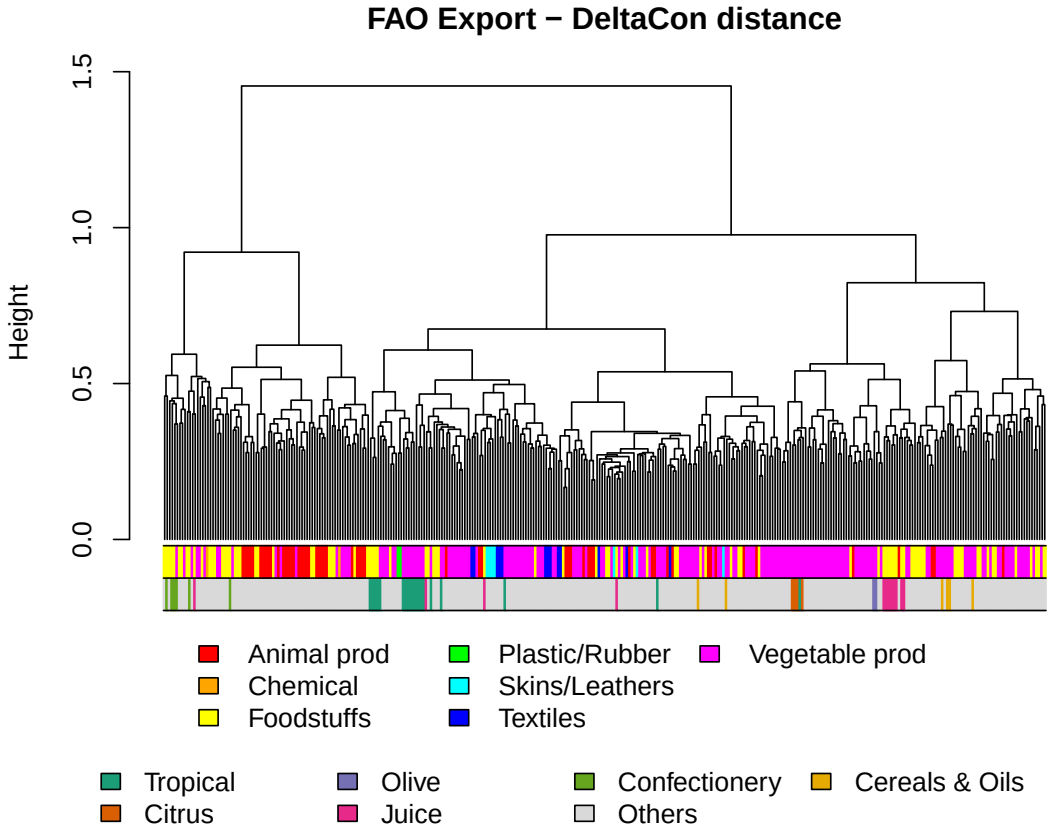


(a)

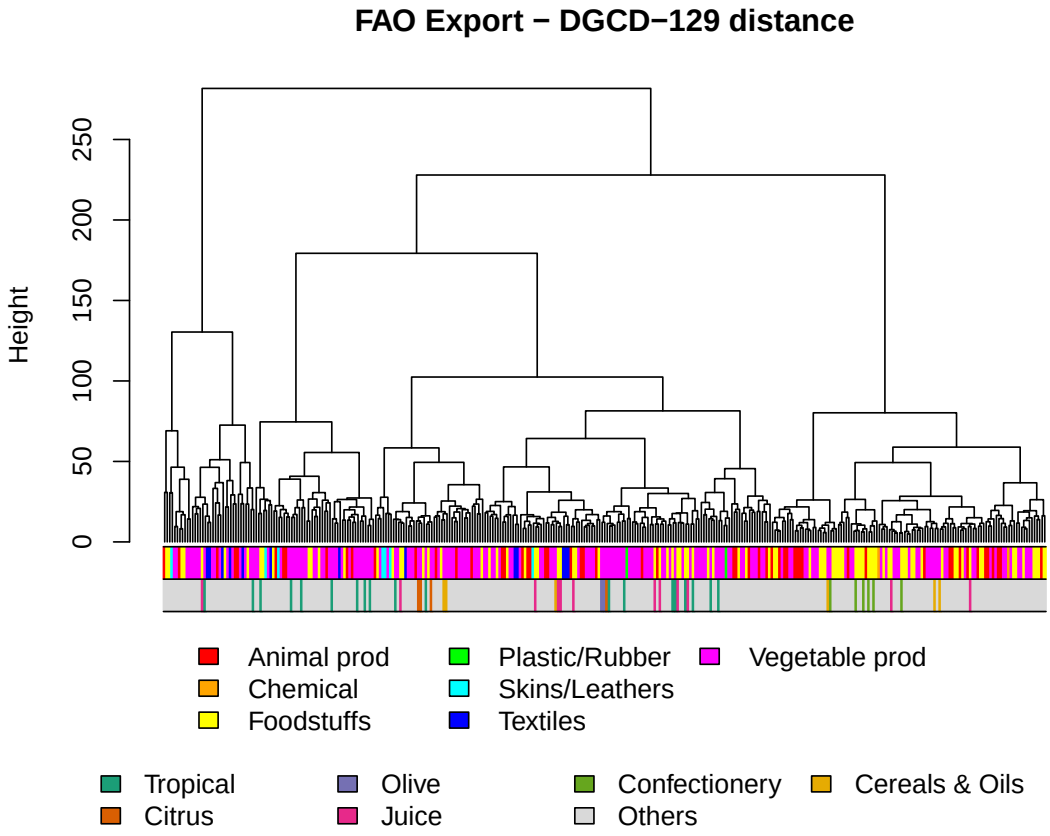


(b)

Figure 3.5: FAO Export dendrograms.

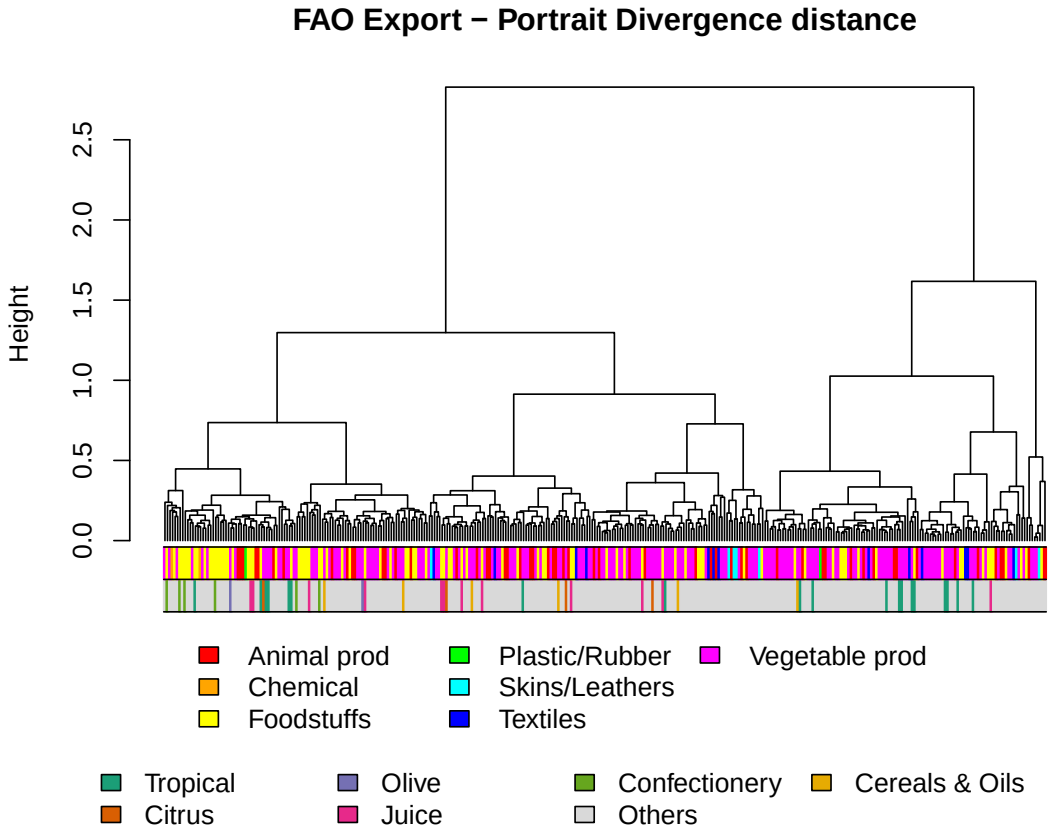


(c)



(d)

Figure 3.5 (cont.): FAO Export dendrograms.



(e)

Figure 3.5 (cont.): FAO Export dendrograms.

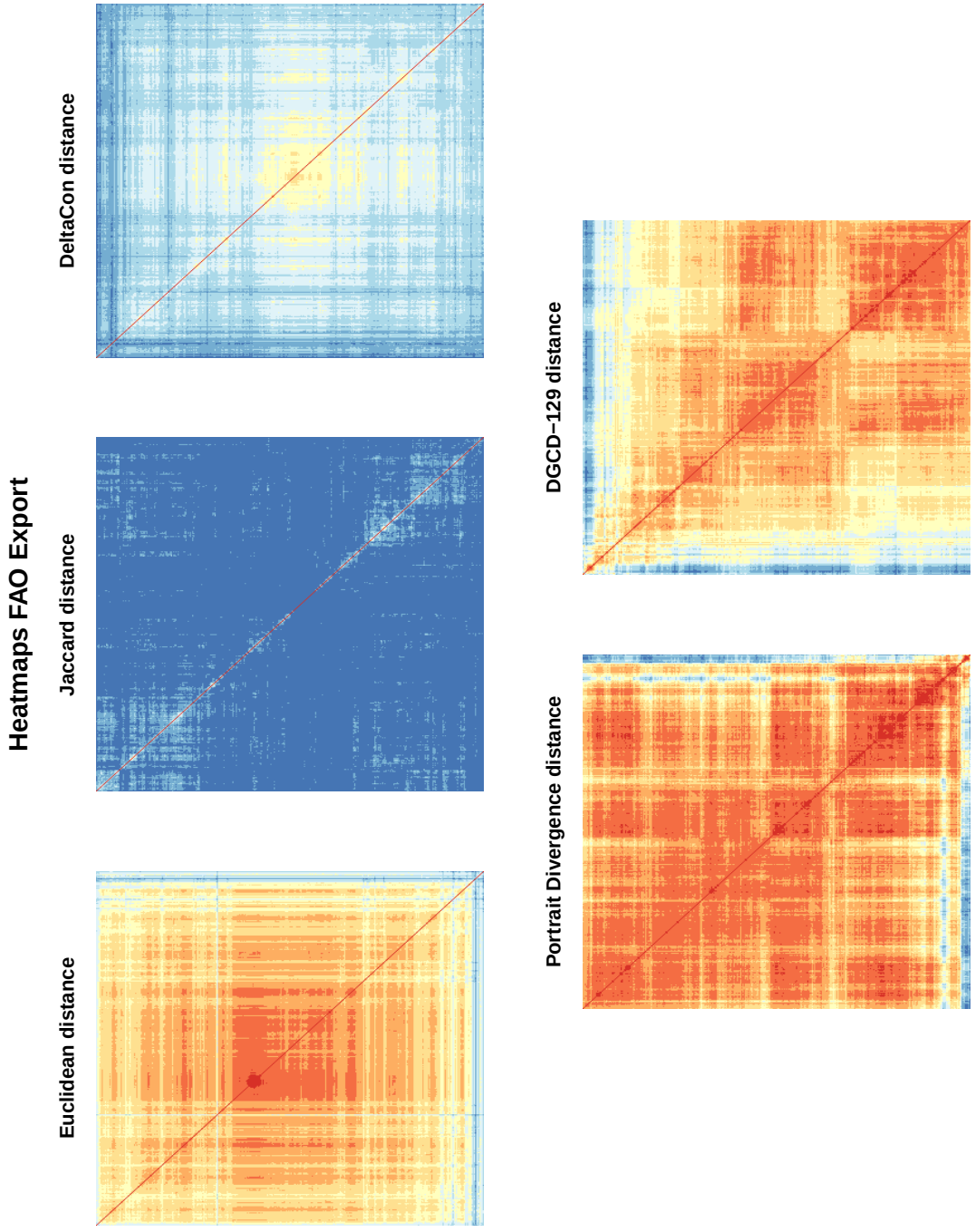
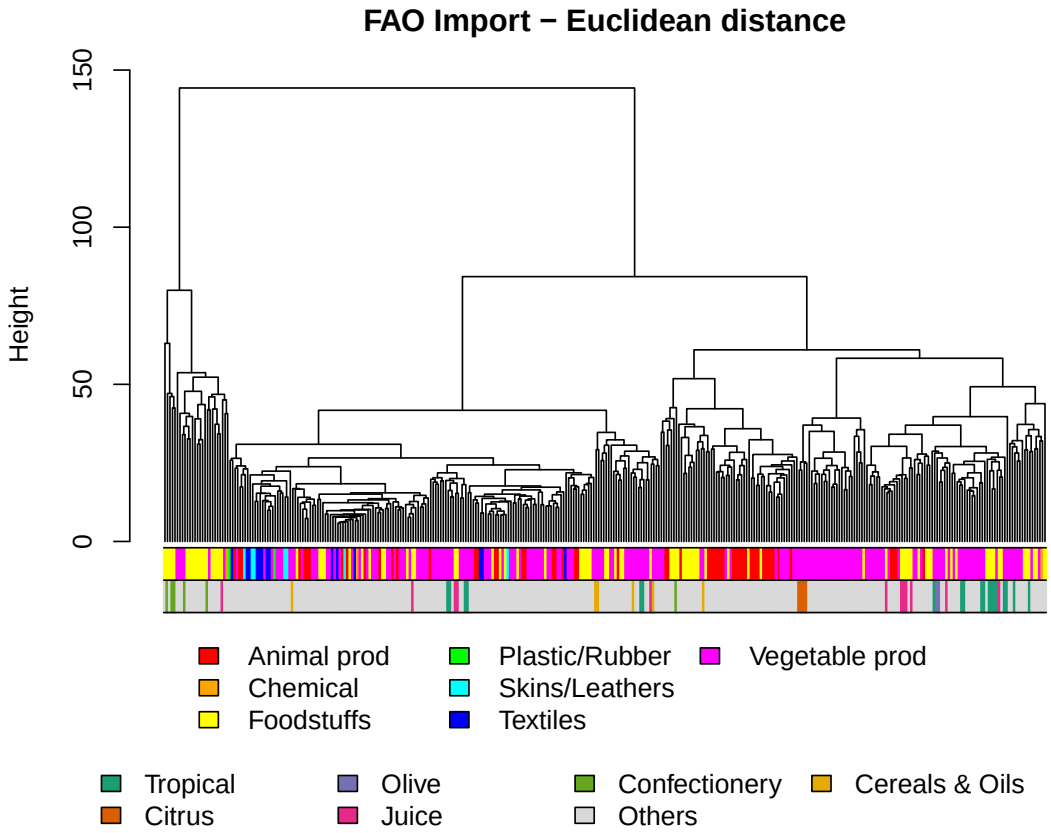
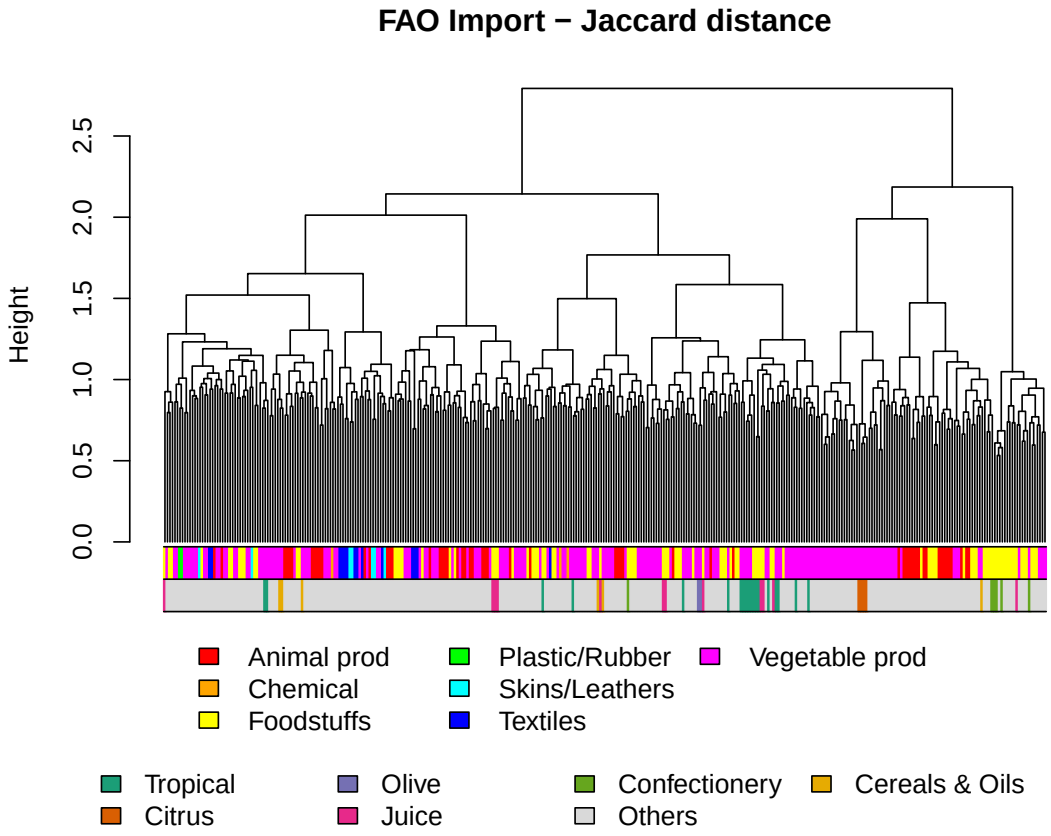


Figure 3.6: FAO Export heatmaps.

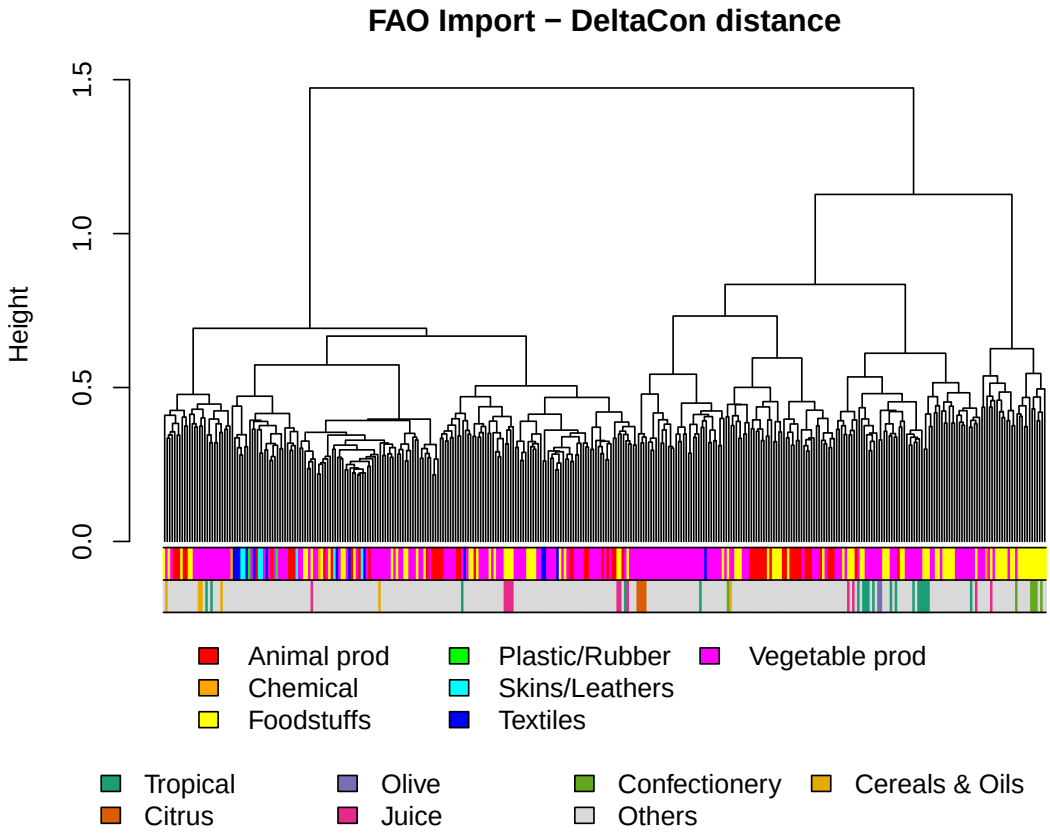


(a)

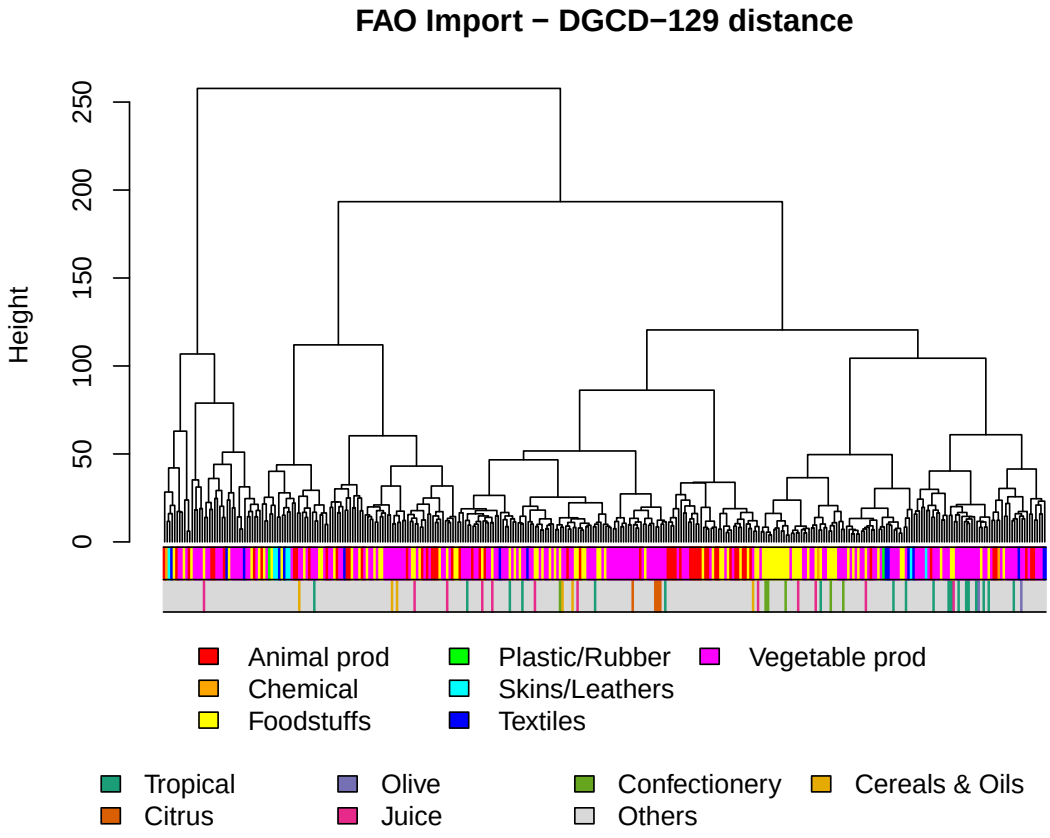


(b)

Figure 3.7: FAO Import dendrograms.

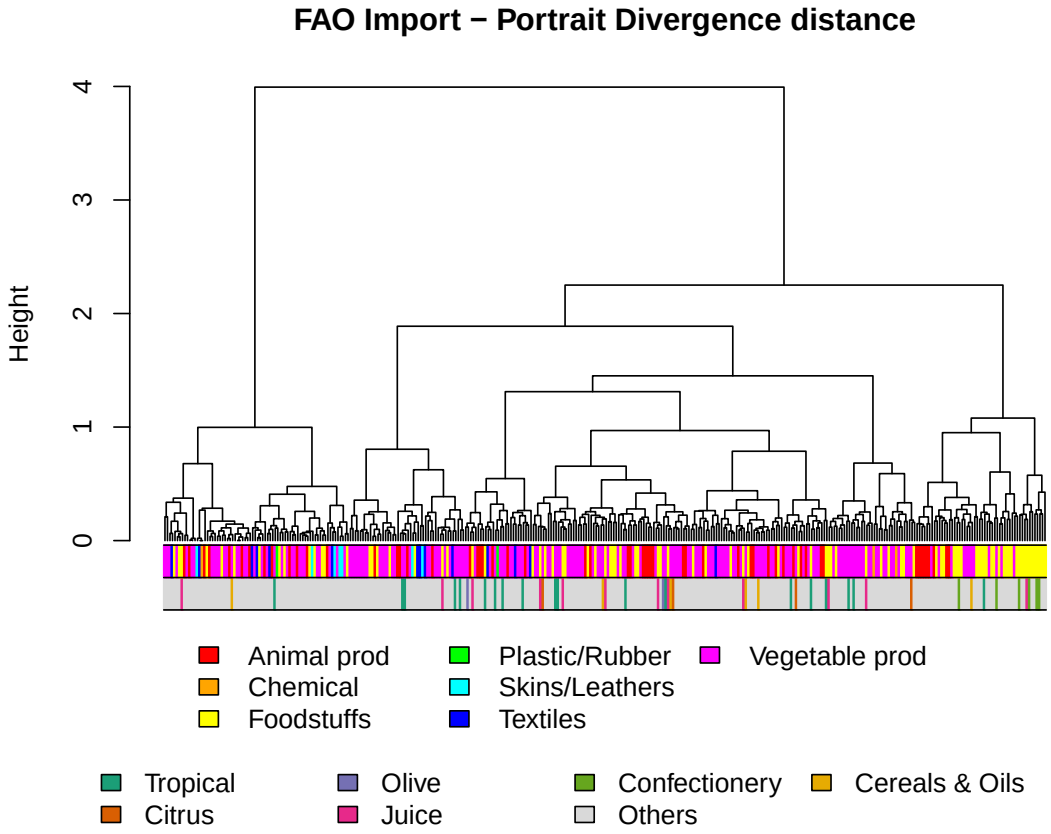


(c)



(d)

Figure 3.7 (cont.): FAO Import dendrograms.



(e)

Figure 3.7 (cont.): FAO Import dendrograms.

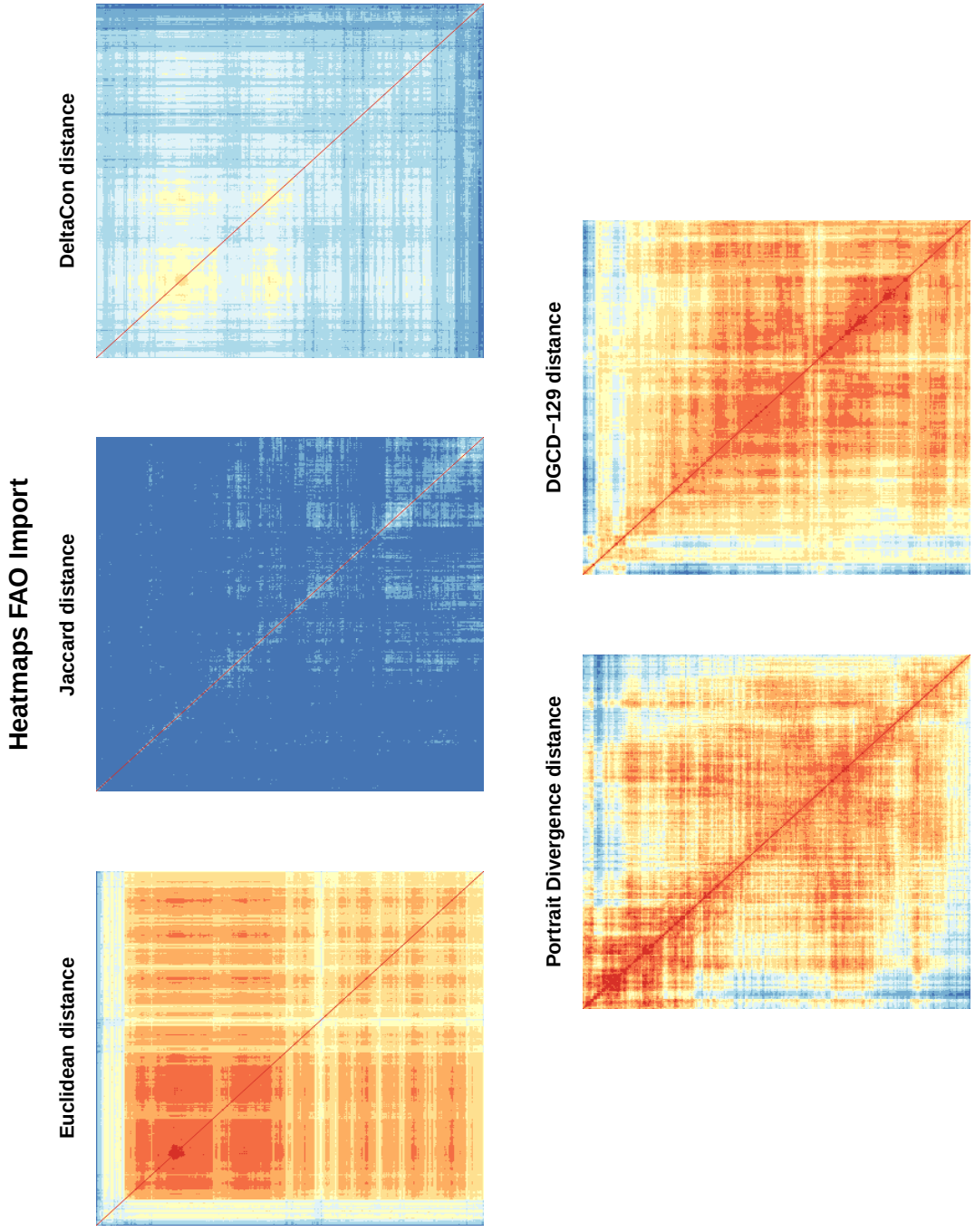


Figure 3.8: FAO Import heatmaps.

Table 3.5: Cophenetic Correlation Coefficients of the dendrograms obtained from the Export and the Import FAO datasets

Distance	**Export**	**Import**
Euclidean	0.6127916	0.7021178
Jaccard	0.3067097	0.1059323
DeltaCon	0.5244079	0.3965127
Portrait Divergence	0.4231213	0.5890709
DGCD-129	0.6528929	0.7604826

the heatmaps are similar, since most of them identifies a large group of layers which have small pairwise distances and a much smaller group of layers which are well separated from all the others. In particular, the Jaccard distance presents the same issue as in the EATN case study, i.e., all pairwise distances have almost equal value, so that it is impossible to identify a clustering. Indeed, the values of the Cophenetic Coefficient both in the Export and in the Import case are very low, denoting that the corresponding dendrograms poorly represent the distance matrices. The Portrait Divergence distance in the Export case and the DeltaCon distance in the Import case also have low values of the Cophenetic Coefficient. Note that a low value of the Cophenetic Coefficient is an issue when considering the global clustering result and when one wants to identify which elements belong to the identified clusters; but this is not really the aim of our analysis, since we are more interested in looking locally at which leaves are adjacent, without considering the global clustering, unless clearly separated clusters can be identified. Nonetheless, we prefer to exclude from the discussions the methods which attain a low value of the Cophenetic Coefficient to avoid presenting unreliable results. For this reasons, we exclude from the discussion the Jaccard distance, the Portrait Divergence distance in the Export case and the DeltaCon distance in the Import case.

Recovering the original HS sections

The first fact that can be noticed by looking at the coloured bars under the dendrograms (Figures 3.5 and 3.7) is that the distances do not recover the subdivision into HS categories neither in the Export nor in the Import case, as expected: we do not have large contiguous blocks of the same colour corresponding to well identified clusters. Even categories with few products such as *Textiles* and *Skins and Leathers*, which comprise very similar products, are fragmented. Only *Plastic and Rubber*, whose layers are *Rubber natural dry* and *Rubber, natural*, are paired together in most cases. The fragmentation of the coloured bars is much more evident in the distances independent on node correspondence.

Presence of global supply chains

The term *global supply chain* (which we will shorten in *supply chain*) denotes "a production process that is distributed over multiple countries, with production in one country providing inputs to production in another, which in turn provides

inputs to a third, and so on" [24]. The literature on this topic is wide: see for instance [8, 9, 36, 74]. Despite the sharp definition given by [24], the debate on supply chains is still open and the evidence suggest that there exists some different types of supply chains with different structures, so that it may be difficult to identify them in the data. Nonetheless, some supply chains are organized in networks of suppliers which provide inputs to produce a certain good [9], so that we should observe clusters containing raw materials, intermediate products and final goods in the data.

The search for supply chains in these FAO datasets produced no results. For instance, even products such as cereals, flour and bread are not put adjacent neither by distance which require node correspondence nor by distance independent on that and neither in the Export nor in the Import case. The same happens for other possible expected supply chains. This might be mainly due to the fact that food supply chains are difficult to be identified since they are not complex and involve only few products and processing steps; they also suffer from problems related to the production of inputs and from transportation issues, mainly due to the perishability of such products.

The only strong and meaningful relationship that we found, which however is not a supply chain, involves some confectionery products. For instance, in the Export case the Euclidean distance identifies at the rightmost part of the dendrogram a group of products (which are also the most distant from all the other layers) among which we find *Sugar refined*; *Coffe, extracts*; *Coffe, roasted*; *Chocolate products nes*; *Sugar confectionery* and *Pastry*. The same happens for the DeltaCon distance, which only misses *Sugar refined*, at the leftmost part of the dendrogram. In the Import case, we find again that the Euclidean distance groups together *Sugar confectionery*; *Pastry*; *Chocolate products nes* and *Coffee, roasted*. Interestingly, also the DGCD-129 distance in the Export case groups four out of six of those products, and the same happens to the Portrait Divergence distance in the Import case. As in many agricultural sectors, the confectionery industry is dominated by few and big companies, which determine most part of the international trade flows with their leading position in the market [74]. The similarity in the trade patterns of confectionery products may be due to the fact that these leading companies source all the different raw materials they need from the producing countries all at once, to lower the costs associated with logistics and transportations.

Identifying productions of specific products

The main part of the analysis was focused on the identification of small groups of products whose layers have large node similarities, so that the same countries have similar importance in the production and in the trading of such products. Indeed, some agricultural products (such as tropical fruits or citrus fruits) can be produced only in specific regions of the world and only few countries can produce them; thus, we expect that these countries will have large node similarities when comparing the layers corresponding to such products and thus the distance between them will be small. This analysis was obviously carried out using only the distances which require node correspondence in the Export case.

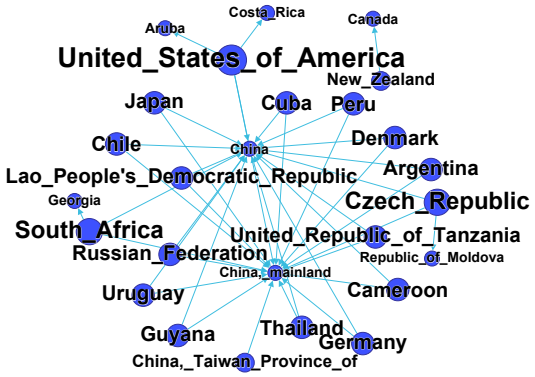
This expectation is indeed confirmed. The considered distances group together

the citrus fruits (which are *Oranges*; *Lemons and limes*; *Grapefruit (inc. pomelos)* and *Tangerines, mandarins, clementines, satsumas*) present in the dataset, and the Euclidean distance (Figure 3.5a) also puts *Olives preserved* and *Oil, olive, virgin* close to them. This is not surprising, since also olives and olive oils are produced in the same areas as citrus, where the climate is Mediterranean. Also most of the tropical fruits are gathered together. The Euclidean distance mainly finds two groups, one which contains *Cocoa, beans*; *Cocoa, powder and cake*; *Cocoa, paste*; *Cocoa, butter*; *Bananas*; *Pineapples*; *Mangoes, mangosteens, guavas*; and the other one which contains *Vanilla*; *Coconuts*; *Coconuts, desiccated*; *Cinnamon (canella)*; *Nutmeg, mace and cardamoms*; *Pepper (piper spp.)* and *Coffee, green*. The DeltaCon distance (Figure 3.5c) also identifies two major groups of tropical fruits: one contains *Cocoa, beans*; *Cocoa, powder and cake*; *Cocoa, butter*; *Cocoa, paste*; *Vanilla*; the other one contains *Coconuts*; *Coconuts, desiccated*; *Cinnamon (canella)*; *Nutmeg, mace and cardamoms*; *Pepper (piper spp.)*; *Coffe, green*; *Bananas*; *Pineapples*; *Mangoes, mangosteens, guavas*. Moreover, we found another group of specific products gathered together by these distances, which contains juices, and in particular juices made with citrus fruits. The DeltaCon distance groups *Juice, citrus, concentrated*; *Juice, grape*; *Juice, grapefruit, concentrated*; *Juice, grapefruit*; *Juice, orange, single strength*; *Juice, pineapple* and *Juice, tomato*; the Euclidean distance also groups *Juice, citrus, single strength* in addition to them. Interestingly, the groups of citrus fruits and citrus juices are not adjacent to each other and denotes the fact that often fresh fruit is not processed in the same country where it is produced.

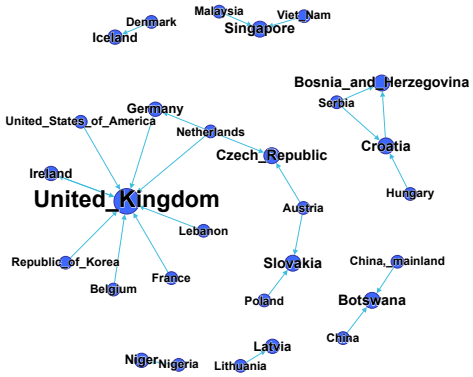
Trade structures

We used the distances which are independent on node correspondence to highlight differences in the trade patterns of some products. In particular, both in the Export and in the Import case the DGCD-129 distance identifies some layers as the most distant ones from all the others (see Figures 3.6 and 3.8), which correspond to the layers at the leftmost part of the related dendrograms (Figures 3.5d and 3.7d), while all the other layers are gathered in a large cluster. The first group is composed of products whose layers have a pronounced star-like structure with few hubs, and the second larger group is composed of all the remaining layers, which have a core-periphery structure with a dense core. In the Export case, the products identified as the most distant from all the others are: *Camels*; *Bulgur*; *Maple sugar and syrups*; *Skins, sheep, wet salted*; *Margarine, liquid*; *Pyrethrum, dried*; *Beehives*; *Feed supplements*; *Cake, copra*; *Fat, cattle* and *Fruit, tropical fresh nes*. Some products are very peculiar and are produced and exported by only one country (such as *Maple sugar and syrup*, see Figure 3.9c) or are greatly requested only by few countries (such as *Beehives*, see Figure 3.9a, and *Camels*), and thus their trade patterns are highly centralized. In the Import case, the products are: *Beehives*; *Oil, citronella*; *Skins, sheep, wet salted*; *Manila fibre (abaca)*; *Cake, groundnuts*; *Asses*; *Sugar beet*; *Copra*; *Bulgur*; *Oil, rice bran* and *Camels*. Note that in the Import case some of these layers also have a fragmented structure, such as *Sugar beet* (see Figure 3.9b), in the sense that they are composed by many and small disconnected components. Also the Portrait Divergence distance in the Import case

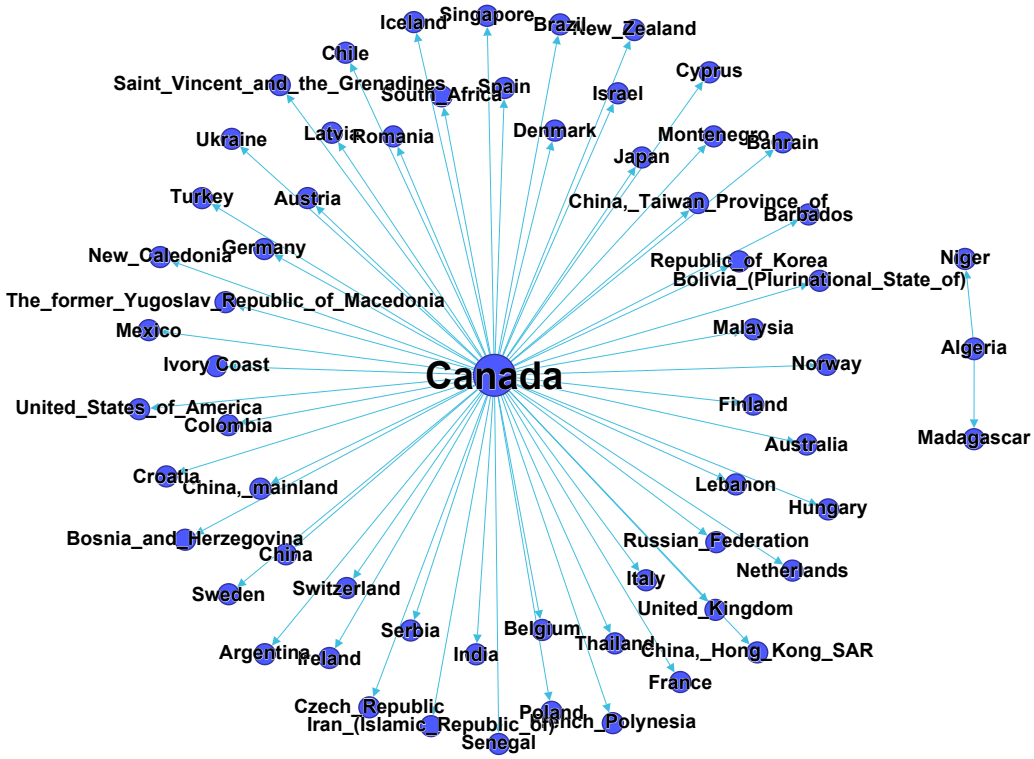
groups together many of the same products we listed above at the leftmost part of the dendrogram (Figure 3.7e), even if from the heatmap (Figure 3.8) they do not seem to be well separated from all the other layers. These products are: *Margarine, liquid*; *Cashew nuts, with shell*, *Rice, milled/husked*; *Wool, greasy*; *Feed supplements*; *Hay (unspecified)*; *Beverages, fermented rice*; *Juice, lemon, concentrated*; *Rabbits and hares*; *Maize, green*; *Camels*; *Oil, rice bran* and *Pyrethrum, dried*. All these layers also have a pronounced star-like structure.



(a) Beehives, Export



(b) Sugar Beet, Import



(c) Maple sugar and syrups, Export

Figure 3.9: Examples of star-like networks in the FAO datasets.

We also looked for some particular exceptions in the trade structure of some specific products, which we expect to have a common trade pattern. We chose six base products which are worldwide spread, which can be grown at different latitudes and which are basic food in many countries. These products are *Rice, paddy*; *Oil, sunflower*; *Oil, maize*; *Oil, soybean*; *Maize* and *Wheat*. We used the DGCD-129 distance to check whether the corresponding layers have some structural similarity, as we would expect. In the Export case, we found that two couples of such products which are adjacent or very close: the first couple is *Rice, paddy* and *Oil, soybean*, and the second couple is *Oil, sunflower* and *Wheat*. The two remaining products are far from the others, and also the two couples are far from each other. In the Import case, we again found two couples; the first one is *Oil, maize* and *Oil, sunflower* and the second one is *Wheat* and *Maize*. Again, the two couples are far from each other.

3.2.3 Results on Directed and Weighted dataset

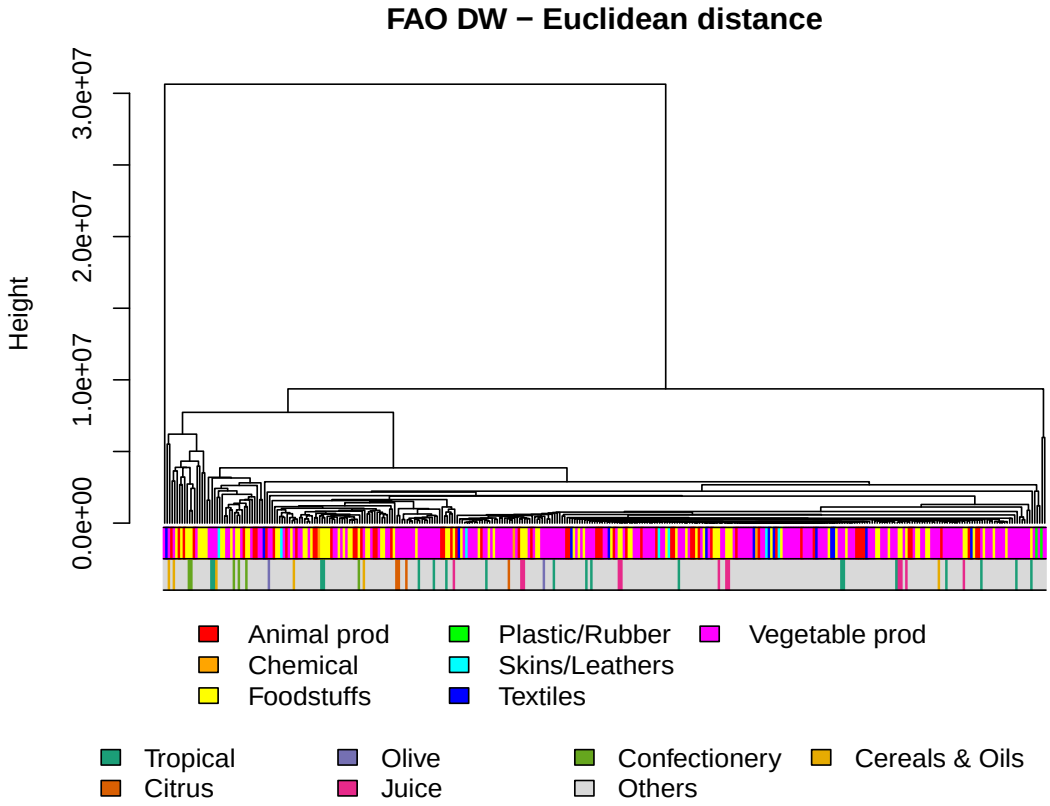
We show the dendrograms obtained from the clustering analysis of the DW dataset in Figure 3.10 and the corresponding values of their Copehenetic Coefficient in Table 3.6; the heatmaps of the corresponding distance matrices are shown in Figure 3.11. The heatmaps show a different behaviour between the distances which require node correspondence and those independent on it. In the first case, almost all the entries of the distance matrices have similar values, except for the Canberra distance, and thus they identify a unique cluster. All distances have good values of the Cophenetic Coefficient, except the Jaccard distance, which still has almost equal values of all the pairwise distances. For this reason, we will not discuss it. The heatmap of the Portrait Divergence distance, instead, does not show a clear clustering but its dendrogram has a large enough value of the Cophenetic Coefficient.

Recovering the original HS sections

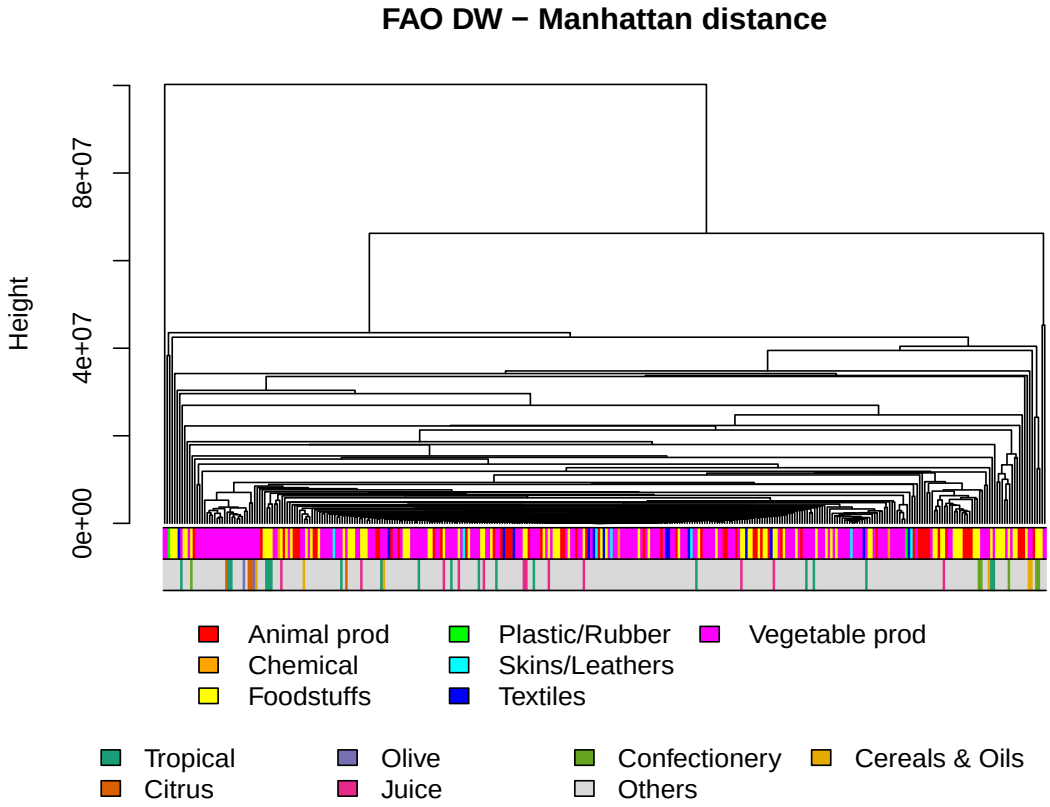
As we can see by looking at the first row of the coloured bars under the dendrograms, none of the distances is able to recover the original subdivision into HS sections. The results are even more fragmented than in the Export or Import case. Only the Canberra distance shows large contiguous blocks of the same colours, but these blocks do not correspond to well defined clusters.

Table 3.6: Cophenetic coefficients of the dendrograms obtained from the DW dataset

Distance	**DW**
Euclidean	0.9354494
Manhattan	0.9625366
Canberra	0.9625366
Jaccard	0.2270193
Portrait Divergence	0.5492191

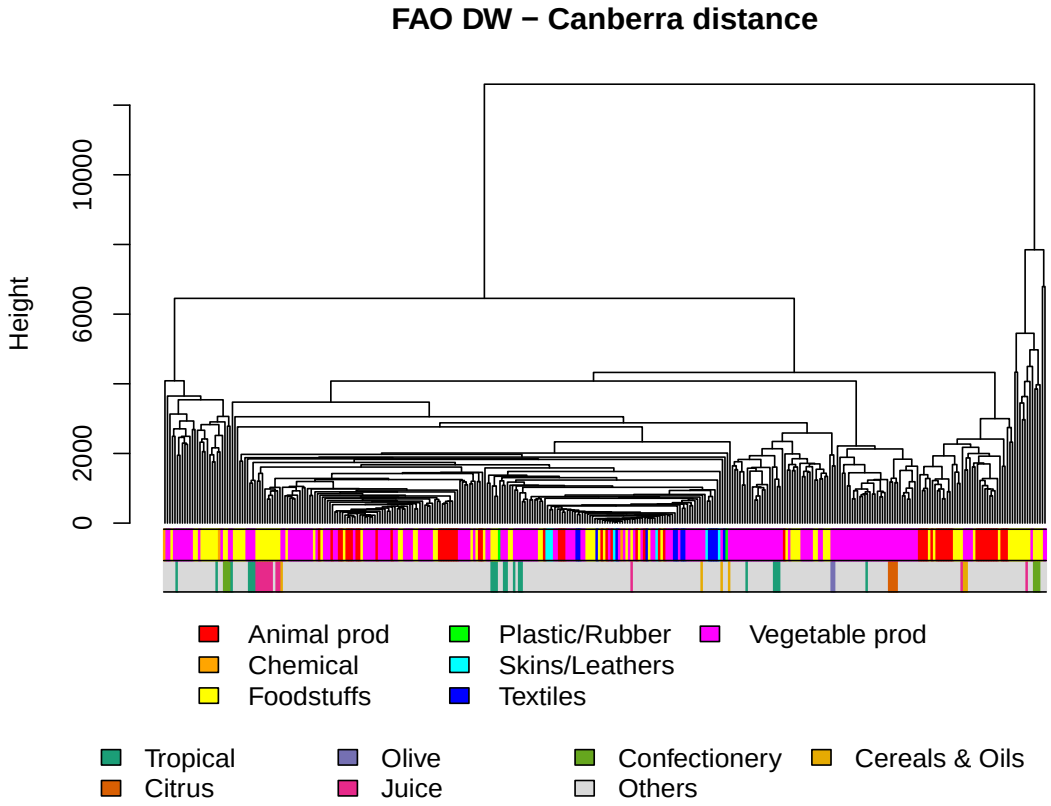


(a)

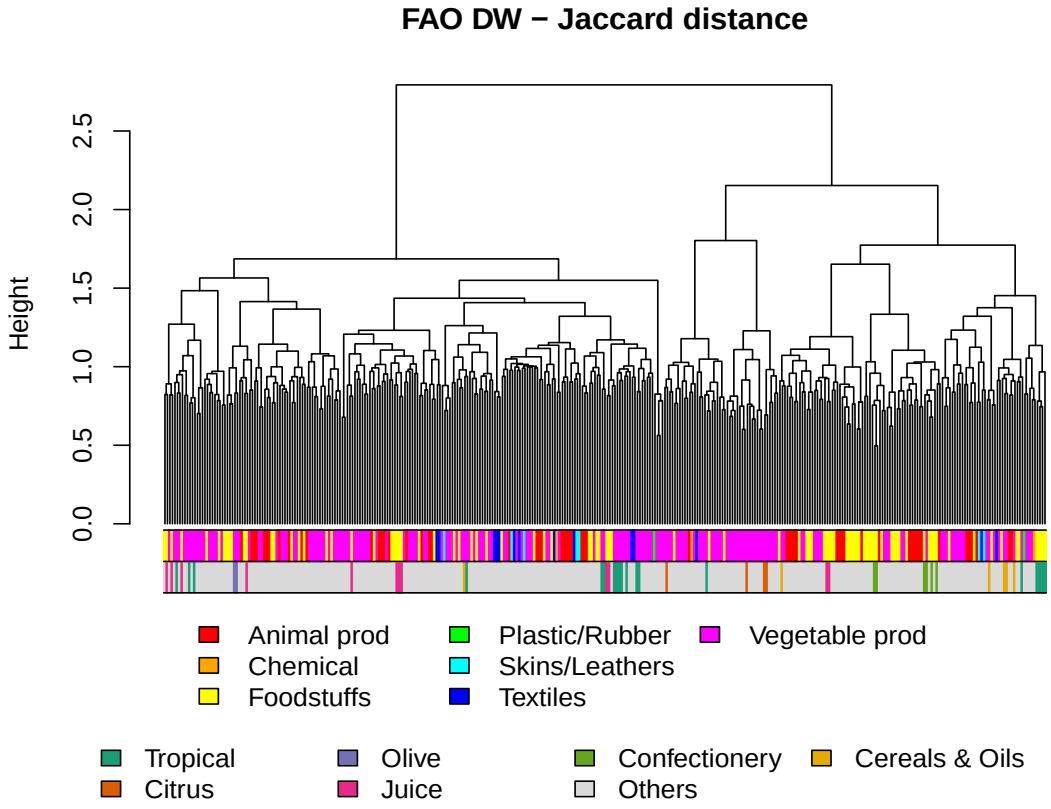


(b)

Figure 3.10: FAO DW dendrograms.

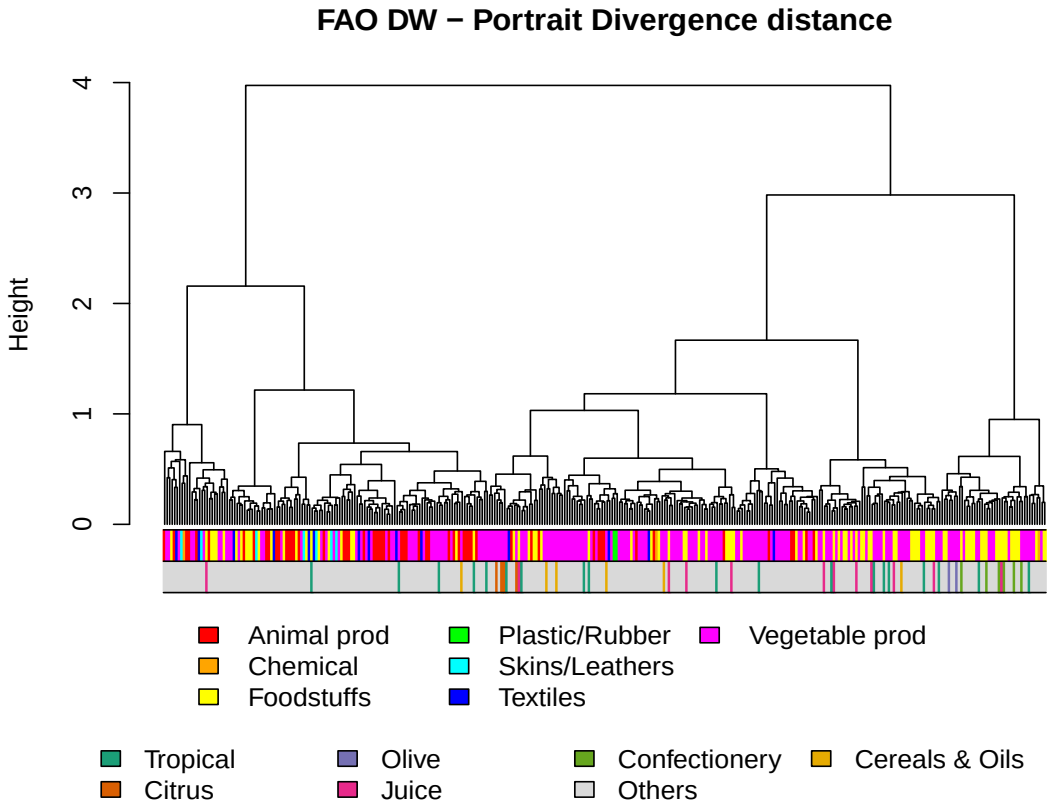


(c)



(d)

Figure 3.10 (cont.): FAO DW dendrograms.



(e)

Figure 3.10 (cont.): FAO DW dendrograms.

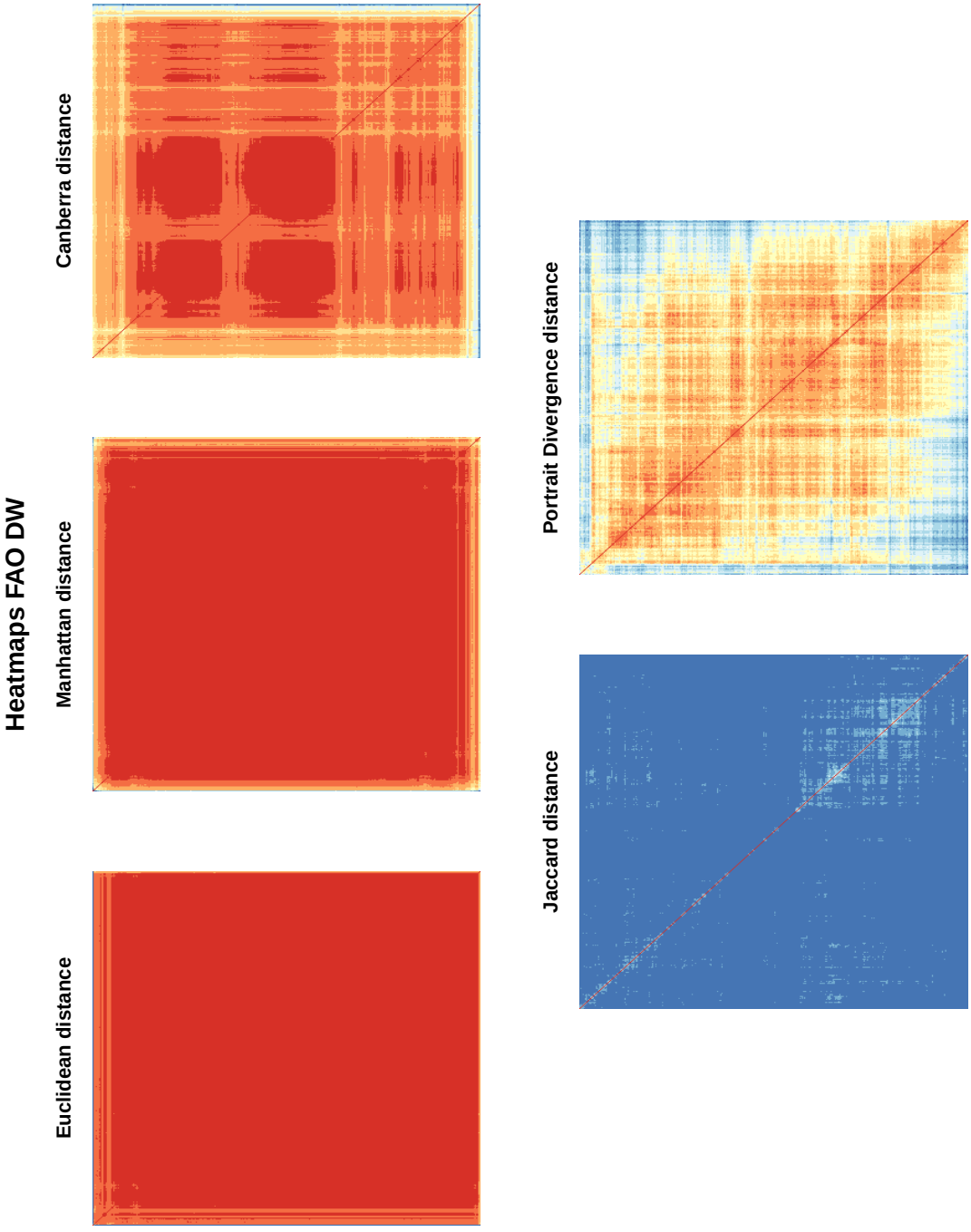


Figure 3.11: FAO DW heatmaps.

Euclidean and Manhattan distances

The Euclidean and the Manhattan distances show a peculiar behaviour, as we already pointed out. They clearly separate one single layer, which is *Soybeans* in both cases, from all the others and group almost all the remaining layers in one large cluster. We also looked at those few layers that, from the heatmaps, appear to be separated from the large contiguous red blocks. For the Euclidean distance, these layers correspond to the first small cluster on the left of the dendrogram, which includes: *Cotton lint*; *Maize*; *Meat, pork*; *Wheat*; *Cigarettes*; *Meat, chicken*; *Sugar raw centrifugal*; *Meat, cattle, boneless (beef & veal)*; *Food prep nes*; *Pastry*; *Chocolate products nes*; *Cheese, whole cow milk*; *Crude materials*; *Cake, soybeans* and *Wine*. For the Manhattan distance, those products are not grouped in a single cluster, but are located at the rightmost and leftmost part of the dendrogram; we considered the first ten layers starting from the left (excluding *Soybeans*) and the first ten layers starting from the right, since, thanks to Optimal Leaf Ordering, leaves in these positions are the most distant from leaves in the middle of the dendrogram. The products we found are: *Oil, palm*; *Rubber natural dry*; *Wine*; *Beverages, distilled alcoholic*; *Cigarettes*; *Cotton lint*; *Coffee, green*; *Sugar raw centrifugal*; *Tobacco, unmanufactured*; *Rice, milled*; *Meat, chicken*; *Meat, cattle, boneless (beef & veal)*; *Cake, soybeans*; *Maize*; *Wheat*; *Cheese, whole cow milk*; *Chocolate products nes*; *Pastry*; *Food prep nes* and *Crude materials*. Note that all the products found in the Euclidean case are present also in the Manhattan case. We further investigated this similar behaviour, and we found that the two distances mainly group networks according to their total edge weight. In fact, *Soybeans* is the layer that has the largest trade volume, equal to 69 billions USD; all the layers separated from the rest of the products by both distances have a trade volume between 14 and 44 billions USD. There are other few layers with a trade volume in the region of 10 billion USD, but most part of the remaining products has a trade volume in the region of 1 billion USD or lower. This behaviour shows a clear dependence of the Euclidean and the Manhattan distances on the edge weights, which is more pronounced when the weights vary of several orders of magnitude, as happens in our case. This leads to the grouping of networks which have comparable total edge weight. As a side effect, we observe that these two distances no longer group together the categories of specific products that we identified for the Export and Import cases, namely juices, citrus fruits and tropical fruits and spices (Figures 3.10a and 3.10b), since they have different trade volumes. Also the confectionery products, that in the Export and in the Import case were adjacent, are now distant and more spread all over the dendrogram.

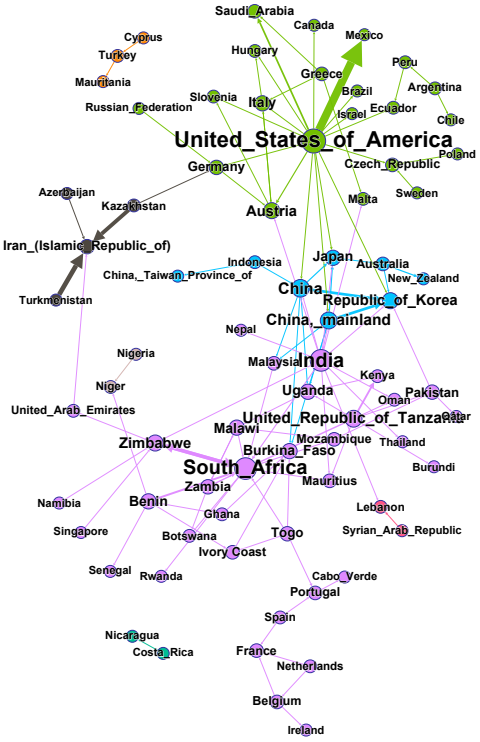
Canberra distance

The Canberra distance, instead, shows a behaviour different than the other distances which require node correspondence, since it it does not groups networks according to their trade volume. In fact, looking at the second row of the coloured bars in Figure 3.10c, we see that the Canberra distance again puts together the citrus fruits and the juices, as in the Export case. Only the tropical fruits are more spread, but anyway they form at least three meaningful groups. In the first

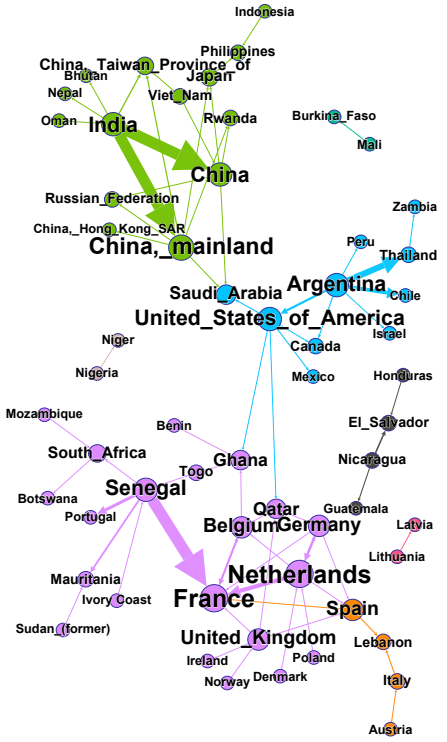
group we find *Bananas*; *Mangoes, mangosteens, guavas* and *Pineapples*, which are only tropical fruits. In the second group we find *Cocoa, beans*; *Cocoa, butter* and *Cocoa, paste*, which are only products related to cocoa. In the third group we find *Ginger*; *Nutmeg, mace and cardamoms*; *Cinnamon (canella)* and *Anise, badian, fennel, coriander*, which are only spices. The confectionery products, instead, are gathered into two separated groups: one contains *Sugar refine*; *Coffee, extracts* and *Coffee, roasted* and the other one contains *Sugar confectionery*; *Chocolate products nes* and *Pastry*.

Trade structures

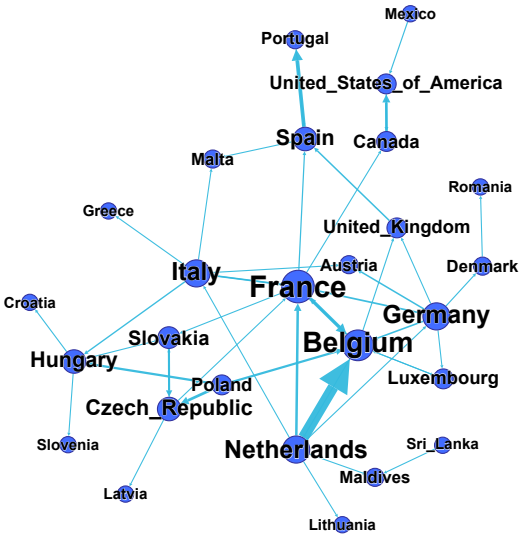
As far as the distances independent on node correspondence are concerned, we analysed the results provided by the Portrait Divergence distance. The methods required a binning strategy to group the real-valued weights into intervals and we left the method to use the default binning strategy for each pairwise comparison, due to the different orders of magnitude in the values of the edge weights. The Portrait Divergence distance does not produce a clear clustering but identifies a small group of layers, at the leftmost part of the dendrogram, which are the most separated from all the others. These products are: *Camels*; *Copra*; *Hay (unspecified)*; *Bulgur*; *Asses*; *Manila fibre (abaca)*; *Oil, rice bran*; *Skins, sheep, wet salted*; *Oil, citronella*; *Beehives*; *Rabbits and hares*; *Sugar beet*; *Canary seed*; *Goats*; *Silk-worm cocoons, reelable*; *Skins, goat, wet salted*; *Pyrethrum, dried*; *Juice, lemon, concentrated*; *Ducks*; *Cake, cottonseed*; *Cake, groundnuts*; *Meat, beef and veal sausages*; *Rice, milled/husked*; *Margarine, liquid*; *Feed supplements* and *Fat, cattle*. Many of these products are the same star-like layers which we identified in the Export or Import case, like *Canary seed* and *Pyrethrum, dried*; but we also find layers with a different structure. For instance, *Cake, cottonseed* and *Cake, groundnuts* have a more defined community structure, while other layers, like *Rabbit and hares* and *Skins, goat, wet salted*, does not have neither a star-like nor a community structure, but instead have a more pronounced core-periphery structure, with a dense core. We show the first two layers in Figures 3.12a and 3.12b, where we coloured the nodes according to the partition given by modularity optimization; the other two layers are shown in Figures 3.12c and 3.12d. Overall, it is not clear whether the similarity between graphs with strongly different structures has an economical interpretation, although not clear at the moment, due to the fact that edge weights are taken into account, or whether it is a clue that the method does not perform well in discriminating between different topologies in the directed and weighted scenario.



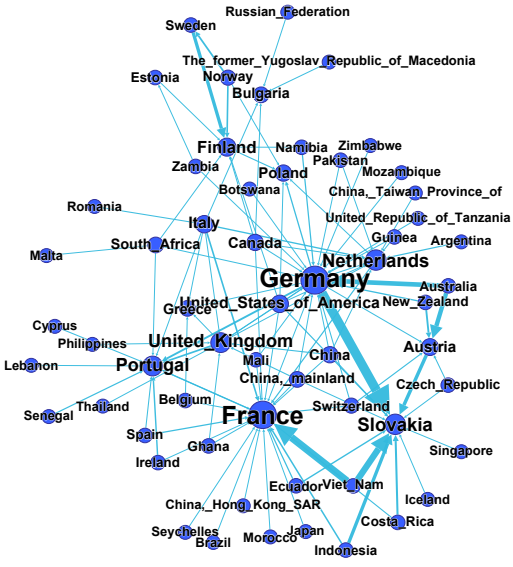
(a) Cake, cottonseed, DW (0.727)



(b) Cake, groundnuts, DW (0.654)



(c) Rabbit and hares, DW (0.365)



(d) Skins goat wet salted, DW (0.147)

Figure 3.12: Examples of directed and weighted networks. The values in parenthesis represent the maximum modularity of the networks.

3.3 World Trade Network

3.3.1 Description of the dataset

The third case study we analysed is the worldwide import/export multilayer network which comprises all the products of the HS 4-digit classification. The dataset was collected from [15] and it is composed of 1 242 layers (i.e., the products), 223 nodes common to all layers (representing the countries) and a total of 17 658 660 431 connections. All the layers are directed and weighted and the weights represent the export values (in thousands of USD) of a product from a country to another one.

Since the classification given by the HS 4-digit codes includes too many products, which are often very specific, we preferred to start our analysis by considering the classification given by the HS 2-digit codes. We recovered this division just by summing up the weight matrices of the HS 4-digit products which had in common the first two digits of the code. We ended up with a multilayer network with 96 layers, labelled using the 15 HS sections described in Table 3.3. In the following, we will always consider the HS 2-digit multilayer network unless otherwise stated.

We carried out the analysis as we did for the FAO case study, i.e, by considering the original directed and weighted ("DW") dataset and two simplified versions of it. We produced the two directed and unweighted versions of the dataset, namely "Export" and "Import", using the same procedure as in the FAO case study. We computed the RCA and the analogous index for import for all countries and all products, and simplified all the layers keeping only the large exporters and all the nodes to which they exported for the Export case, and all the large importers and all the nodes from which they imported for the Import case. We checked the presence of too small or too fragmented graphs, but we did not find any layer with largest weakly connected component smaller than 10 nodes. We adopted the same procedure on the HS 4-digits dataset, to have comparable results. In this case, we removed from the Export and Import datasets 20 products whose largest weakly connected component was smaller than 10 nodes.

We performed a clustering analysis on the three datasets, using the same distances we used for the FAO case study. The range of densities and sizes of the three datasets is reported in Table 3.7. The size does not change too much from case to case, while the density varies considerably. This means that again DGCD-129 will presumably be the most reliable measure among distances independent on node correspondence in the Import and Export cases, while we have more uncertainty in the DW case.

With the clustering analysis, we want again to highlight economical aspects

Table 3.7: Ranges of size and density in the three WTN datasets

	Export		**Import**		**DW**	
	Size	Density	Size	Density	Size	Density
Min	115	0.08181565	84	0.09610164	175	0.06009881
Max	220	0.35562474	220	0.35562474	220	0.35562474

emerging from similarity between layers. Also in this case, we do not expect that the methods will be able to group together layers belonging to the same category, because some categories contain very different products, which may have dissimilar trade patterns. To investigate this, we analysed the HS 4-digit products of some categories, to better understand how much products labelled in the same HS section are different and how they are grouped inside the category. We also looked for the presence of supply chains, and unlike the FAO case study we expect to find some, since the considered products are more complex than the FAO ones.

3.3.2 Results on Export and Import datasets

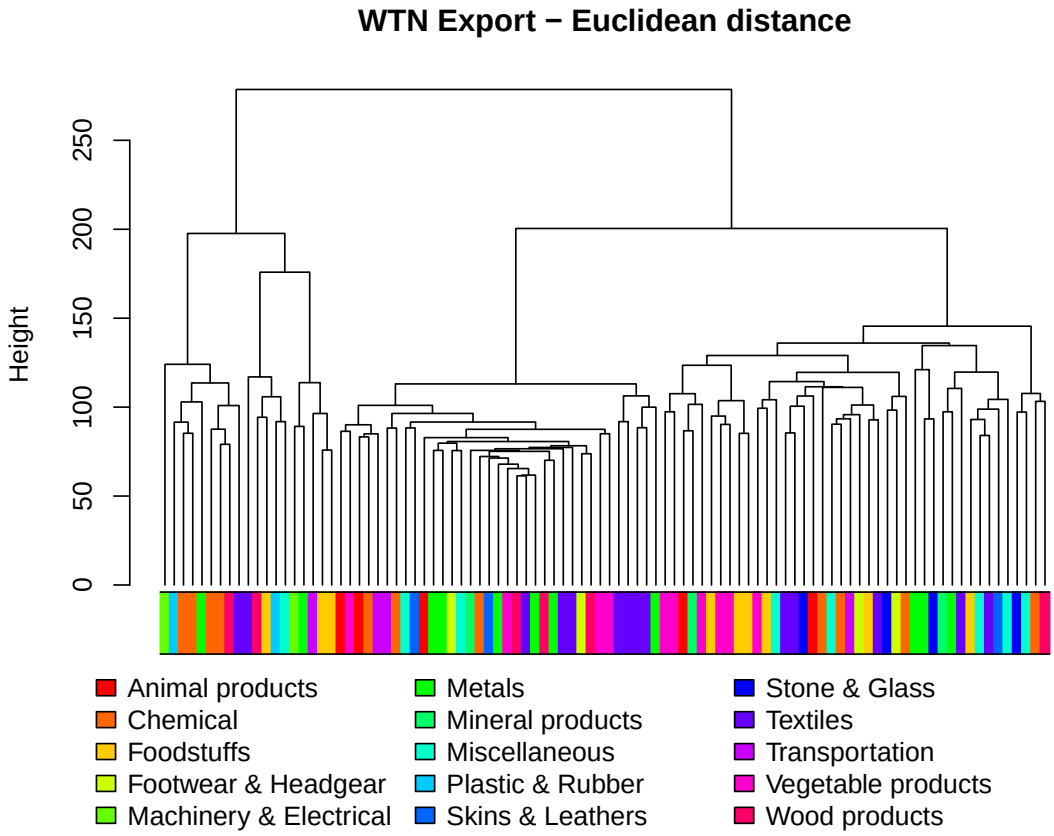
We present in Figures 3.13 and 3.15 the dendrograms that we obtained from the clustering analysis. We coloured the leaves according to the HS section the corresponding product belongs to. We also provide the Cophenenetic Coefficients of the dendrograms in Table 3.8 and the heatmaps of the distance matrices in Figures 3.14 and 3.16. We see that the values of the Cophenetic Coefficients are not high, but still acceptable for all the distances, except for the Jaccard distance. We again exclude it from the discussion. Looking at the heatmaps, instead, it can be noticed that in most cases all the pairwise distances are very similar and a clear grouping is difficult to be identified.

Recovering the original HS sections

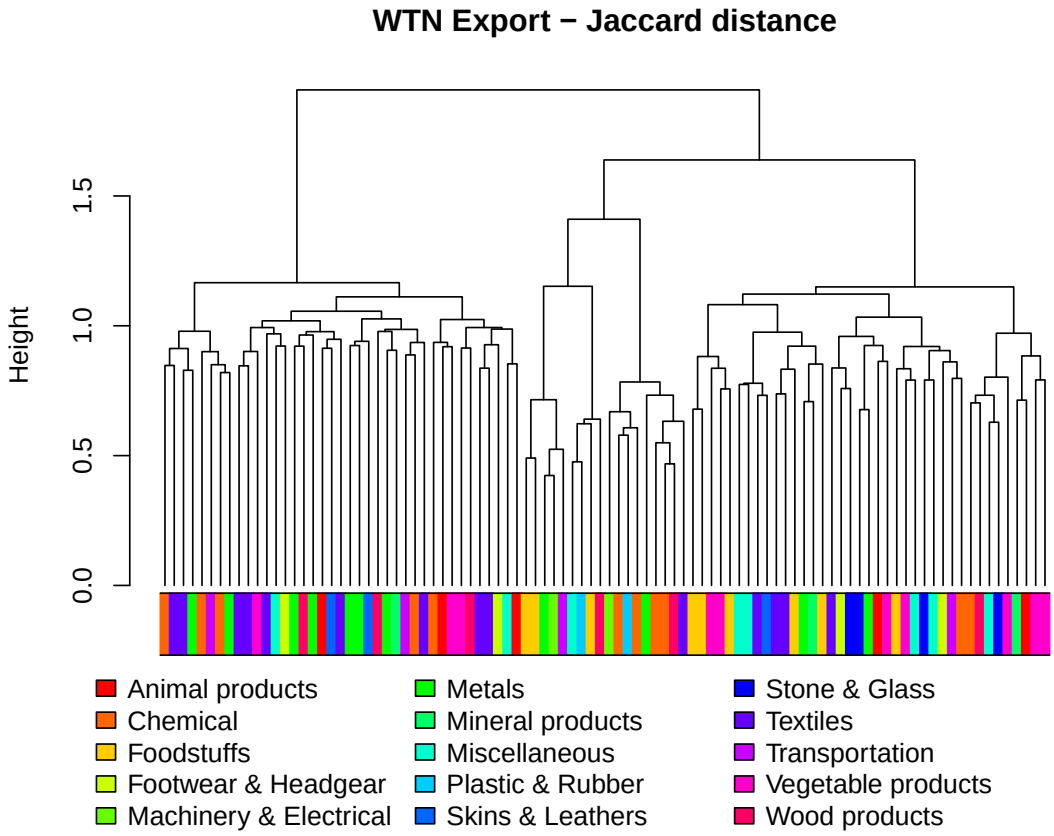
Looking at the coloured bars under the dendrograms (Figures 3.13 and 3.15) it can be noticed that none of the distances, neither in the Export nor in the Import case, is able to recover the original subdivision into 15 categories. There are some contiguous blocks of *Textiles*, *Foodstuffs*, *Vegetable products*, *Chemical* and *Metals* layers, showing similarity between some products in these category, but mainly layers of the same section are spread all over the dendrograms, showing a great variety in the trade patterns inside each section. The distances which require node correspondence show larger contiguous blocks of the *Textiles* sections in the Export dataset. This is not surprising, since many textiles products are common and simple to produce, so that almost any country produces and trades them and the corresponding layers have high node similarities.

Table 3.8: Cophenetic Correlation Coefficients of the dendrograms obtained from the Export and Import WTN datasets

Distance	Export	Import
Euclidean	0.6391596	0.6521786
Jaccard	0.3203508	0.1678422
DeltaCon	0.6406740	0.6620450
Portrait Divergence	0.6143004	0.6904337
DGCD-129	0.5727253	0.5078582

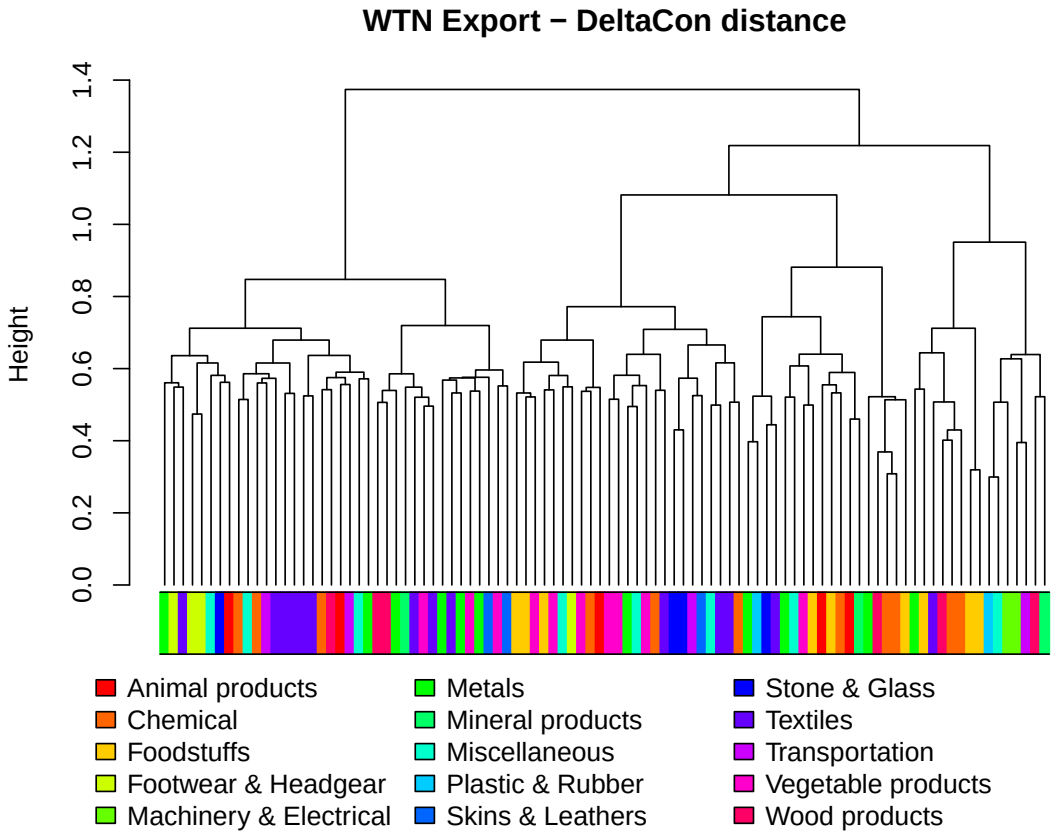


(a)

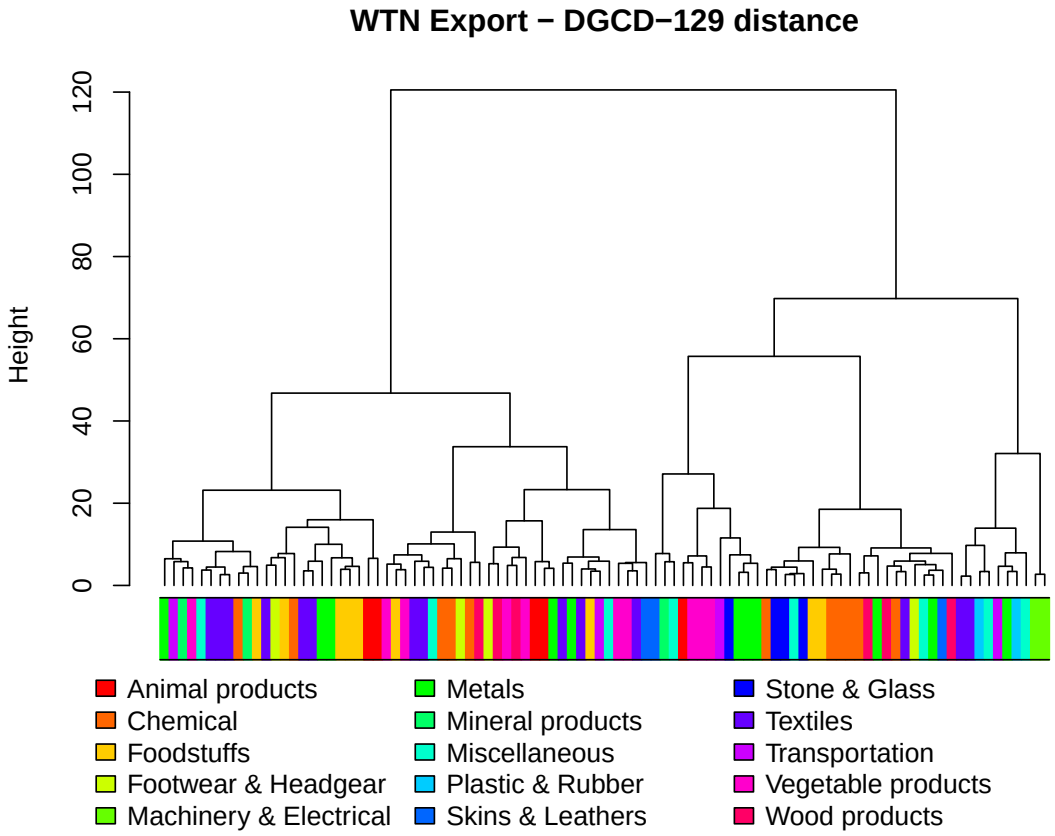


(b)

Figure 3.13: WTN Export dendrograms.

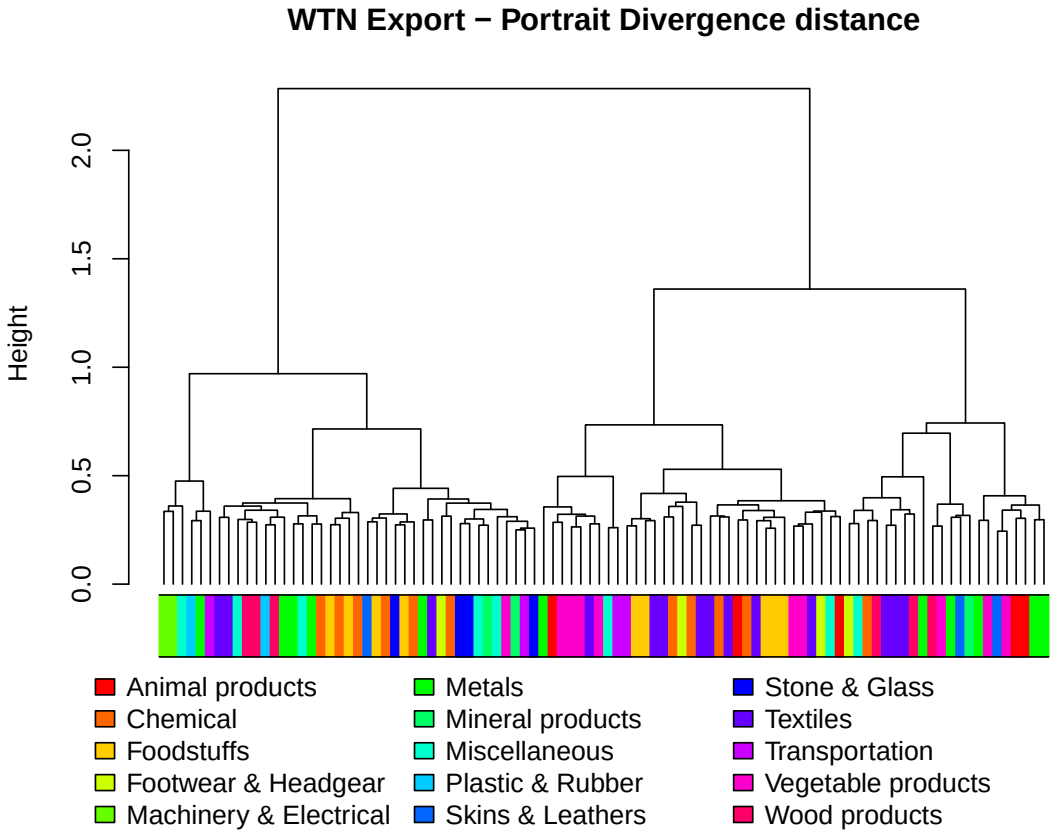


(c)



(d)

Figure 3.13 (cont.): WTN Export dendrograms.



(e)

Figure 3.13 (cont.): WTN Export dendrograms.

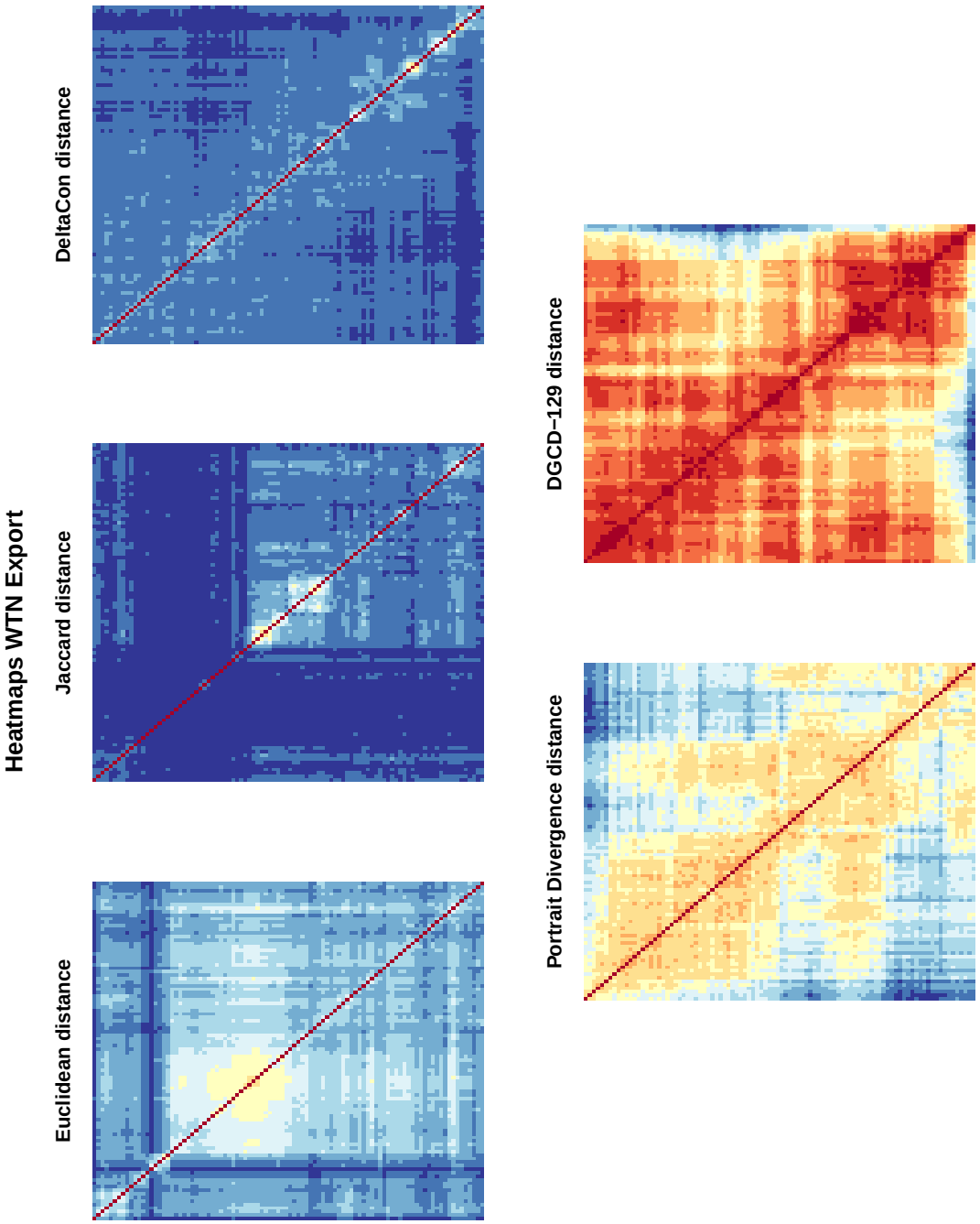
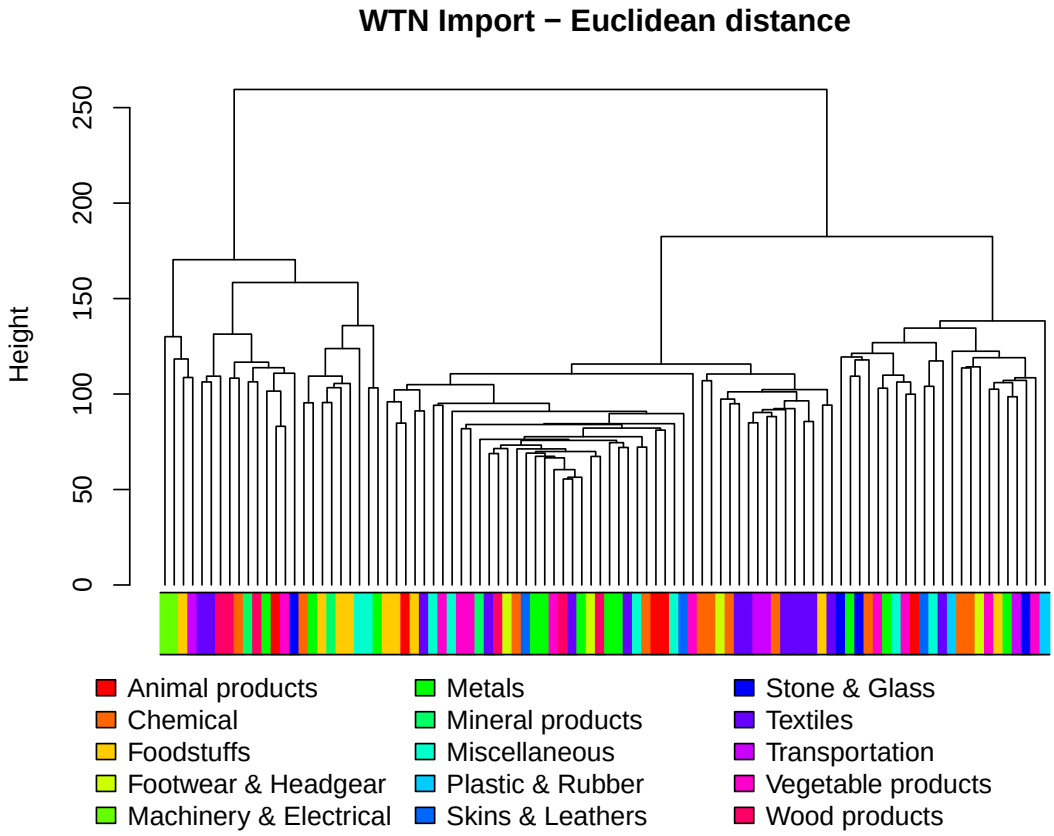
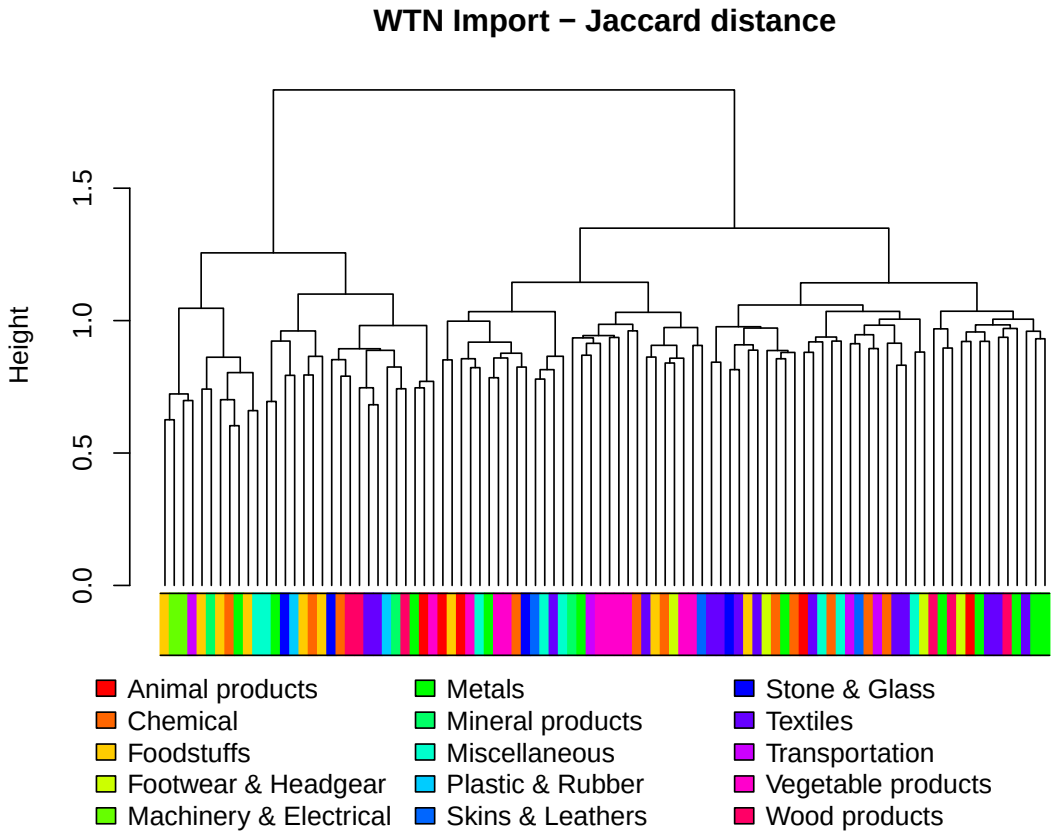


Figure 3.14: WTN Export heatmaps.

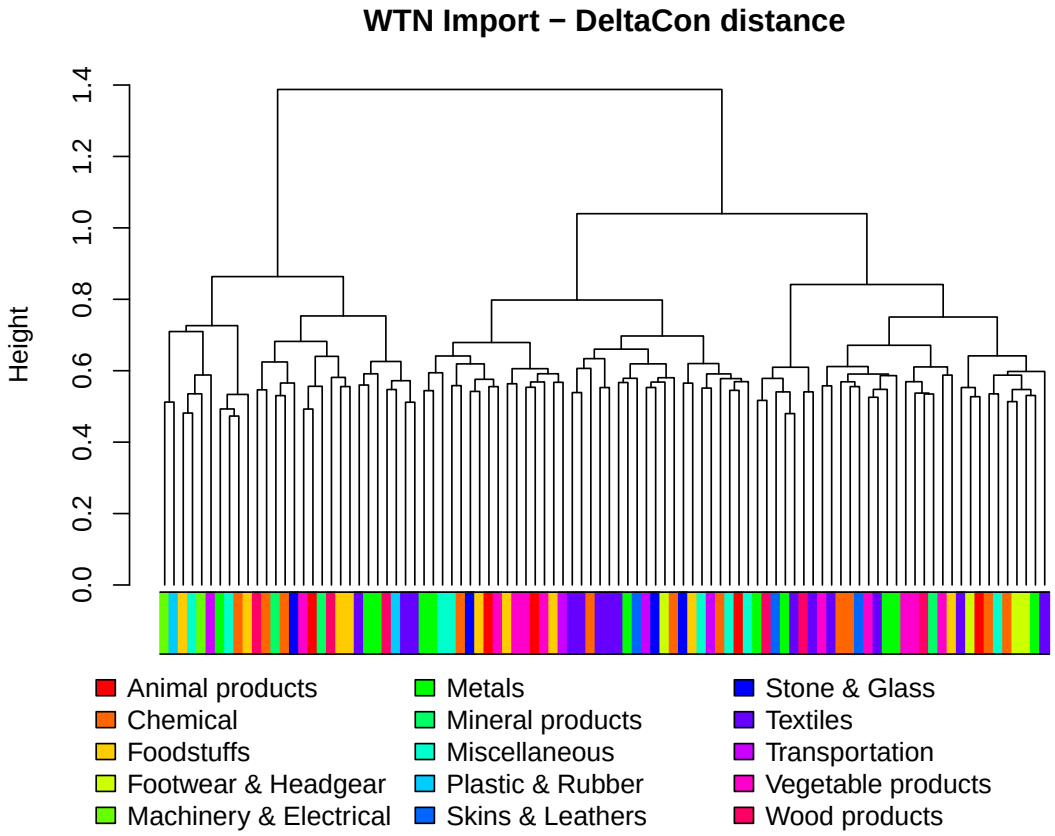


(a)

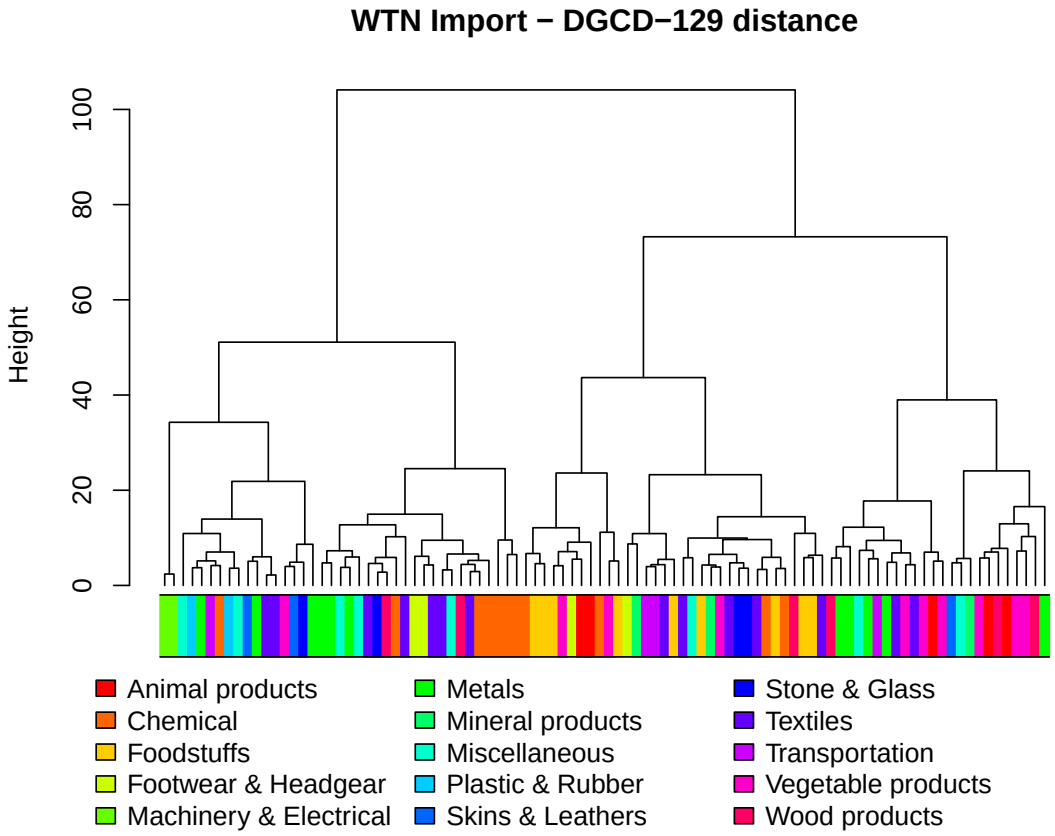


(b)

Figure 3.15: WTN Import dendrograms.

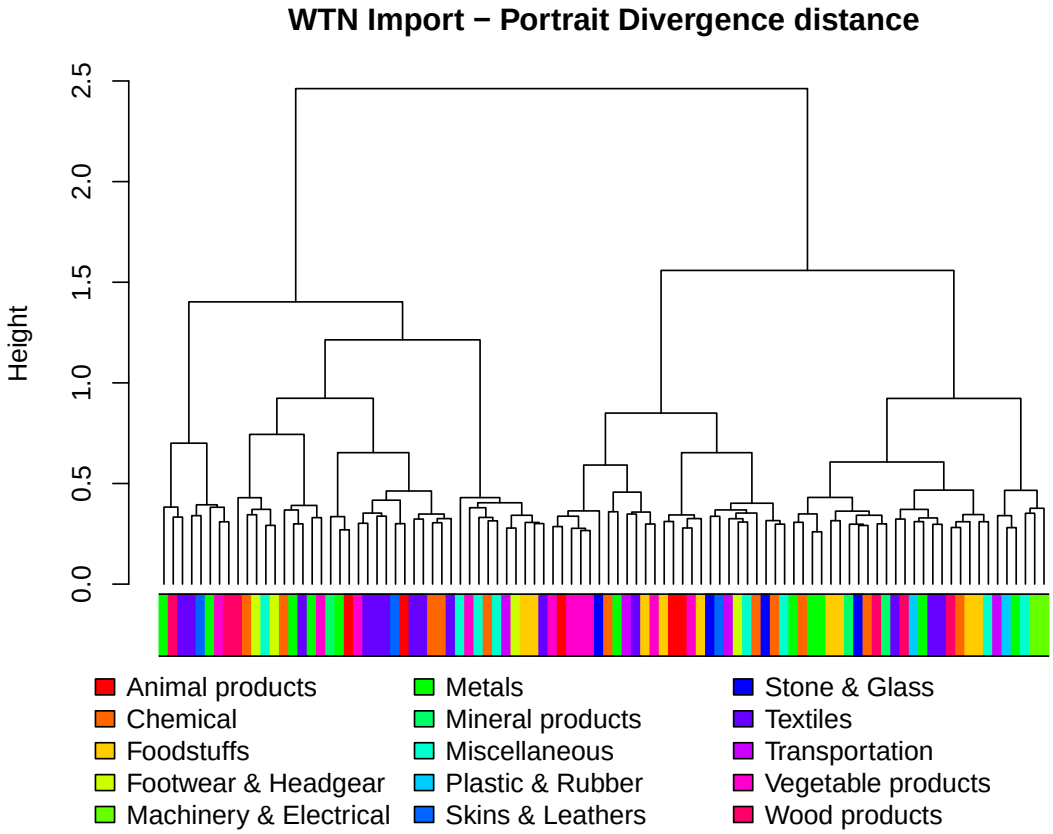


(c)



(d)

Figure 3.15 (cont.): WTN Import dendrograms.



(e)

Figure 3.15 (cont.): WTN Import dendrograms.

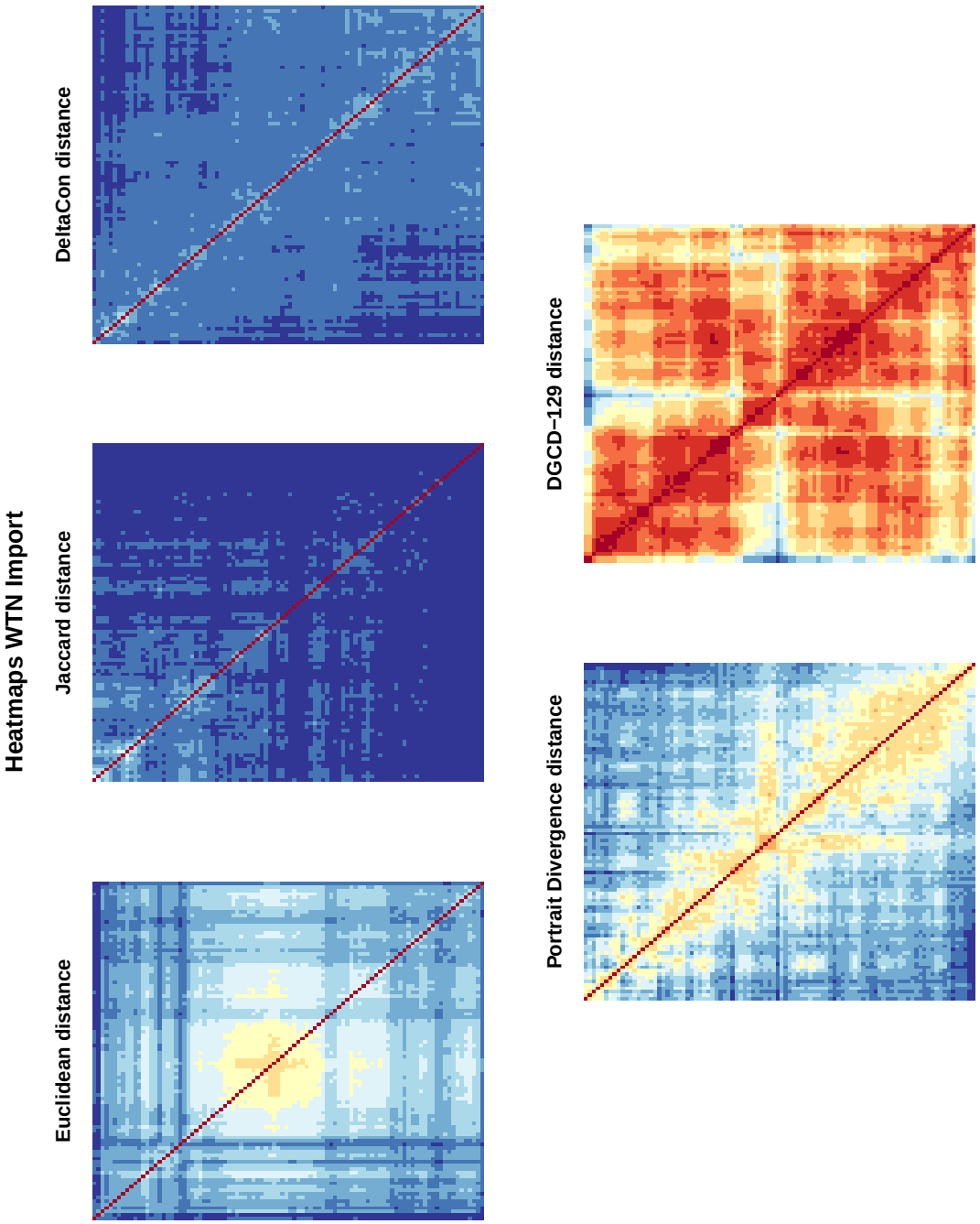


Figure 3.16: WTN Import heatmaps.

Presence of supply chains

Unlike in the FAO case study, we found a few examples of supply chain. We first considered the distances independent on node correspondence, since we expect that products belonging to a supply chain have a similar trade pattern.

In the Export case, the DGCD-129 (Figure 3.13d) distance identifies two layers as the most separated from all the others, namely *Nuclear reactors, boilers, machinery and mechanical appliances; parts thereof* and *Electrical machinery and equipment and parts thereof; sound recorders and reproducers; television image and sound recorders and reproducers, parts and accessories of such articles*; note that they both belong to the section *Machinery and Electrical*. Then, the leaves adjacent to these layers are: *Vehicles; other than railway or tramway rolling stock, and parts and accessories thereof*; *Iron or steel articles*; *Plastics and articles thereof* and *Optical, photographic, cinematographic, measuring, checking, medical or surgical instruments and apparatus; parts and accessories*. This is clearly a first example of supply chain, since we find raw materials, instruments and parts to assemble the final products, and the final products themselves. Moreover, by looking at the heatmaps of the DGCD-129 distance, it can be noticed that this group is separated from all the other layers. This grouping is a consequence of the economical importance of the considered industries (think for instance to the automotive industry) and of the fact that these final goods requires an high level of technology and technological skills to be produced. We find exactly the same group at the leftmost part of the Portrait Divergence dendrogram (Figure 3.13e). Another supply chain is constituted by *Cereals* and *Products of the milling industry; malt, starches, inulin, wheat gluten*, which are close in both the dendrograms. In the Import case, we find the supply chain related to vehicles and highly-technological instruments both in DGCD-129 and in Portrait Divergence (Figures 3.15d and 3.15e), even if the separation of this group from all the other layers in not pronounced. Again, DGCD-129 distance reveals that the two layers belonging to the *Machinery and Electrical* section are the most separated from all the others. The other layers are grouped in such a way that it is not possible to identify any other meaningful supply chain.

It is interesting to notice that in the Export case we again find close each other the layers related to the vehicles and highly-technological instruments supply chain that we found with the distances independent on node correspondence. In particular, both distances put adjacent five out of the six layers of the supply chain; the Euclidean distance excludes *Electrical machinery and equipment and parts thereof; sound recorders and reproducers; television image and sound recorders and reproducers, parts and accessories of such articles*, while the DeltaCon distance excludes *Iron or steel articles*. This fact is relevant, because it shows that there are high similarities between the countries with respect to the products belonging to this supply chain. In the Import case, instead, only the DeltaCon distance recover the supply chain at the leftmost part of the dendrogram, with all the six layers, while the Euclidean distance only gathers three layers: *Nuclear reactors, boilers, machinery and mechanical appliances; parts thereof*; *Electrical machinery and equipment and parts thereof; sound recorders and reproducers; television image and sound recorders and reproducers, parts and accessories of such articles* and *Vehicles; other than railway or tramway rolling stock, and parts and accessories*

thereof; the other layers are spread in the dendrogram.

We also looked for other possible supply chains that arose from the distances which require node correspondence. The Euclidean distance in the Export case puts adjacent *Articles of leather; saddlery and harness; travel goods, handbags and similar containers; articles of animal gut (other than silk-worm gut)* and *Wadding, felt and nonwovens, special yarns; twine, cordage, ropes and cables and articles thereof*. The DeltaCon distance in the Export case instead puts adjacent three out of four layers of the *Footwear and Headgear* section, namely *Headgear and parts thereof*; *Umbrellas, sun umbrellas, walking-sticks, seat sticks, whips, riding crops; and parts thereof* and *Feathers and down, prepared; and articles made of feather or of down; artificial flowers; articles of human hair* along with *Man-made filaments*. In the Import case, instead, the DeltaCon distance pairs together again *Headgear and parts thereof*, but this time with *Man-made staple fibres*. Another example of supply chain that DeltaCon identifies in the Import case is composed of *Vegetable plaiting materials; vegetable products not elsewhere specified or included* and *Manufactures of straw, esparto or other plaiting materials; basketware and wickerwork*.

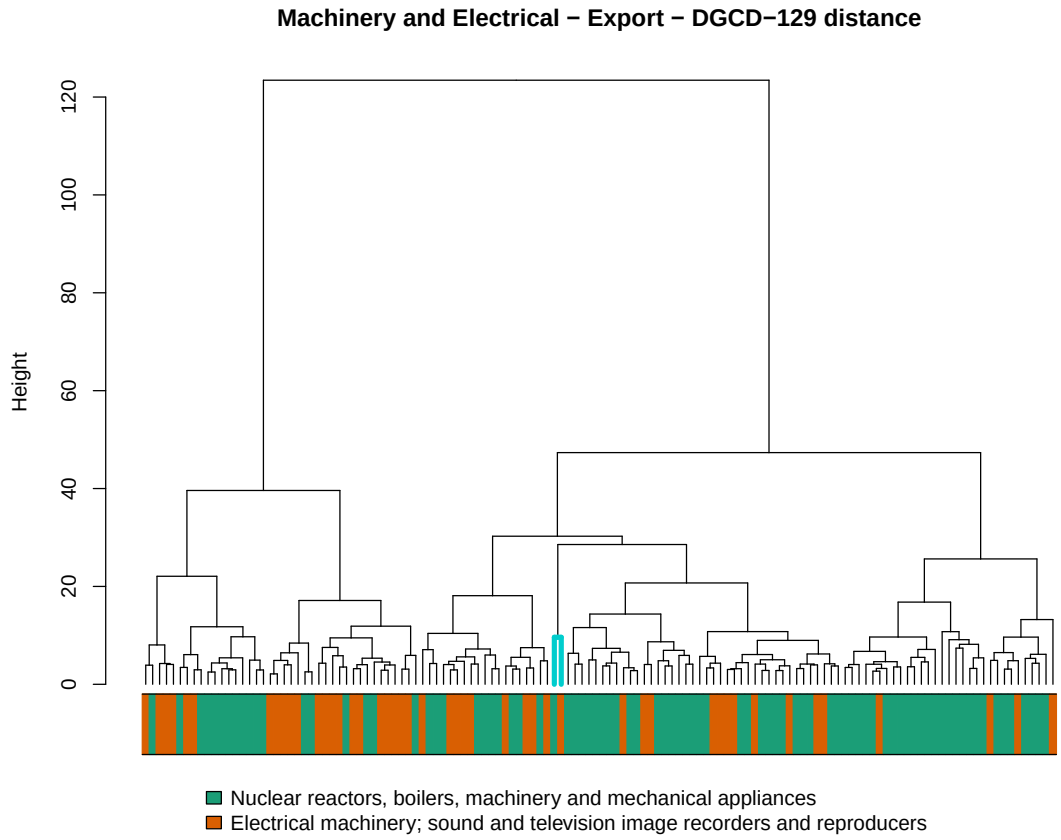
The last result we state is not related to the supply chains, but with the production of specific products. The Euclidean distance in the Export case pairs together *Coffee, tea, mate and spices* and *Cocoa and cocoa preparations*, which are both produced in countries with similar and peculiar climate. This result agree with what we found in the FAO case study.

Machinery and Electrical section

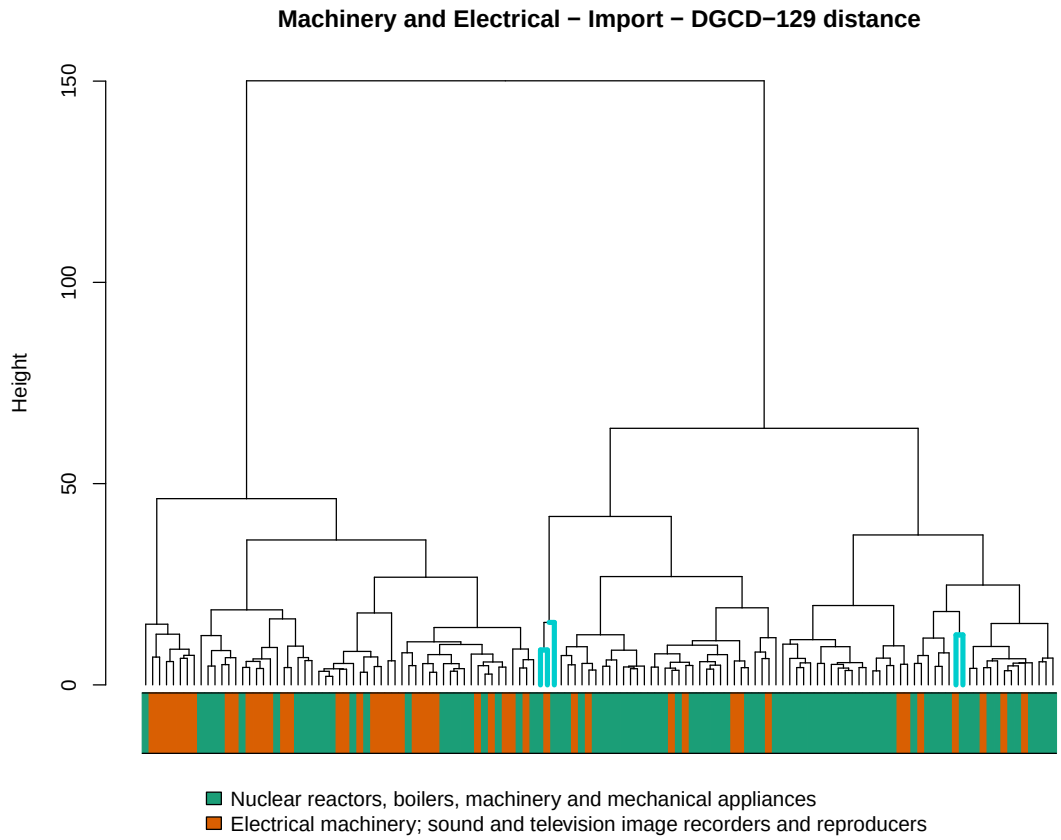
We wanted to further investigate the differentiation of the trade patterns at a more disaggregated level, i.e., by considering the HS 4-digits classification of products of some specific HS sections. In this part of the analysis we only consider the results provided by the DGCD-129 distance, since we are interested in investigating the structure of the considered networks.

The first HS section that we consider is *Machinery and Electrical*. This section contains two layers with 2-digit classification, namely *Nuclear reactors, boilers, machinery and mechanical appliances; parts thereof* and *Electrical machinery and equipment and parts thereof; sound recorders and reproducers; television image and sound recorders and reproducers, parts and accessories of such articles*, which contain a total of 132 products with 4-digit classification. Since all these products are complex, in the sense that they requires many technological knowledge and skills to be produced, we expect to find that they all have a similar structure, with a very dense core. For the same reason, we do not expect to find great dissimilarities between the products belonging to the two different 2-digit layers.

We performed a clustering analysis and we obtained the dendrograms shown in Figure 3.17 and the heatmaps shown in Figure 3.18; the Cophenetic Coefficient of the dendrograms is 0.695 in the Export case and 0.629 in the Import case. In both dendrograms a clear division in two clusters is displayed; nonetheless, we can not give a clear interpretation of these partitions, since products in each cluster are very heterogeneous and used for many different tasks. Moreover, we see from the



(a)



(b)

Figure 3.17: Dendrograms of the products in the *Machinery and Electrical* HS section. The leaves (products) highlighted in blue are discussed in the text.

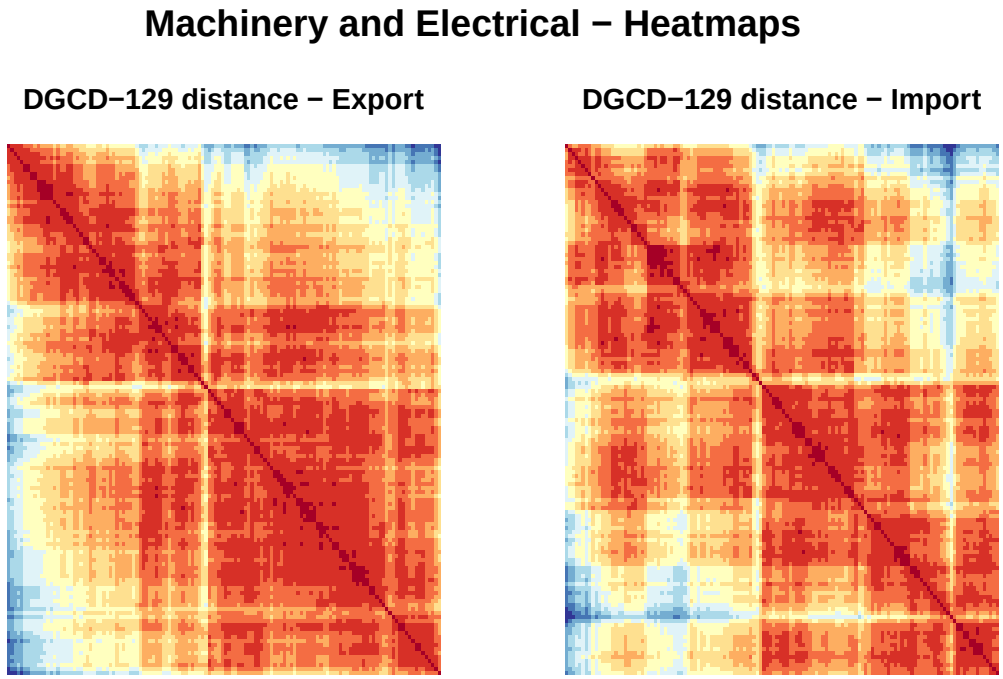
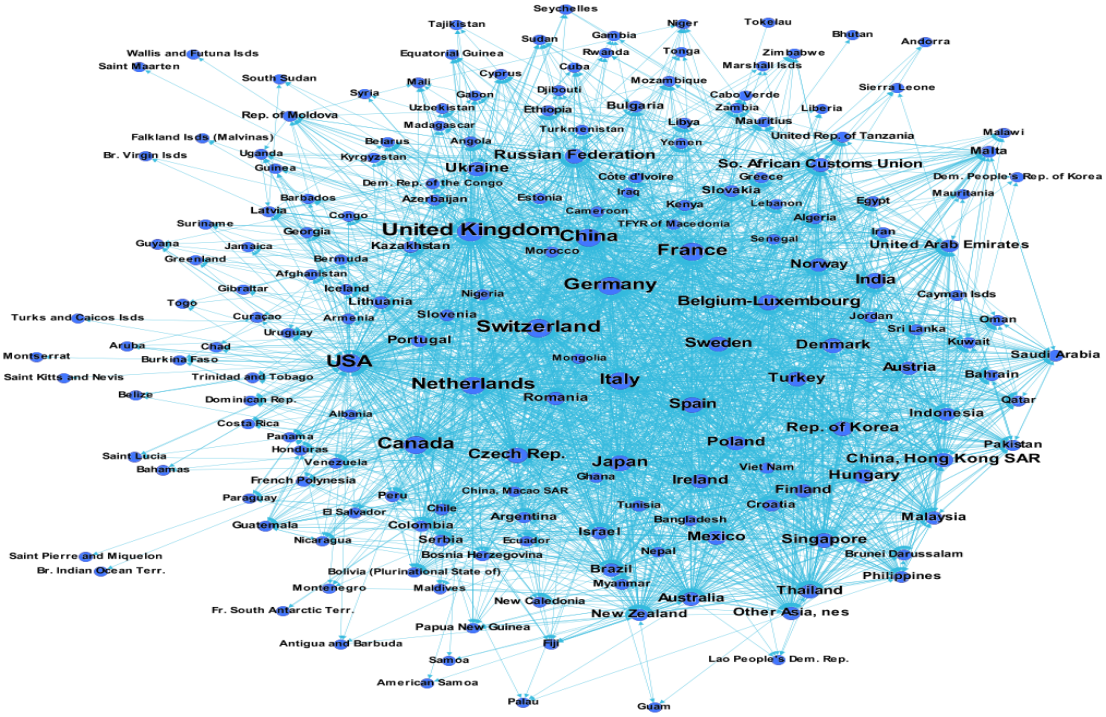


Figure 3.18: Heatmaps of the products in the *Machinery and Electrical* HS section.

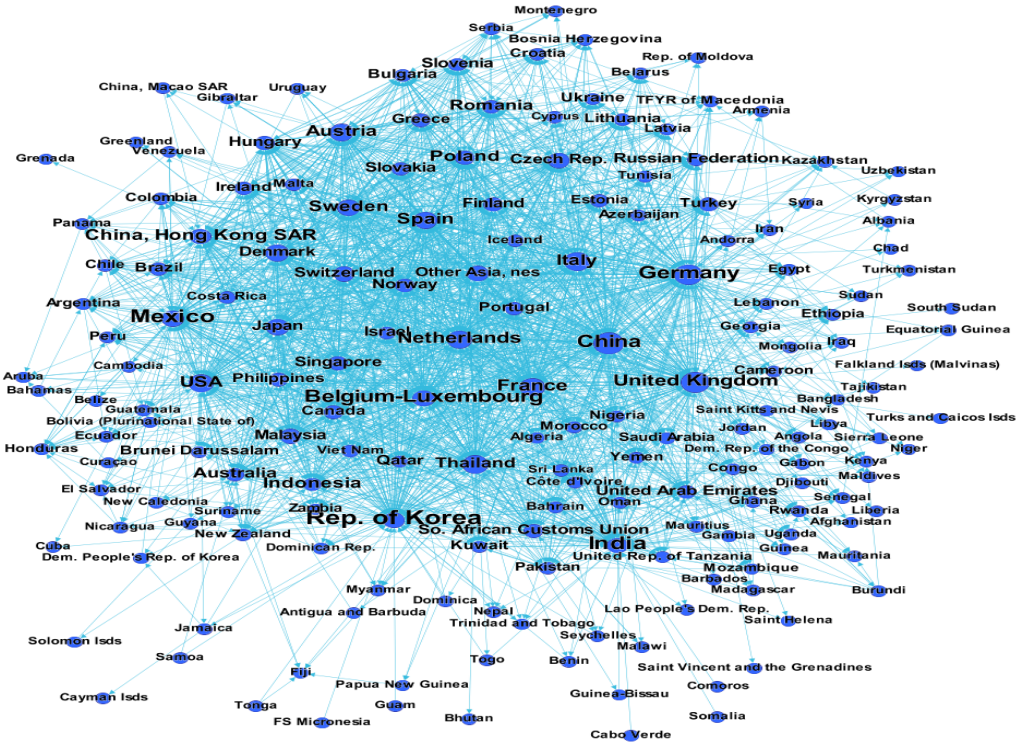
coloured bars under the dendrograms that the products belonging to the two 2-digit layers are spread all over the dendrograms and do not form large contiguous blocks. We can also see from the heatmaps that more or less all the layers are at small distance to each other, especially in the Export case. These results clearly show that, even if they are very heterogeneous, these kind of products share a common trade structure, which is mainly due to their intrinsic complexity.

We note that there are two products in the Export case, and five in the Import case, which seem to be more distant from all the others. These products are, for the Export case, *Turbo-jets, turbo-propellers and other gas turbines* (see Figure 3.19a) and *Waste and scrap of primary cells, primary batteries and electric accumulators* (see Figure 3.19b), which we highlighted in Figure 3.17a. In the Import case, these products are *Nuclear reactors; fuel elements (cartridges), non-irradiated, for nuclear reactors, machinery and apparatus for isotopic separation* (see Figure 3.19c); *Machinery for preparing, tanning or working hides, skins or leather* (see Figure 3.19d) and *Magnetic tape recorders and other sound recording apparatus*, other than the two products identified in the Export case. These are partly the most complex and high-technological products present in this section (such as turbo-jets and nuclear reactors) and partly worldwide used products (such as batteries and machinery for working skins), so that we can expect they are identified to have the most dissimilar structure among the products of this section.

At last, we want to stress that this section contains HS 4-digit layers with very high density with respect to other sections. In particular, the density in the Export case ranges from 0.040 to 0.227, with 34 products out of 132 (around the 26%) having a density smaller or equal than 0.1. In the Import case, instead, we observe

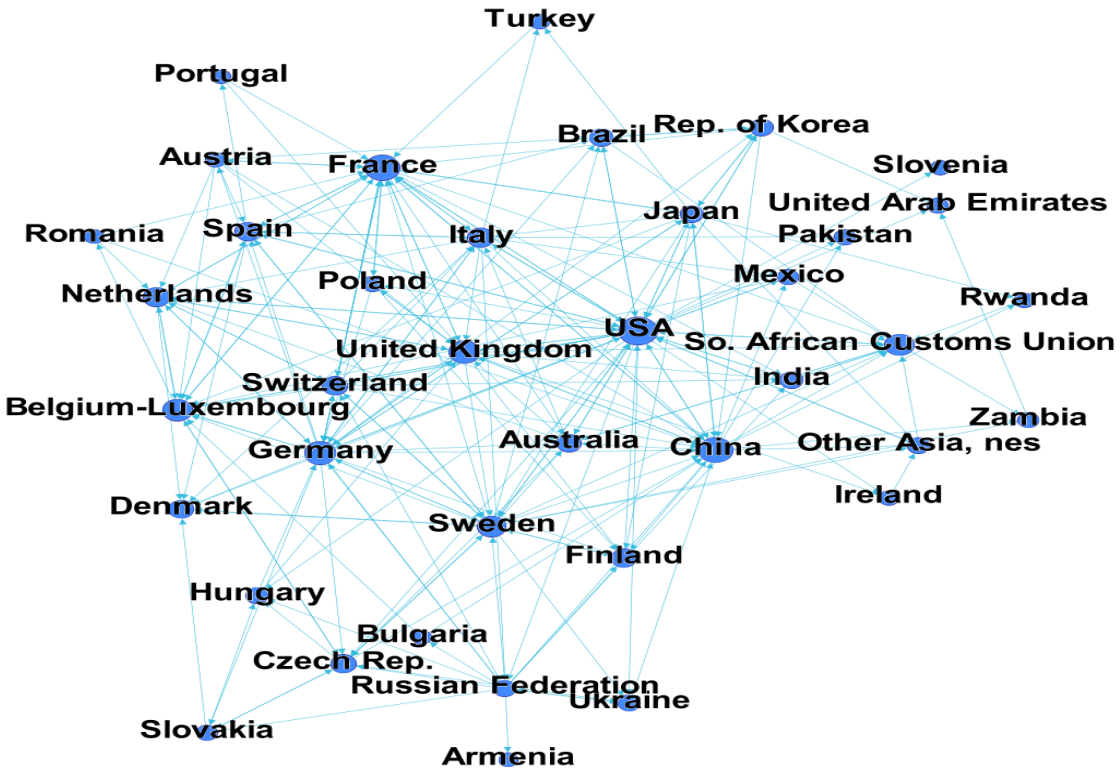


(a) Turbo-jets, turbo-propellers and other gas turbines (Export)

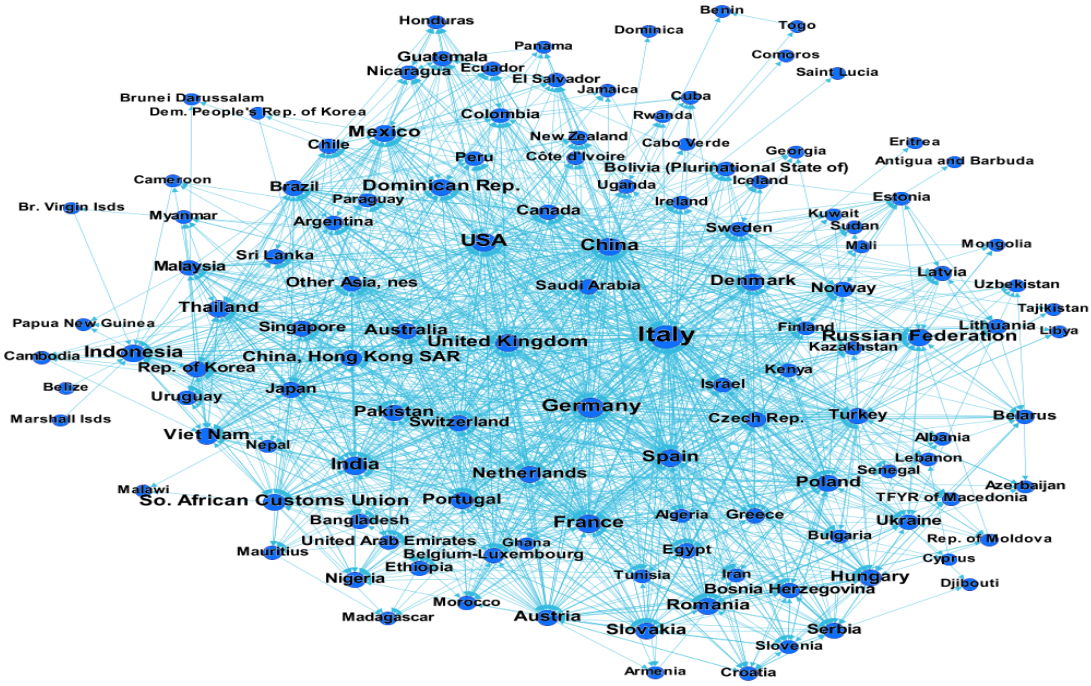


(b) Waste and scrap of primary cells, primary batteries and electric accumulators (Export)

Figure 3.19: Some of the most distant products from all the others in the *Machinery and Electrical* section.



(c) Nuclear reactors; fuel elements (cartridges), non-irradiated, for nuclear reactors, machinery and apparatus for isotopic separation (Import)



(d) Machinery for preparing, tanning or working hides, skins or leather (Import)

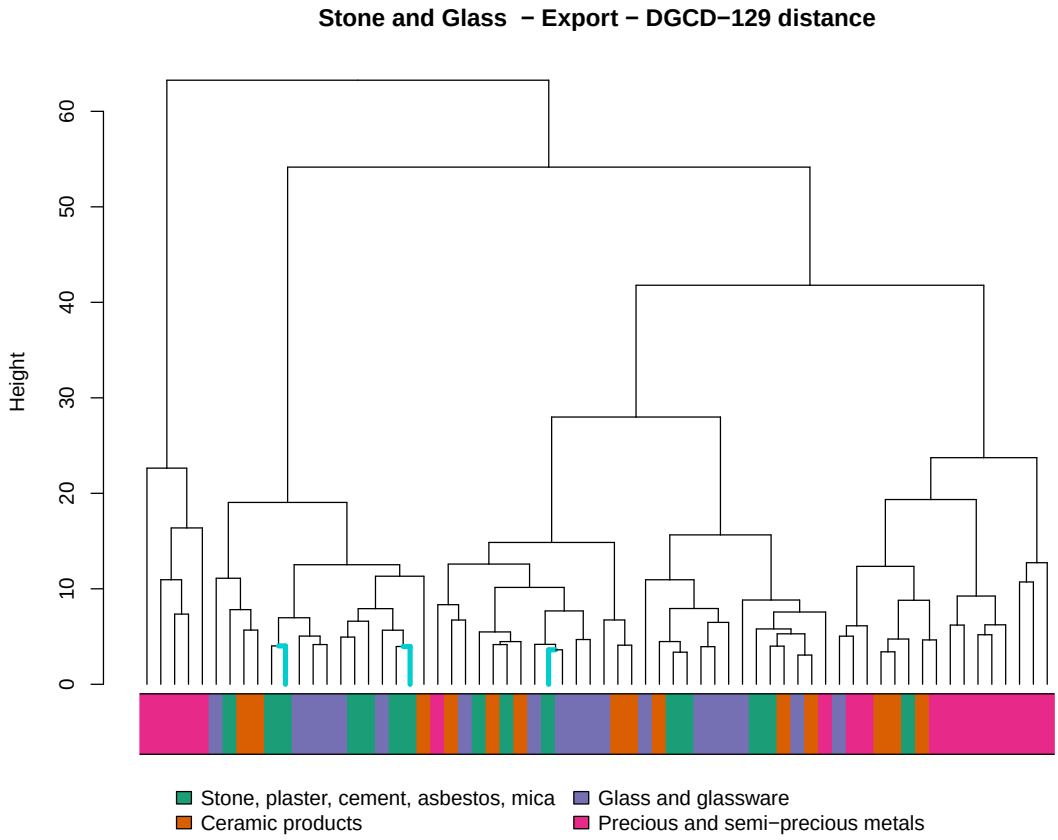
Figure 3.19 (cont.): Some of the most distant products from all the others in the *Machinery and Electrical* section.

a much larger density, ranging from 0.063 to 0.381, with now only 4 products having a density smaller or equal than 0.1. The different values of the density between Export and Import case are due to the pruning of the layers obtained using the RCA index. There could be various explanation for this difference in the density. For instance, as suggested in [23], usually countries tend to highly diversify import sources for competition reasons while export partners are generally more limited in number due to costs in penetrating new markets, so that the Import case show higher densities. Nonetheless, this difference may also be due to the availability and the precision of data, since countries often record import data better than export data.

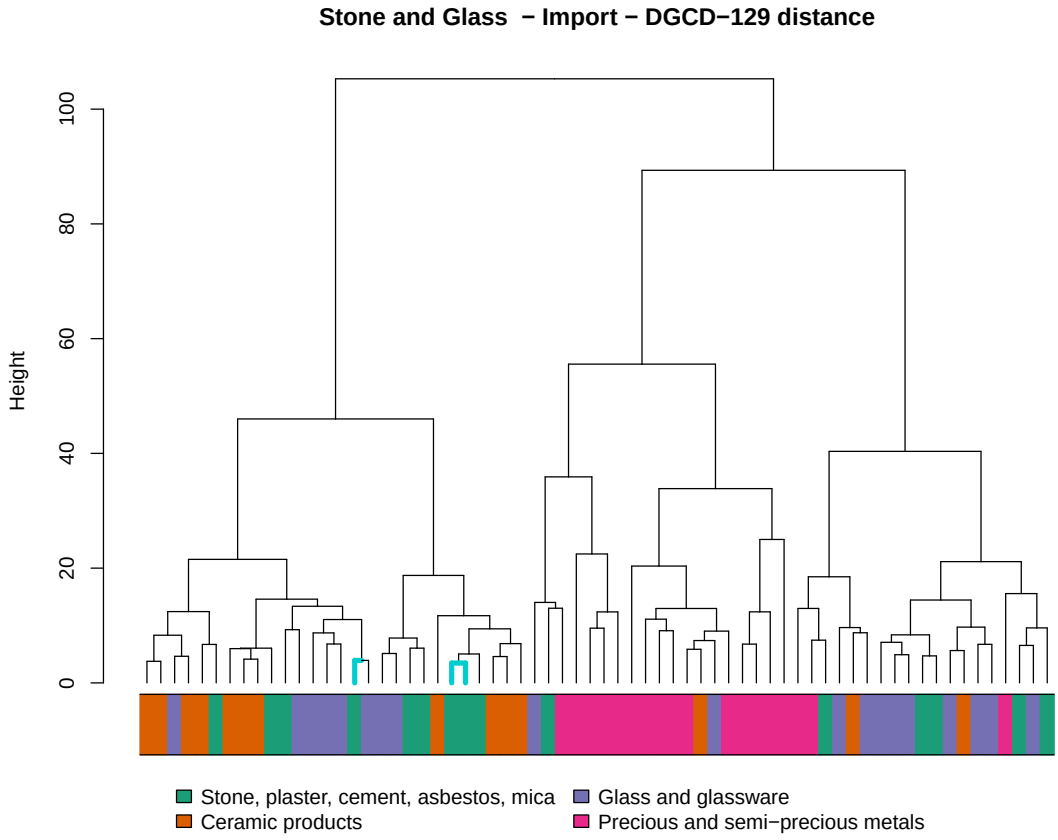
Stone and Glass section

The second HS section that we consider for a deeper analysis is *Stone and Glass*. This sections contains four HS 2-digit layers: *Stone, plaster, cement, asbestos, mica or similar materials; articles thereof*; *Ceramic products*; *Glass and glassware* and *Natural, cultured pearls; precious, semi-precious stones; precious metals, metals clad with precious metal, and articles thereof; imitation jewellery; coin*. The HS 4-digit classification of this section comprises 66 products. We are interested in highlighting differences in their trade patterns, especially in those concerning stones and cement. In fact, they are simple but heavy materials with a high transportation cost. We expect that each country trades this kind of products only with neighbouring countries, so that their trade patterns should display some regional structure.

The dendrograms we obtained from the clustering analysis are shown in Figure 3.20, while the heatmpas in Figure 3.21. The Cophenetic Coefficients of the dendrograms are 0.648 for the Export case and 0.580 for the Import case, which are sufficiently large. In the Export case our expectation are not met. We see from the heatmap that almost all products in this section are close to each other. We can see from the coloured bar under the dendrogram that the products of the first three 2-digit layers are mixed up together. We also highlighted the leaves corresponding to *Articles of plaster or of compositions based on plaster*; *Articles of cement, of concrete or of artificial stone* and *Articles of asbestos-cement, of cellulose fibre-cement or the like*, which should be the materials with the highest transportation costs; we can see that they are not even adjacent. Looking at the *Articles of cement, of concrete or of artificial stone* layer, we find for instance that the USA trades with almost all countries in the world, even with the ones which are difficult or costly to reach. There is an exception to this behaviour: the products belonging to the precious metals layer form two large contiguous blocks. Few of them form a clearly separated cluster, which then have a different trade structure from all the other products of the section; these products are *Waste and scrap of precious metal or of metal clad with precious metal*; *Gold (including gold plated with platinum)*; *Diamonds, whether or not worked, but not mounted or set.*; *Precious stones (other than diamonds) and semi-precious stones* and *Coin*. Another group of precious metals products, instead, is gathered at the rightmost part of the dendrogram; such articles are: *Articles of natural or cultured pearls, precious or semi-precious stones*; *Platinum, unwrought or in semi-manufactured forms, or in powder form*; *Silver (including silver plated with gold or platinum)*; *Articles of*



(a)



(b)

Figure 3.20: Dendrograms of the products in the *Stone and Glass* HS section.

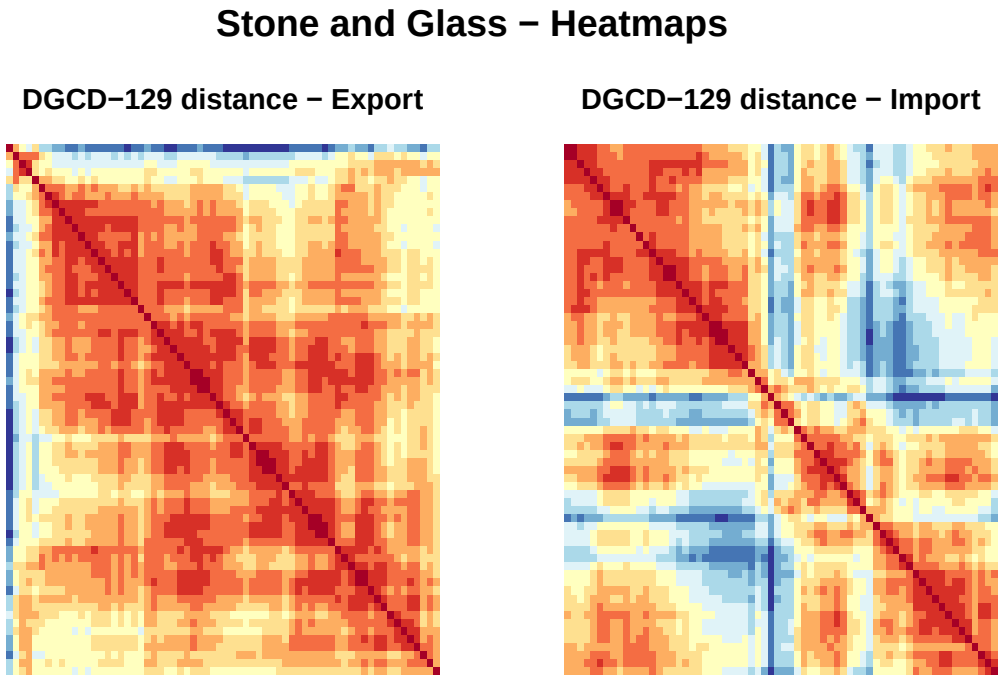


Figure 3.21: Heatmaps of the products in the *Stone and Glass* HS section.

jewellery and parts thereof, of precious metal; *Imitation jewellery*; *Synthetic or reconstructed precious or semi-precious stones*; *Pearls, natural or cultured*; *Base metals or silver, clad with gold, not further worked* and *Base metals, silver or gold, clad with platinum*. From the heatmap, we see that this second group has similar trade structures to all the other products, so that we found a double behaviour in the trade structure of the precious metals. The fact that all products, except some of the precious metals, display similar trade patterns may be due to different but equivalent issues in the transportation of such goods. In fact, cement and similar building materials have high transportation costs due to their weight, while glass and ceramics also have high transportation costs but due to their fragility.

In the Import case we found a quite different behaviour. The dendrogram and the heatmap show a division in three clusters. One of them contains almost all the precious metals products. The cluster at the left part of the dendrogram gathers together most of the products related to the building industries, such as *Unglazed ceramic flags and paving, hearth or wall tiles*; *Glazed ceramic flags and paving, hearth or wall tiles*; *Float glass and surface ground or polished glass, in sheets*; *Refractory bricks, blocks, tiles and similar refractory ceramic constructional goods*; *Worked monumental or building stone (except slate) and articles thereof*; *Worked slate and articles of slate or of agglomerated slate*; *Multiple-walled insulating units of glass*; *Articles of cement, of concrete or of artificial stone*; *Paving blocks, slabs, bricks, squares, tiles*; *Fabricated asbestos fibres; mixtures with a basis of asbestos*; *Slag wool, rock wool and similar mineral wools*; *Roofing tiles, chimney-pots, cowls, chimney liners, architectural ornaments*; *Articles of plaster or of compositions based on plaster*; *Articles of asbestos-cement, of cellulose fibre-cement or the like*; *Panels,*

boards, tiles, blocks and similar articles of vegetable fibre and *Ceramic building bricks, flooring blocks, support or filler tiles*. Instead, the third cluster at the right part of the dendrogram gathers more heterogeneous products, but we can find a prevalence of glass products and some high-technological or precision products, such as *Articles of jewellery and parts thereof, of precious metal*; *Signalling glassware and optical elements of glass*; *Ceramic wares for laboratory, chemical or other technical uses*; *Safety glass, consisting of toughened (tempered) or laminated glass* and *Laboratory, hygienic or pharmaceutical glassware*. Therefore, in the Import case our initial expectations are met, since many of the products used in the building industry share a similar structure; this may also denote the fact that importing countries tend to buy all the building materials they need from the same suppliers, to lower transportation costs. Moreover, we find that almost all precious metals products show a different structure with respect to all other products in the *Stone and Glass* section.

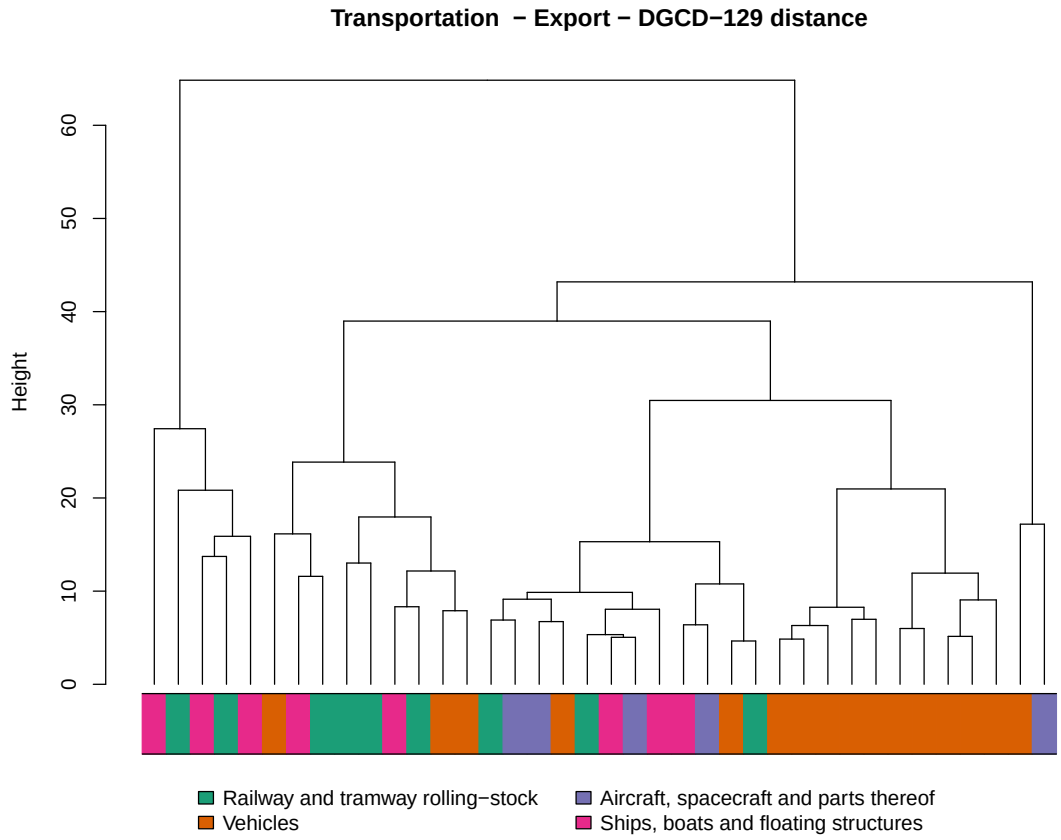
As a last remark, we observe again the increase in the density when switching from the Export to the Import case. In the Export case, density ranges from 0.042 to 0.146, while in the Import case from 0.048 to 0.311.

Transportation section

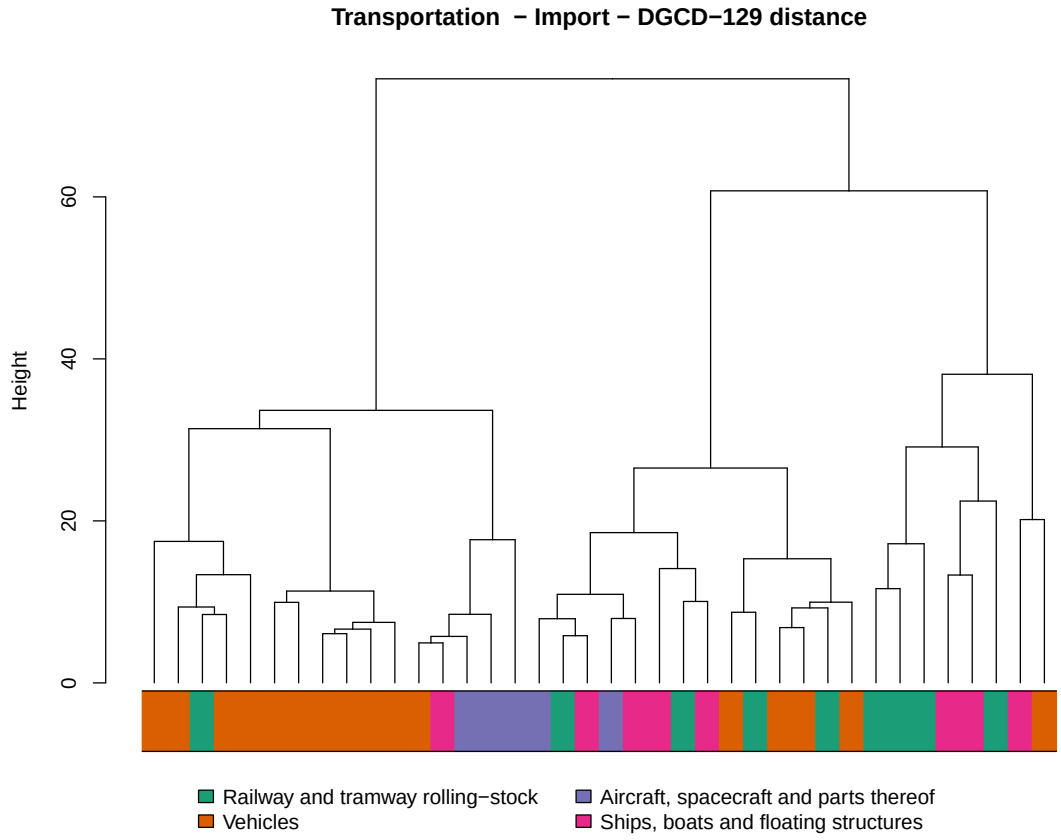
Another HS section that we consider is *Transportation*. It contains four 2-digit layers, namely *Railway, tramway locomotives, rolling-stock and parts thereof; railway or tramway track fixtures and fittings and parts thereof; mechanical (including electro-mechanical) traffic signalling equipment of all kinds*; *Vehicles; other than railway or tramway rolling stock, and parts and accessories thereof*; *Aircraft, spacecraft and parts thereof* and *Ships, boats and floating structures*. These four layers contain in total 38 4-digit products. In this case, our expectation is to find some different grouping between products that are intended as consumer goods, such as most of the vehicles products, and products that are instead intended for more specific companies, such as the products in the railway and tramway layer and those in the aircraft and spacecraft layer.

We show the dendrograms we obtained from the clustering analysis in Figure 3.22 and the corresponding heatmaps in Figure 3.23. The Cophenetic Coefficient of such dendrograms is 0.788 in the Export case, and 0.549 in the Import case, which is still acceptable.

We observe a similar behaviour as in the *Stone and Glass* section. In fact, in the Export case, the heatmap shows a large number of products close to each others, with only two groups that are more separated. The first group contains five products and it forms a single cluster; these products are *Vessels and other floating structures for breaking up*; *Rail locomotives powered from an external source of electricity*; *Tugs and pusher craft*; *Other rail locomotives; locomotive tenders* and *Fishing vessels, factory ships and other vessels; for processing or preserving fishery products*. The other group also form a small separated cluster which contains only two products, the first referring to parts and accessories of motor vehicles and the second referring to parts and accessories of aircraft and spacecraft. This suggests that in general the market concerning parts and accessories for vehicles has a different structure than the market of the final goods. Another similarity to the



(a)



(b)

Figure 3.22: Dendrograms of the products in the *Transportation* HS section.

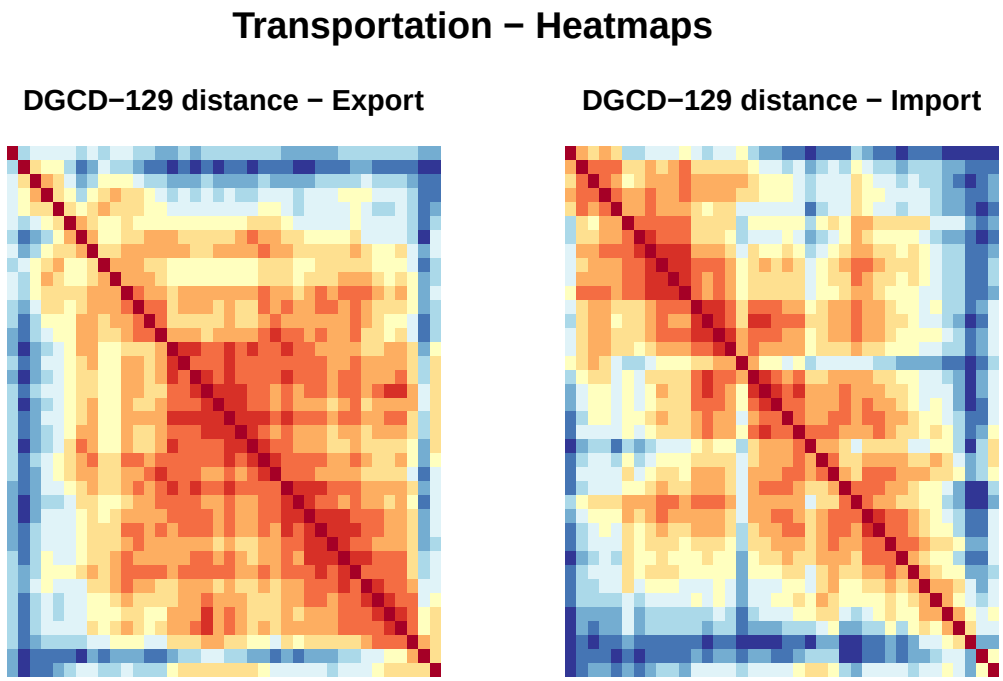


Figure 3.23: Heatmaps of the products in the *Transportation* HS section.

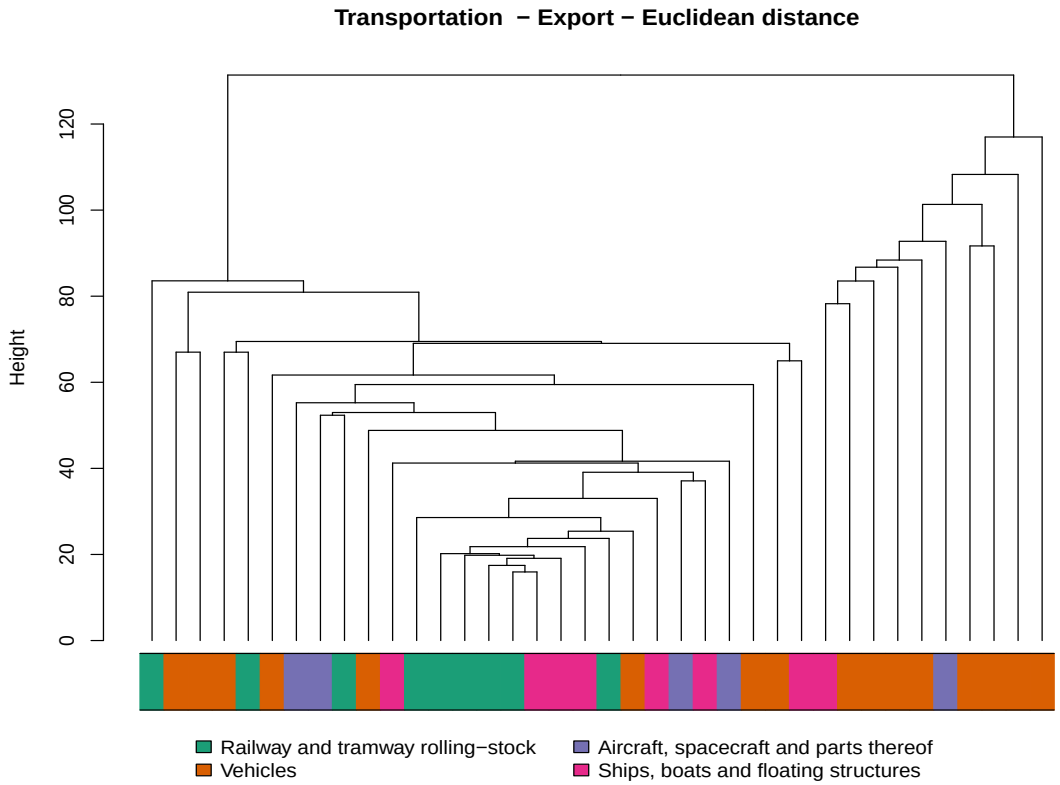


Figure 3.24: Dendrogram, obtained with the Euclidean distance, of the products in the *Transportation* HS section.

Stone and Glass case is that we find the products of one of the 2-digit layer, namely *Vehicles; other than railway or tramway rolling stock, and parts and accessories thereof*, gathered together both in the Export and in the Import case, forming a large contiguous block, while the products belonging to the other three 2-digit layers are mixed up together. Only the *Aircraft, spacecraft and parts thereof* also form a contiguous block in the Import case. This behaviour of the vehicle layer may be due to its market structure, formed by few and very large companies which, with their strength and leading position, shape the world exchanges. This is confirmed also by the dendrograms obtained from the Euclidean distance, shown in Figure 3.24. Indeed, we see that in the coloured bars we find again large contiguous blocks of the products belonging to the vehicles layer, denoting that for this products the same countries have high node similarities. We looked at the highest out-degrees in the layers representing the products *Vehicles; public transport passenger type*; *Motor cars and other motor vehicles; principally designed for the transport of persons (other than those of heading no. 8702), including station wagons and racing cars*; *Motor vehicles for the transport of goods* and *Motorcycles (including mopeds) and cycles fitted with an auxiliary motor*. We found that in all these cases the largest exporters are China, Japan, South Korea, Germany, USA, Spain, United Kingdom, Belgium-Luxembourg, Thailand, Italy and India, where the largest automotive companies are based.

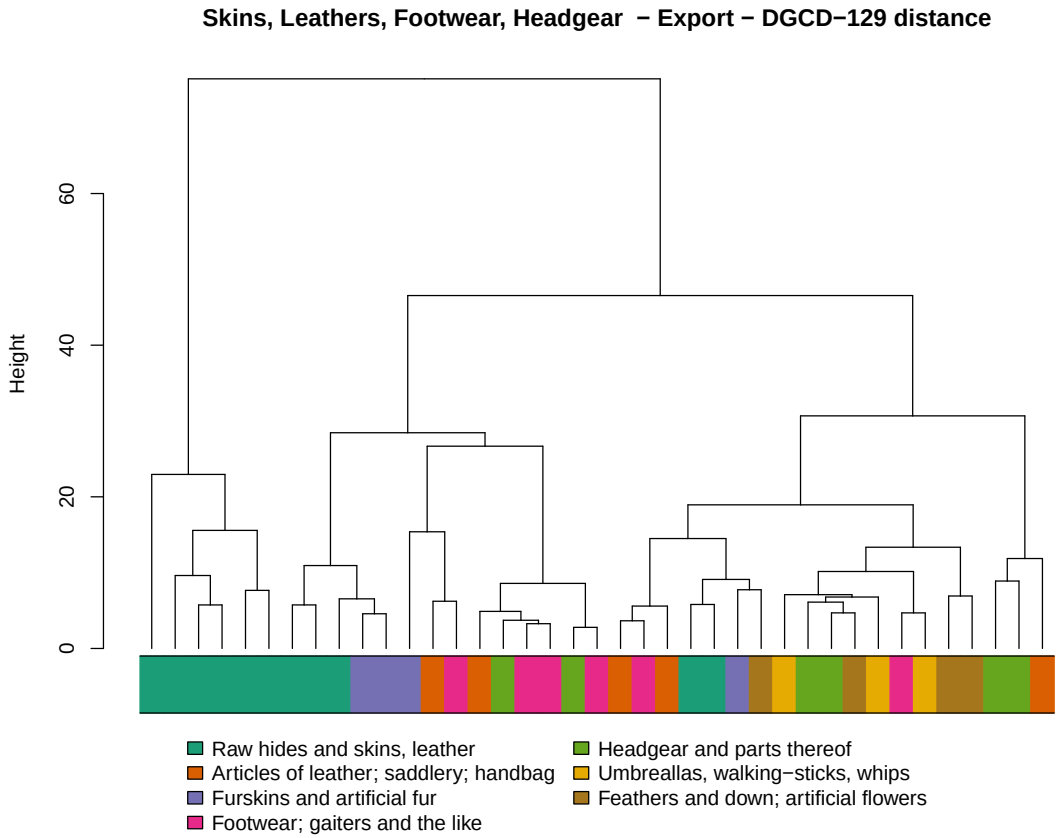
We again observe the same shifting of the density to larger values when switching from Export to Import layers, even if in this case the extent is smaller. In the Export case, the density ranges from 0.031 to 0.202, while in the Import case from 0.035 to 0.251.

Raw hides, Skins and Leathers and Footwear and Headgear sections

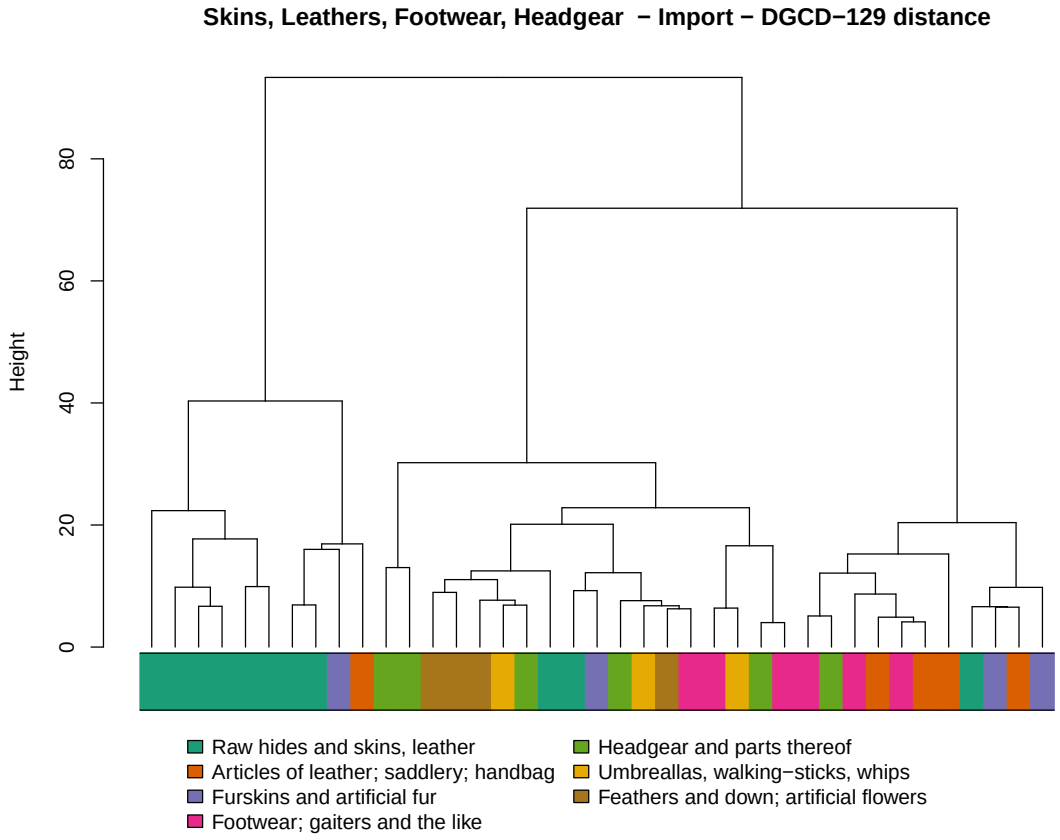
Finally, we consider together two sections which form a supply chain, namely *Raw hides, Skins and Leathers* and *Footwear and Headgear*. The first one contains three layers: *Raw hides and skins (other than furskins) and leather*; *Articles of leather; saddlery and harness; travel goods, handbags and similar containers; articles of animal gut (other than silk-worm gut)* and *Furskins and artificial fur; manufactures thereof*. The second section instead includes four layers: *Footwear; gaiters and the like; parts of such articles*; *Headgear and parts thereof*; *Umbrellas, sun umbrellas, walking-sticks, seat sticks, whips, riding crops; and parts thereof* and *Feathers and down, prepared; and articles made of feather or of down; artificial flowers; articles of human hair*. We have a total of 39 products. The aim of this investigation is to check whether products belonging to categories which form a supply chain, such as footwear and raw skins and leathers, share a common trade structure.

The dendrograms we obtained are displayed in Figure 3.25, while the corresponding heatmaps in Figure 3.26. The Cophenetic Coefficients of the two dendrograms are 0.798 in the Export case and 0.733 in the Import case, which are both large.

We observe in the Export case that the products belonging to the *Raw hides and skins (other than furskins) and leather* are almost all adjacent and six of them (*Tanned or crust skins of sheep or lambs, without wool on*; *Raw skins of sheep or lambs*; *Other raw hides and skins*; *Raw hides and skins of bovine (including buffalo) or equine animals*; *Tanned or crust hides and skins of other animals, without wool*



(a)



(b)

Figure 3.25: Dendrograms of the products in the *Skins and Leathers* and *Footwear and Headgear* HS sections.

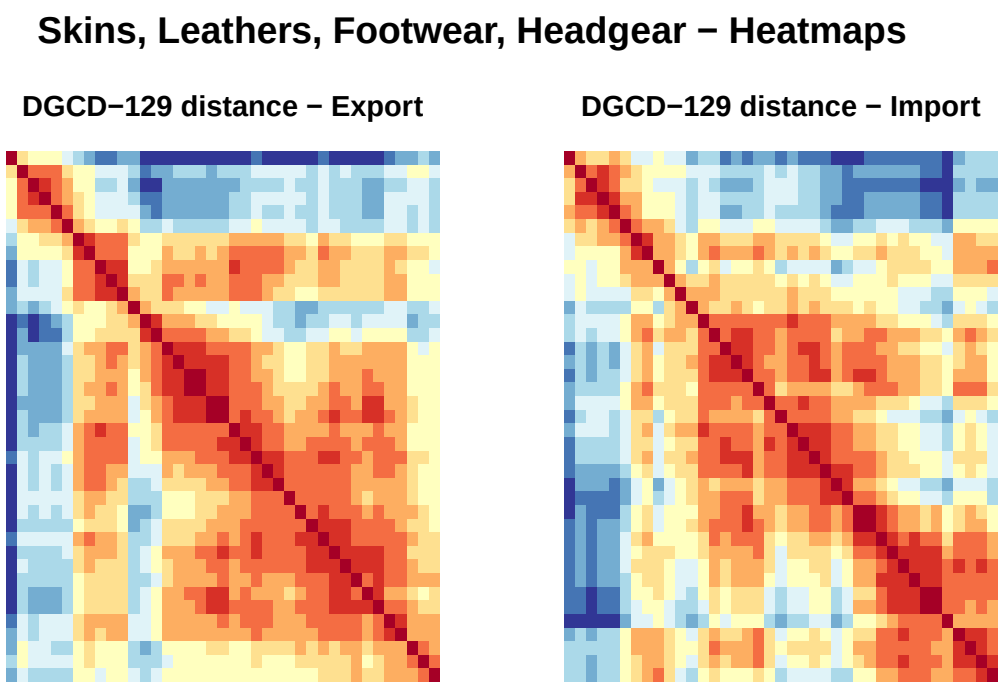


Figure 3.26: Heatmaps of the products in the *Skins and Leathers* and *Footwear and Headgear* HS sections.

or hair on and *Tanned or crust hides and skins of bovine (including buffalo)*) form a cluster well separated from all the other products. Other three products of the same category are adjacent to this cluster. From the heatmap, we see that there are other three products that are quite distant from all the others, and they are *Raw furskins (including heads, tails, paws and other pieces or cuttings*; *Trunks, suit-cases, vanity-cases, executive-cases, brief-cases* and *Footwear with outer soles of rubber, plastics, leather or composition leather*. Interestingly, we also find paired together the products *Artificial fur and articles thereof* and *Artificial flowers, foliage and fruit and parts thereof*. Our expectations are not met, since the products of the *Footwear; gaiters and the like; parts of such articles* layer are not adjacent to any of the products from the *Raw hides and skins (other than furskins) and leather*, which are the raw material needed for footwear production. Instead, we find such articles quite distant from each other and not grouped together. It is also interesting that the products in the *Raw hides and skins (other than furskins) and leather* are the only ones to have a clearly different structure from all the others, which instead are mixed up together. The Import case displays the same behaviour, with six products belonging to the *Raw hides and skins (other than furskins) and leather* layer which form a well separated cluster, with all other products mixed up together and with the footwear products distant from the raw skins and leathers.

As a last remark, even in this case we found a pronounced increase in the density of the graphs when switching from the Export to the Import case. In the first case the density ranges from 0.044 to 0.189, while in the second case from 0.065 to 0.299.

3.3.3 Results on Directed and Weighted dataset

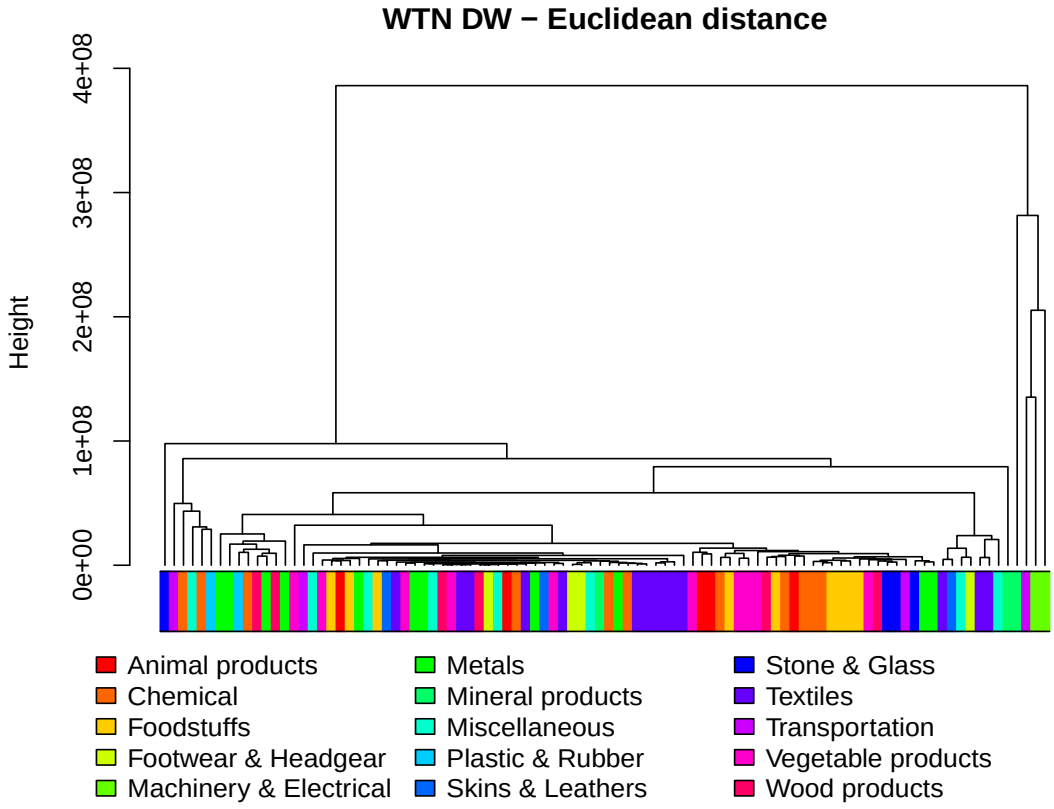
We show the heatmaps of the distance matrices obtained from the clustering analysis in Figure 3.28, the corresponding dendrograms in Figure 3.27 and their Cophenetic Correlation Coefficients in Table 3.9.

All Cophenetic Coefficients are large. We can see from the heatmaps that again there is a different behaviour among the distances which requires node correspondence. As in the FAO case study, the Euclidean and the Manhattan distances show a large cluster where all layers are very close to each other, and only few layers which instead are clearly separated from those ones. Instead, the Canberra distance shows a less pronounced clustering, with one cluster containing around half of the layers and another one less cohesive. The Portrait Divergence distance shows a division into two cluster, which anyway is not well defined, with a small group of layers which is the farthest from the others.

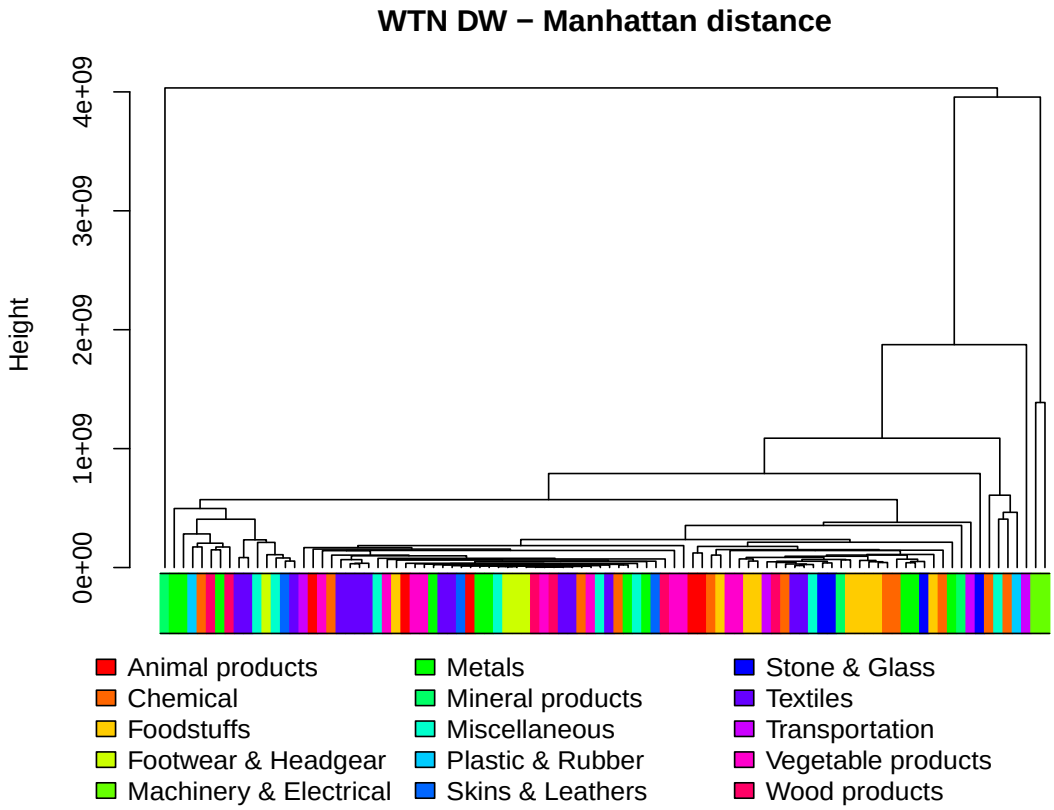
The most distant layers from all the others identified by the Euclidean distance (Figure 3.27a) are those gathered in a small cluster at the rightmost part of the dendrogram: *Mineral fuels, mineral oils and products of their distillation; bituminous substances; mineral waxes*; *Nuclear reactors, boilers, machinery and mechanical appliances; parts thereof*; *Electrical machinery and equipment and parts thereof; sound recorders and reproducers; television image and sound recorders and reproducers, parts and accessories of such articles* and *Vehicles; other than railway or tramway rolling stock, and parts and accessories thereof.* The Manhattan distance (Figure 3.27b) identifies the same layers as the ones farthest from all the others, with the difference that *Mineral fuels, mineral oils and products of their distillation; bituminous substances; mineral waxes* constitutes a single leaf. As in the FAO case study, we discover that these layers are those which have the highest trade volumes, in the order of the thousands of billion USD, and the layer representing mineral oils and fuels has the maximum trade volume, namely 2 864 billions USD. This again shows a strong dependence of these two distances on the values of the edge weights, such that they tend to put close each other layers with similar trade volumes. None of the supply chains identified in the Export and Import case is recovered here. The Canberra distance (Figure 3.27c), instead, again recovers the vehicles and high-technological instrument supply chain at the rightmost part of the dendrogram, gathering all the six layers which form the supply chain. This shows the importance of this supply chain also in terms of economical value, and not only

Table 3.9: Cophenetic Correlation Coefficients of the dendrograms obtained from the DW WTN dataset

Distance	DW
Euclidean	0.9639546
Manhattan	0.9736030
Canberra	0.6082103
Jaccard	0.5259792
Portrait Divergence	0.5953052

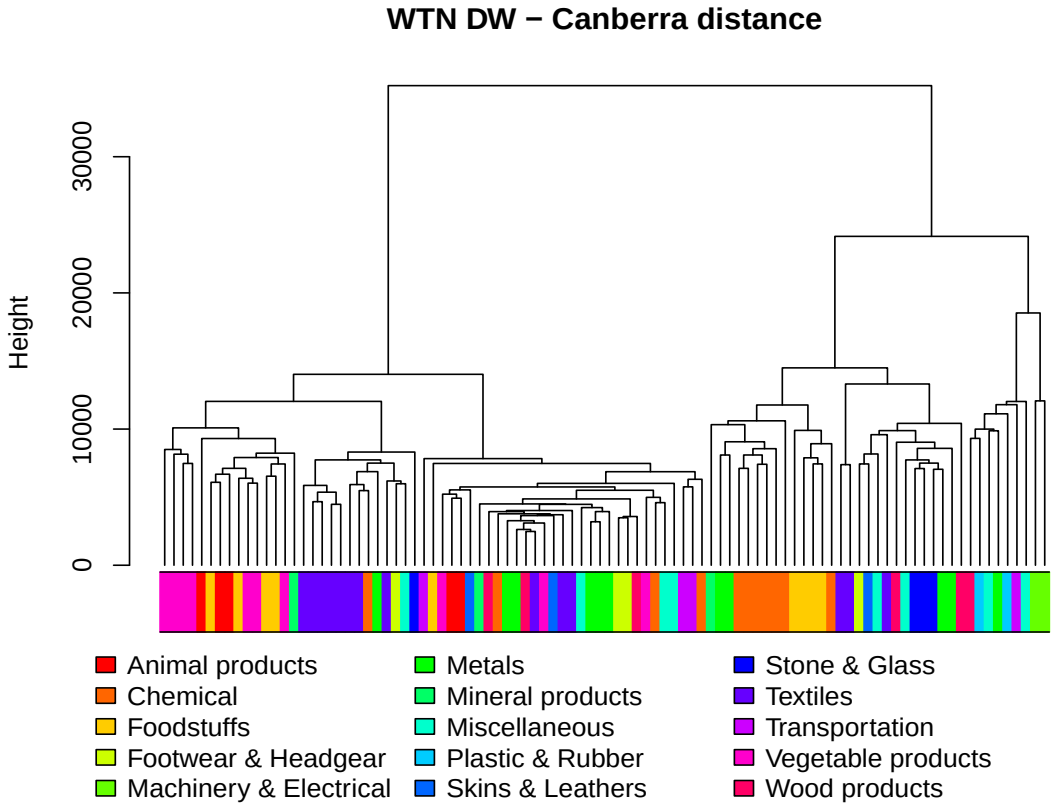


(a)

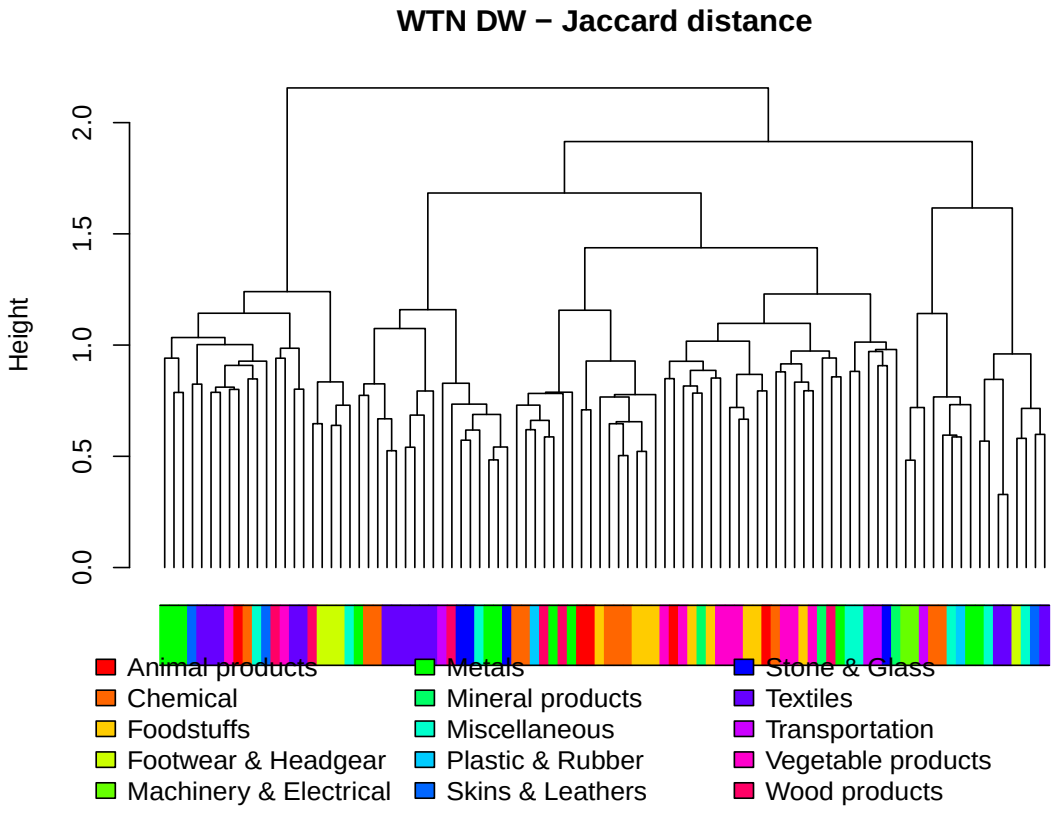


(b)

Figure 3.27: WTN DW dendrograms.

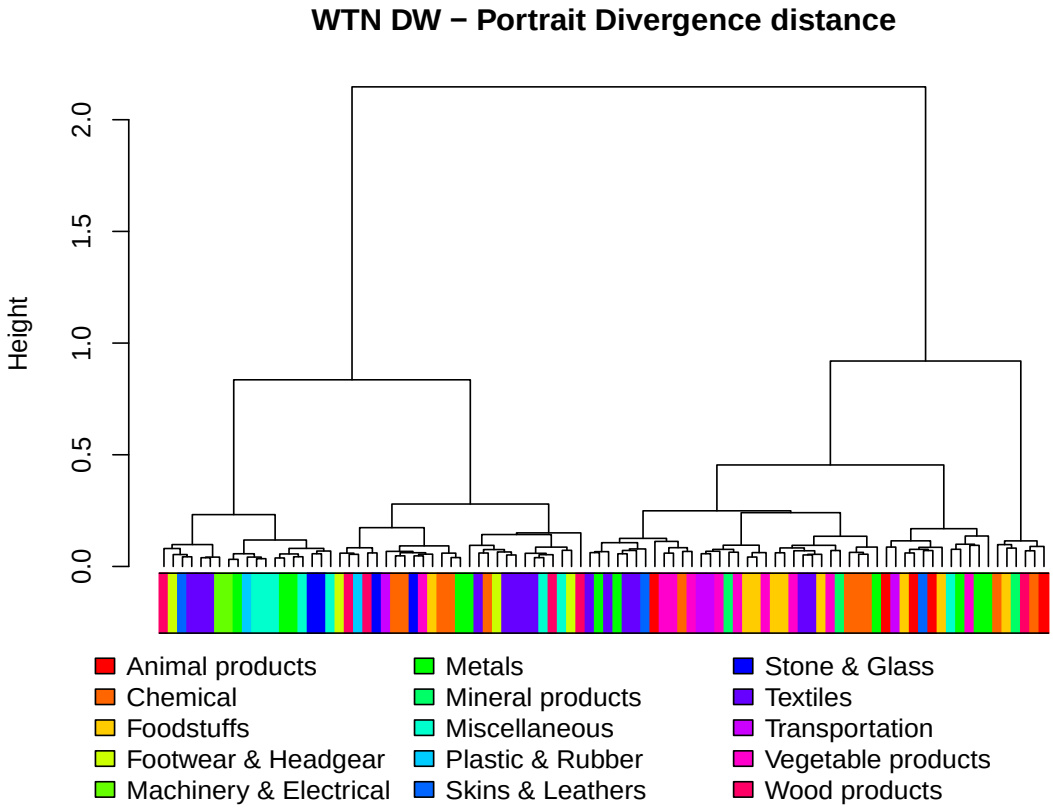


(c)



(d)

Figure 3.27 (cont.): WTN DW dendrograms.



(e)

Figure 3.27 (cont.): WTN DW dendrograms.

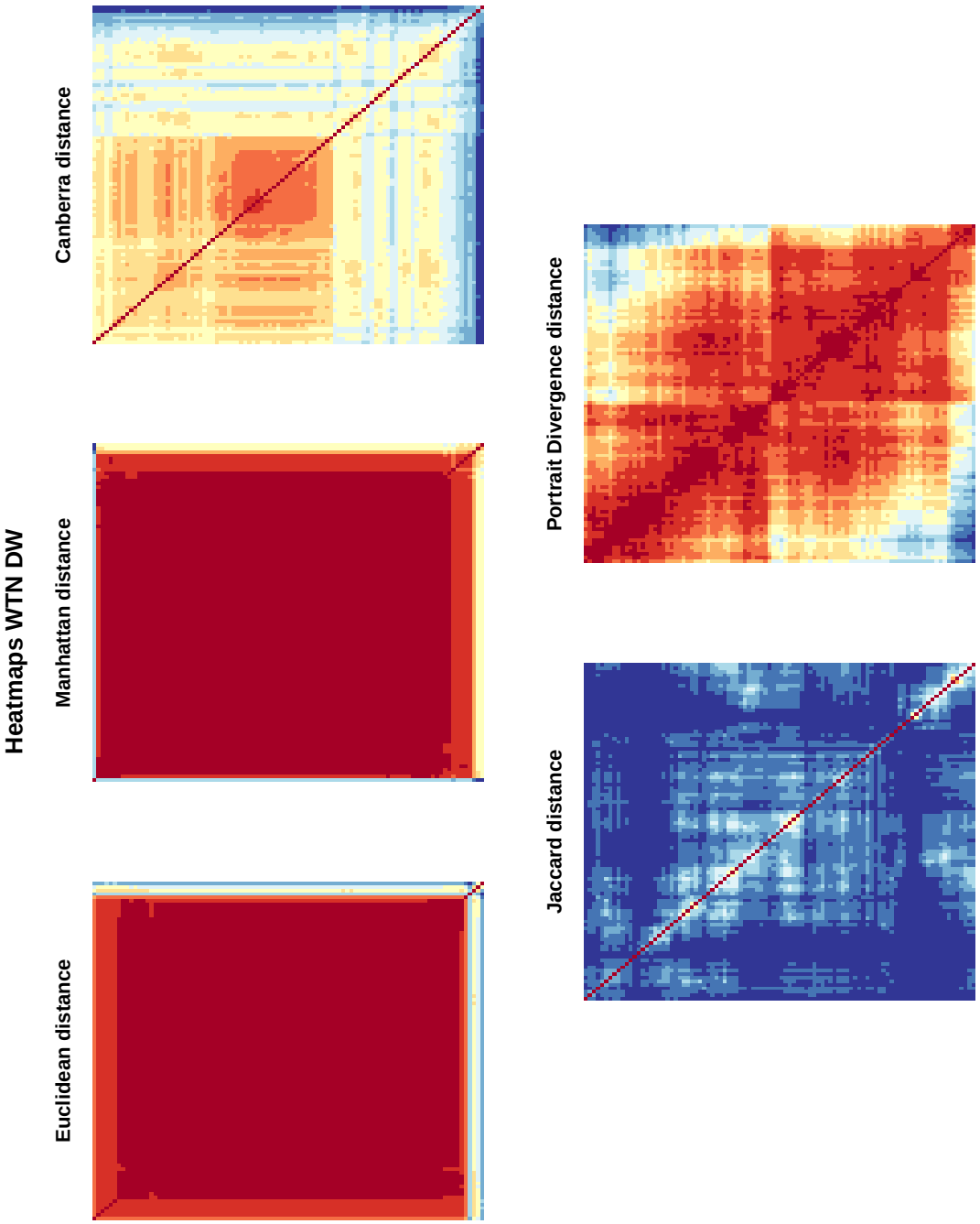


Figure 3.28: WTN DW heatmaps.

in terms of trade structure. Other supply chains are not present. As a last note, we observe that the Canberra distance shows a larger number of contiguous blocks, and with more layers than before, in the coloured bar under the dendrogram. This may indicate the fact that, even if the products in the same HS section have different trade patterns, they are also traded with similar proportions of trade volumes on the same connections.

As far as distances independent on node correspondence are concerned, the Portrait Divergence distance (Figure 3.27e) again gathers five out of the six layers of the vehicles and high-technological instruments supply chain, but the missing one is precisely the *Vehicles; other than railway or tramway rolling stock, and parts and accessories thereof* layer. Other supply chains cannot be identified.

Chapter 4

Conclusions and future developments

In this work, we reviewed the most interesting and promising methods that we found in literature for the task of network comparison, we evaluated their performances in idealized conditions and we used them to analyse real-world multilayer networks.

Analysing the existing literature, we proposed a novel classification of methods based on their functioning. The first class gathers all the methods which require *a priori* to know the correspondence between the nodes of the compared networks. Methods belonging to this class, such as the Euclidean, the Jaccard or the DeltaCon distances, are most suitable to evaluate the extent of the changes that occur in a graph, for instance due to temporal evolution or to some kind of failure, to spot anomalies or to check the differences between graphs representing various scenarios, like in the EU Air Transportation case study (Section 3.1). The second class gathers all the methods which do not require any *a priori* knowledge of a correspondence between the nodes. Methods in this class, such as the Spectral distances, the graphlet-based measures and the Portrait Divergence distance, are specifically suited for the comparison of the structure of the considered graphs, so that they give information about how much, and in what sense, the topology of the graphs differs.

We carried out two main analysis on synthetic (undirected/unweighted and directed/unweighted) networks to evaluate the performances of each method. The perturbation tests (Section 2.2) confirmed that all methods behave well, in the sense that they tend to zero as the similarity of the networks increases, that they tend to a plateau after a large number of perturbations and that they do not fluctuate too much if perturbations are repeated. Instead, the clustering tests (Section 2.3) - which can be carried out only on distances independent on node correspondence - highlighted different behaviour and performances. When networks of the same size and density are considered, in both the undirected/unweighted and the directed/unweighted case all the methods are able to discriminate between different network topologies, achieving a perfect classification in some cases. The Spectral SNL distance never achieves a perfect classification and it is the worst performing

method in all situations. Instead, when considering networks of different sizes and densities, the results change considerably. In the undirected/unweighted case, the graphlet-based measure GCD-11 is the best performing distance in discriminating between different network topologies and clearly outperforms the other methods. The Spectral SNL distance is the worst performing method, being only slightly better than a random classifier; the other two Spectral distances and Portrait Divergence have comparable performances. In the directed/unweighted case, the graphlet-based measure DGCD-129 is able to achieve an almost perfect classification, outperforming the Portrait Divergence distance, which performs better than in the undirected/unweighted case. Therefore, since in many real-world applications density and size of the graphs may vary considerably, graphlet-based measures are by far the most reliable tools to investigate the differences between networks structure.

The evaluation of the distances which require node correspondence is more difficult, since tests on synthetic networks to asses their performances are difficult to design. Nonetheless, they were able to recover important features that were expected in the real-world case studies that we analysed. For example, in the European Air Transportation case study, they paired at small distances many of the airlines based in the same nations, which are expected to have high nodes similarities since they share the same airports and routes. Another example comes from the FAO case study, in which distances which require node correspondence were able to group some specific products with common production country, due to climate reasons. A common feature of this class of methods that emerged from the analysis of real-world case studies is their strong dependence on density, size and total edge weight of the compared graphs, which is not in principle a weakness, but requires a deeper investigation.

In the analysis of the economical case studies concerning the directed and unweighted versions of the FAO and of the WTN datasets, some interesting economical aspects were highlighted. In general, network distances were able to recover expected behaviours, like the grouping of products with common production origin or the individuation of products with pronounced centralized structures in the FAO dataset, or like the presence of supply chains in the WTN dataset. Remarkable is the presence of products with different trade patterns inside some of the HS sections. As our analysis highlighted in the *Stone and Glass*, *Transportation* and *Raw hides, Skins and Leathers* sections, some products belonging to only one of the HS 2-digit layers of those sections share a common and peculiar trade structure which is different from the trade structure of all the other products in the same section. Moreover, neither in the FAO nor in the WTN datasets, none of the methods was able to recover the original classification into HS sections. These facts may denote that the current HS classification misses some economical aspects that strongly characterise the goods and their trading.

The analysis of the directed and weighted versions of the two datasets, instead, gave poor results. In economical networks, edge weights accounts for the amount of trade flows between countries, so that this is a significant information to be considered. Then, we argue that the scarce results are due to the lack of proper methods able to deal with directed and weighted networks and to the lack of an

exhaustive analysis of their behaviour on idealized conditions.

In this work we did not consider all the existent methods and all the possible experiments to test them, therefore many further developments and improvements can be carried out in the field of network comparison. Among them, we mention:

- the evaluation of the newest available methods, like GRAFENE and the Cut distance;
- an exhaustive analysis of the scalability of the methods;
- a full analysis of the tests on synthetic networks using many values of size and density or using other network topologies;
- the design of new tests aimed at clearly evaluating the performances of distances which require node correspondence;
- the design of new methods (especially those independent on node correspondence) suitable to compare directed and weighted networks and the design of proper and meaningful tests on synthetic networks to understand and evaluate the behaviour of such methods.

Appendix A

Basics of Graph Theory

A.1 Definitions and representations

Definition 1. — *A **graph** (or **network**) is a couple*

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \,,$$

where $\mathcal{V}$ is the set of nodes (or vertices) and $\mathcal{E}$ is the set of edges (or links) connecting the nodes.

A graph can be *directed*, if the direction of edges is taken into account, or *undirected*, if not. It can also be *weighted*, if a real-valued weight $w_{ij} > 0$ is associated to each edge (i, j), or *unweighted*, if not.

Definition 2. — *The **degree** d_i of a node i is the number of incident edges in node i.*

If the network is directed, an *in*-degree and an *out*-degree can be defined, counting the number of edges ending in node i and the number of edges starting from node i, respectively. If the network is weighted, an analogous indicator can be defined:

Definition 3. — *The **strength** of a node i is defined as the sum of the weights of the edges incident in node i:*

$$s_i = \sum_{j \in \mathcal{V}} w_{ij} \,.$$

If, other than weighted, the graph is also directed, an *in*-strength and an *out*-strength can be defined as before.

Undirected and unweighted graphs can be represented in matrix form in several ways, among which the most common and useful are the adjacency and the laplacian matrix.

Definition 4. — *The **adjacency matrix** of a graph $\mathcal{G}$ is a matrix $\mathbf{A} = [a_{ij}]$, $i, j = 1, 2, \ldots, N$, where*

$$a_{ij} = \begin{cases} 1 & \text{if the edge connecting node } i \text{ and node } j \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

Definition 5. — *The* ***laplacian matrix*** *of a graph $\mathcal{G}$ is a matrix $\mathbf{L} = [l_{ij}]$, $i, j = 1, 2, \ldots, N$, where*

$$l_{ij} = \begin{cases} d_i & \text{if } i{=}j \\ -1 & \text{if the edge connecting node } i \text{ and node } j \text{ exists} \\ 0 & \text{otherwise} \end{cases} .$$

Defining $\mathbf{D} = diag(d_1, \ldots, d_N)$ the matrix whose diagonal contains the degree of each node, we can define the laplacian matrix also as $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

Note that $\mathbf{A}$ and $\mathbf{L}$ are both symmetric. The definition of adjacency matrix can also be used to represent directed and unweighted graphs: in that case, the matrix is no more symmetric. Instead, in the case the graph is weighted, another matrix is defined for its representation:

Definition 6. — *The* ***weight matrix*** *of a weighted graph $\mathcal{G}$ is a matrix $\mathbf{W} = [w_{ij}]$, $i, j = 1, 2, \ldots, N$, where*

$$w_{ij} = \begin{cases} w_{ij} > 0 & \text{if the edge connecting node } i \text{ and node } j \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

This definition can be used to represent both undirected and directed weighted graphs; if the graph is undirected, $\mathbf{W}$ is symmetric.

An alternative matrix representation for both directed and undirected unweighted graphs is the following.

Definition 7. — *The* ***symmetric normalized laplacian*** $\mathbf{L}^*$ *is the normalization of* $\mathbf{L}$ *with respect to the nodes' degree, i.e.*

$$l^*_{ij} = \begin{cases} 1 & \text{if } i{=}j \\ -\frac{1}{\sqrt{d_i d_j}} & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{A.1}$$

An analogous definition of $\mathbf{L}^*$ is given by $\mathbf{L}^* = \mathbf{S}\mathbf{S}^T$, where $\mathbf{S}$ is the matrix whose rows are indexed by the nodes and whose columns are indexed by the edges of the graph, such that each column corresponding to the edge (i, j) has entry $\frac{1}{\sqrt{d_i}}$ in the row corresponding to node i and entry $-\frac{1}{\sqrt{d_j}}$ in the row corresponding to node j, and zero entries elsewhere [18]. This representation can also be used to represent directed graphs, but note that in this case the network is considered as undirected, since Equation (A.1) considers the total degree. A generalization to weighted graphs exists. Consider a weighted graph, either directed or undirected, without self-loops and with weight w_{ij} for the edge (i, j); then the symmetric normalized laplacian is defined as

$$l^*_{ij} = \begin{cases} 1 & \text{if } i{=}j \\ -\frac{w_{ij}}{\sqrt{d_i d_j}} & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} .$$

Also in this case $\mathbf{L}^*$ can be expressed in the form $\mathbf{L}^* = \mathbf{S}\mathbf{S}^T$, where now $\mathbf{S}$ has, in each column corresponding to edge (i, j), entry $\sqrt{\frac{w_{ij}}{d_i}}$ in the row corresponding to node i and entry $-\sqrt{\frac{w_{ij}}{d_j}}$ in the row corresponding to node j. Again, this representation can be applied to directed and weighted graphs, but they are actually considered as undirected and weighted.

Other important definitions to understand the structure of graphs are related to how the nodes are connected inside the network.

Definition 8. — *A network $\mathcal{G}$ is said to be **connected** if, for any pair of nodes i and j, there exists a path from i to j*

In other words, a graph is connected if from each node it is possible to reach each other node.

Definition 9. — *A maximal connected subgraph of $\mathcal{G}$ is a **component** of $\mathcal{G}$.*

Here maximal means that the component has not to be a proper subgraph of any other connected subgraphs of $\mathcal{G}$ [16]. When considering directed graphs, two types of connectivity are defined:

Definition 10. — *The **Strongly Connected Component (SCC)** is the set of nodes in which there is a directed path between each pair of nodes. The **Weakly Connected Component (WCC)** is the set of nodes in which there is a path between each pair of nodes, considering the edges as undirected.*

Note that only the SCC is a proper component.

A.2 Network properties

Many indicators exist to quantify and describe the characteristics of a network, each one capturing a different aspect of the network topology. For all the following definitions we will consider an undirected and unweighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with N nodes and L links.

Definition 11. — *The **distance** between node i and node j is the length of the shortest path connecting the two nodes.*

Definition 12. — *The **diameter** of a graph is the maximum distance in the graph.*

Definition 13. — *The **density** of a graph is the fraction of the actually existing links over all the possible existing links, which can be computed as*

$$\rho = \frac{2L}{N(N-1)}\,.$$

Note that the density for directed graphs is just $\rho = \frac{L}{N(N-1)}$.

Definition 14. — *The **clustering coefficient**, or **transitivity**, for a node i is the fraction of triangles that node i forms with its neighbours over all the possible triplets centred in i:*

$$c_i = \frac{\#\ of\ triangles}{\#\ of\ all\ triplets\ centered\ in\ i}\,.$$

Definition 15. — *The **global clustering coefficient** is the average of the clustering coefficients of all nodes of the graph:*

$$C = \frac{1}{N}\sum_{i\in\mathcal{V}} c_i\,.$$

The clustering coefficient is a measure of the local density of a graph, since it measures, for every node i, the tendency of the neighbour nodes to be connected each other, i.e. whether they cluster together. The global clustering coefficient gives an evaluation of this tendency for the whole graph. Extensions to directed and weighted graphs are possible.

Another important statistics about the graph is the degree distribution.

Definition 16. — *The **degree distribution** $P(k)$ is the fraction of nodes having exactly degree k:*

$$P(k) = \frac{\#\ of\ nodes\ with\ degree\ k}{N}\,.$$

In most practical cases the degree distribution has not a smooth behaviour when counting nodes with very high degree; to overcome this issue, that may cause numerical problem, a closely related quantity is defined.

Definition 17. — *The **cumulative degree distribution** is the fraction of nodes having at least degree k:*

$$\bar{P}(k) = \frac{\#\ of\ nodes\ with\ degree\ \geq k}{N}\,.$$

Definition 18. — *The **average degree** is defined as*

$$\langle k\rangle = \sum_k kP(k) = \frac{1}{N}\sum_{i\in\mathcal{V}} k_i = \frac{2L}{N}\,.$$

A.3 Network models

A.3.1 Erdős-Rényi model

The Erdős-Rényi model [29] is the prototype of random graphs. Two equivalent versions of the model exist. In the first one, called $G(N, M)$ model, a random graph is selected uniformly at random from all the graphs which have N nodes and M edges (which is the same as connecting M pairs of nodes extracted uniformly at random). In the second one, called $G(N, p)$ model, a random graph on N nodes is obtained by connecting all pairs of nodes with a given probability p. Programming

languages implement these two ways to generate Erdős-Rényi graphs.

The two versions turn out to have the same asymptotic properties as N grows to infinity:

- the typical scale of the nodes degree is the mean degree with small fluctuations around it, so that the resulting network is almost homogeneous;
- the degree distribution is Poisson distributed with parameter equal to the mean degree;
- the average distance grows logarithmically with N (small-world effect);
- the clustering coefficient is small and tends to 0 as N grows.

A.3.2 Barabási-Albert model

The Barabási-Albert model [11] is a scale-free model, meaning that the degree of the nodes does not have a typical scale. This kind of network is better suited to model real-world networks coming from many application fields. The generation of a Barabási-Albert network is an iterative procedure that follows a *preferential attachment* mechanism. It starts with m_0 nodes arbitrarily connected; at each step, a new node is added to the graph connecting it to $m \leq m_0$ existing nodes with a probability that is proportional to the degree that the existing nodes have. In this way, at each step m new edges are added to the graph, which are preferentially connected to nodes that already have a large number of connections. This results in a network which contains few nodes with large degree (called *hubs*) and many nodes with very few connections, so that the network is strongly heterogeneous and there is not a typical scale for the nodes degree.

The main properties of Barabási-Albert graphs when N tends to infinity are:

- the degree distribution has power law distribution $P(k) \approx k^{-3}$;
- also in this case the average distance grows logarithmically with N (small-world effect);
- again the clustering coefficient vanishes with N.

A.3.3 Lancichinetti-Fortunato-Radicchi model

The Lancichinetti-Fortunato-Radicchi (LFR) model was introduced in [52] to provide a new benchmark for testing community detection methods. Until then, only small and nearly homogeneous networks with constant size of communities were used, while real-world networks are known to have power law degree distribution and also the distribution of community sizes is broad and well approximated by a power law. The LFR model takes into account all these facts to generate community structured networks that better resembles real-world networks.

In the LFR model, both the degree and the community size distributions are assumed to be power law with exponent γ and β respectively. The number of nodes is denoted as N and the average degree as $\langle k \rangle$. The mixing parameter μ allows to tune the strength of the community structure: each node shares a fraction $1 - \mu$ of its links with nodes in its same community and a fraction μ with nodes outside its community, so that a high value of μ (> 0.5) denotes a weak community structure. The generation of an instance of undirected and unweighted network from the LFR model proceeds through the following steps.

1. All the nodes are assigned a degree from the power law distribution with exponent γ.
2. The sizes of communities are taken from the power law distribution with exponent β.
3. Nodes are randomly assigned to communities with an iterative procedure.
4. Some rewiring steps are performed to enforce the condition on the fraction of links μ shared by each node inside its community.

To generate directed and unweighted networks, the LFR model follows the same procedure [51]. The main difference is in the first step, since in this case the nodes are assigned the in-degree from a power law distribution with exponent γ and the out-degree from a δ distribution. The constraints needed in the undirected case are generalized to fit to the directed case.

The algorithm is shown to have linear relation between the computational time and the number of links in the graph, so that the LFR model can be used to generate large graphs (even 10^6 nodes) in reasonable time.

Appendix B

Precision-Recall analysis

A standard procedure to evaluate the performances of a probabilistic classifier in the task of recovering the original grouping of data is the Precision-Recall analysis [4]. Consider some data which are labelled, for simplicity, with one between two different labels, for instance *positive* and *negative*. A probabilistic classifier is a method which is able to assign to each datum a probability to belong to one of the two classes; the *positive* label is assigned to a datum when its assigned probability to belong to the *positive* class is larger than a certain threshold. The aim of the classifier is to use the data to infer the correct label that each datum has. The Precision-Recall analysis is then used to evaluate how well the classifier performed in recovering the correct labelling. First, a confusion matrix is built, whose entries are the number of correctly classified and wrongly classified data. In particular, we define:

- **True positives** (TP): it is the number of truly *positive* data which are correctly classified as *positive*;
- **True negatives** (TN): it is the number of truly *negative* data which are correctly classified as *negative*;
- **False positives** (FP): it is the number of truly *negative* data which are wrongly classified as *positive*;
- **False negatives** (FN): it is the number of truly *positive* data which are wrongly classified as *negative*;

Then, precision and recall can be computed from the entries of the confusion matrix:

$$precision = \frac{TP}{TP + FP}\,, \qquad recall = \frac{TP}{TP + FN}\,.$$

The precision represents the fractions of the correctly classified *positive* data over all the data that the classifier labelled as *positive*. In the information retrieval context, precision represents the percentage of actually good documents that have been shown as a result. The recall instead represents the fractions of the correctly classified *positive* data over all the existing *positive* data. In the information retrieval context, recall represents the percentage of good documents shown with respect to

all the existing good documents. The higher precision and recall, the better the classifier: the ideal classifier will have both indexes equal to 1.

An important variable to determine the values of precision and recall is the thresholds with respect to which labels are assigned. Think for instance to a classifier which labels as *positive* only data which have a probability larger than 0.9 to be *positive*: this leads to an increase of the precision (*positive* labelled data are much likely to be truly *positive*), and to a decrease of the recall, since the number of false negatives increases. Conversely if the threshold is low. Then, we can analyse the performances of the classifier by varying the threshold; for each value of the thresholds, precision and recall can be computed and the Precision-Recall curve can be drawn. The Precision-Recall curve of an ideal classifier is a straight horizontal line, which corresponds to precision 1 for each value of recall. The Precision-Recall curve can be used to optimize the value of the threshold used in the classifier; it can also be used to evaluate the overall performances of the classifier for any threshold, by computing the Area Under the Precision-Recall curve (AUPR), also called average precision. The AUPR ranges in [0,1] and the higher the AUPR, the better the classifier, since it gets close to the ideal situation. In general, when more than one classifier is available for the task, we can evaluate which one performs best by looking at their AUPR.

The Precision-Recall analysis framework can be extended to the evaluation of the network distance measures, as done in [69, 77, 78]. In this context, a pair of graphs belong to the *positive* class if they come from the same network model and to the *negative* class otherwise. The aim of the network distance is to correctly retrieve pairs of graphs which come from the same network model. To do this, a distance threshold $\epsilon > 0$ is set and all the pairs of networks whose distance is smaller than ϵ are labelled as coming from the same network family. This allows to define a confusion matrix in an analogous way as previously explained. We obtain many values of precision and recall by varying the threshold ϵ whit small increments, and the Precision-Recall curve can be drawn. Then, by computing the AUPR we have an overall evaluation of the network distance performance in the task of clustering different network topologies. By comparing the AUPR of each method, we can conclude which one has the best performances.

Appendix C

Computational environment

All the computations were carried out on a virtual machine mounting Ubuntu 16.04.4 32bit and providing 6 GB of RAM and 2 CPU from the host system, which is a Windows 10 machine with 16 GB of RAM and an AMD A10-9600P RADEON R5 processor with 4 cores at 2.40 GHz. The analysis were run within a R framework: we used RStudio version 1.1.453 and R version 3.4.4. Our own code is written in R, while the code from other sources is written in different languages, mainly python, C++ and R. Python version 3.5.2 is used. The C++ compiler used is gcc version 5.4.0.

We used the following R packages to handle data and to analyse the results:

- *igraph* [20]: to generate and handle graphs.
- *dendextend* [33]: used for the cluster analysis, to handle dendrograms and plot them.
- *RColorBrewer* [61]: to produce nicer colours in some plots.
- *seriation* [41]: to reorder the leaves of the dendrograms with Optimal Leaf Ordering [10].
- *PRROC* [38]: to perform Precision-Recall analysis [4].

Bibliography

[1] Waqar Ali, Tiago Rito, Gesine Reinert, Fengzhu Sun, and Charlotte M. Deane. "Alignment-free protein interaction network comparison". In: *European Conference on Computational Biology* 30 (2014), pp. i430–i437. DOI: 10.1093/bioinformatics/btu447.

[2] Waqar Ali, Anatol E. Wegner, Robert E. Gaunt, Charlotte M. Deane, and Gesine Reinert. "Comparison of large networks with sub-sampling strategies". In: *Scientific Reports*. Vol. 6. 2016, p. 28955. DOI: 10.1038/srep28955.

[3] David Aparıcio, Pedro Ribeiro, and Fernando Silva. "Network comparison using directed graphlets". In: *ArXiv: 1511.01964* (2015).

[4] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[5] J. P. Bagrow, E. M. Bollt, J. D. Skufca, and D. ben-Avraham. "Portraits of complex networks". In: *Europhysics Letters* 81 (2008), p. 68004. DOI: 10.1209/0295-5075/81/68004.

[6] James P. Bagrow and Erik M. Bollt. "An information-theoretic, all-scales approach to comparing networks". In: *arXiv:1804.03665* (2018).

[7] Bela Balassa. "Trade Liberalisation and Revealed Comparative Advantage". In: *The Manchester School* 33 (1965), pp. 99–123. DOI: 10.1111/j.1467-9957.1965.tb00050.x.

[8] Richard Baldwin and Javier Lopez-Gonzalez. "Richard Baldwin & Javier Lopez-Gonzalez, 2015. "Supply-chain Trade: A Portrait of Global Patterns and Several Testable Hypotheses,"" in: *The World Economy* 38 (2015), pp. 1682–1721.

[9] Richard Baldwin and Anthony J. Venables. "Spiders and snakes: Offshoring and agglomeration in the global economy". In: *Journal of International Economics* 90 (2013), pp. 245–254. DOI: 10.1016/j.jinteco.2012.09.

[10] Ziv Bar-Joshep, David k. Gifford, and Tommi S. Jaakkola. "Fast optimal leaf ordering for hierarchical clustering". In: *Bioinformatics* 17 (2001), pp. 22–29.

[11] Albert-Laszlo Barabasi and Reka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286 (1999), pp. 509–512. DOI: 10.1126/science.286.5439.509.

[12] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiments* 10 (2008), p. 10008. DOI: 10.1088/1742-5468/2008/10/P10008.

[13] Horst Bunke, Peter J. Dickinson, Miro Kraetzl, and Walter D. Wallis. *A Graph-Theoretic Approach to Enterprise Network Dynamics (Progress in Computer Science and Applied Logic (PCS))*. Birkhauser, 2006. DOI: 10.1007/978-0-8176-4519-9.

[14] Alessio Cardillo, Jesus Gomez-Gardenes, Massimiliano Zanin, Miguel Romance, David Papo, Francisco del Pozo, and Stefano Boccaletti. "Emergence of network features from multiplexity". In: *Scientific Reports* 3 (2013), p. 1344. DOI: doi:10.1038/srep01344.

[15] *CEPII-BACI database*. 2014. URL: http://www.cepii.fr/CEPII/en/bdd_modele/presentation.asp?id=1.

[16] Gary Chartrand and Ping Zhang. *A first course in graph theory*. Courier Corporation, 2012.

[17] Flavio Chierichetti, Ravi Kumar, Sandeep Pandey, and Sergei Vassilvitskii. "Finding the Jaccard Median". In: *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2010, pp. 293–311. DOI: 10.1137/1.9781611973075.25.

[18] F. R. K. Chung. *Spectral Graph Theory*. Vol. CMBS. 92. American Mathematical Society, 1997. DOI: 10.1090/cbms/092.

[19] S. A. Cook. "The Complexity of Theorem-proving Procedures". In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. 1971, pp. 151–158. DOI: 10.1145/800157.805047.

[20] Gabor Csardi and Tamas Nepusz. "The igraph software package for complex network research". In: *InterJournal* Complex System (2006), p. 1695.

[21] Jesse Davis and Mark Goadrich. "The Relationship Between Precision-Recall and ROC Curves". In: *Proceedings of the 23rd International Conference on Machine Learning*. 2006, pp. 233–240. DOI: 10.1145/1143844.1143874.

[22] Luca De Benedictis and Lucia Tajoli. "Comparing sectoral international trade networks". In: *Aussenwirtschaft* 65 (2010), pp. 167–189.

[23] Luca De Benedictis and Lucia Tajoli. "The World Trade Network". In: *The World Economy* 34 (2009), pp. 1417–1454. DOI: 10.1111/j.1467-9701.2011.01360.x.

[24] Alan V. Deardorffs. *Deardorffs' Glossary of International Economics*. 2016. URL: http://www-personal.umich.edu/~alandear/glossary/.

[25] Manlio De Domenico. *CoMuNe Lab*. 2017. URL: https://comunelab.fbk.eu/data.php.

[26] Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. "Structural reducibility of multilayer networks". In: *Nature Communications* (2015), p. 6864. DOI: 10.1038/ncomms7864.

[27] Daniele Durante, David B. Dunson, and Joshua T. Vogelstein. "Nonparametric Bayes Modeling of Populations of Networks". In: *Journal of the American Statistical Association* 112 (2017), pp. 1516–1530. DOI: 10.1080/01621459.2016.1219260.

[28] Frank Emmert-Streib, Matthias Dehmer, and Yongtang Shi. "Fifty years of graph matching, network alignment and network comparison". In: *Information Sciences* 346-347 (2016), pp. 180–197. DOI: 10.1016/j.ins.2016.01.074.

[29] P. Erdös and A. Rényi. "On random graphs, I". In: *Publicationes Mathematicae (Debrecen)* 6 (1959), pp. 290–297.

[30] Fazle E. Faisal, Khalique Newaz, Julie L. Chaney, Jun Li, Scott J. Emrich, Patricia L. Clark, and Tijana Milenković. "GRAFENE: Graphlet-based alignment-free network approach integrates 3D structural and sequence (residue order) data to improve protein structural comparison". In: *Scientific Reports* 7 (2017), p. 14890. DOI: 10.1038/s41598-017-14411-y.

[31] FAO. *Detailed Trade Matrix*. 2018. URL: http://www.fao.org/faostat/en/#data/TM.

[32] Santo Fortunato. "Community detection in graphs". In: *Physics Reports* 486 (2010), pp. 75–174. DOI: 10.1016/j.physrep.2009.11.002.

[33] Tal Galili. "dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering". In: *Bioinformatics* (2015), pp. 3718–3720. DOI: 10.1093/bioinformatics/btv428.

[34] T. Gärtner, P. Flach, and S. Wrobel. "On Graph Kernels: Hardness Results and Efficient Alternatives". In: *Learning Theory and Kernel Machines. Lecture Notes in Computer Science* 2777 (2003), pp. 129–143. DOI: 10.1007/978-3-540-45167-9_11.

[35] Ralucca Gera, L. Alonso, Brian Crawford, Jeffrey House, J. A. Mendez-Bermudez, Thomas Knuth, and Ryan Miller. "Identifying network structure similarity using spectral graph theory". In: *Applied Network Science* 3.2 (2018), p. 2. DOI: 10.1007/s41109-017-0042-3.

[36] *Global Value Chain Development Report 2017: Measuring and Analyzing the Impact of GVCs on Economic Development*. World Trade Organization, 2017.

[37] C. D. Godsil and B. D. McKay. "Constructing cospectral graphs". In: *Aequationes Mathematicae* 25 (1982), pp. 257–268. DOI: 10.1007/BF02189621.

[38] Jan Grau, Ivo Grosse, and Jens Keilwagen. "PRROC: computing and visualizing precision-recall and receiver operating characteristic curves in R". In: *Bioinformatics* 31 (2015), pp. 2595–2597.

[39] Robert H.A. Sneath Peter & R. Sokal. "Numerical Taxonomy. The Principles and Practice of Numerical Classification". In: *Taxon* 12 (1963), pp. 190–199. DOI: 10.2307/1217562.

[40] Willem H. Haemers and Edward Spence. "Enumeration of cospectral graphs". In: *European Journal of combinatorics* 25 (2004), pp. 199–211.

[41] Michael Hahsler, Christian Buchta, and Kurt Hornik. *seriation: Infrastructure for Ordering Objects Using Seriation*. R package version 1.2-3. 2018.

[42] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. The MIT Press, 2001.

[43] Tomaž Hočevar and Janez Demšar. "A combinatorial approach to graphlet counting". In: *Bioinformatics* 30.4 (2014), pp. 559–565. DOI: 10.1093/bioinformatics/btt717.

[44] Sergey Ioffe. "Improved Consistent Sampling, Weighted Minhash and L1 Sketching". In: *Proceedings of the 2010 IEEE International Conference on Data Mining*. 2010, pp. 246–255. DOI: 10.1109/ICDM.2010.80.

[45] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. "Multilayer networks". In: *Journal of Complex Networks* 2 (2014), pp. 203–271. DOI: 10.1093/comnet/cnu016.

[46] Danai Koutra, Tai-You Ke, U. Kang, Duen Chau, Hsing-Kuo Pao, and Christos Faloutsos. "Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms". In: *Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*. Vol. 6912. 2011, pp. 245–260. DOI: 10.1007/978-3-642-23783-6_16.

[47] Danai Koutra, Neil Shah, Joshua T. Vogelstein, Brian Gallagher, and Christos Faloutsos. "DELTACON: A Principled Massive-Graph Similarity Function with Attribution". In: *Transaction on Knoledge Discovery from Data* 10 (2016), pp. 1–43. DOI: 10.1145/2824443.

[48] Danai Koutra, Joshua T. Vogelstein, and Christos Faloutsos. "DELTACON: A Principled Massive-Graph Similarity Function". In: *Proocedings of the 2013 SIAM International Conference on Data Mining* (2013), pp. 162–170.

[49] Oleksii Kuchaiev, Tijana Milenković, Vesna Memišević, Wayne Hayes, and Nataša Pržulj. "Topological network alignment uncovers biological function and phylogeny". In: *Journal of The Royal Society Interface* (2010), pp. 1341–1354. DOI: 10.1098/rsif.2010.0063.

[50] Oleksii Kuchaiev and Nataša Pržulj. "Integrative network alignment reveals large regions of global network similarity in yeast and human". In: *Bioinformatics* 27 (2011), pp. 1390–1396. DOI: 10.1093/bioinformatics/btr127.

[51] Andrea Lancichinetti and Santo Fortunato. "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities". In: *Physical Review E* 80 (2009), p. 016118. DOI: 10.1103/PhysRevE.80.016118.

[52] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. "Benchmark graphs for testing community detection algorithms". In: *Physical Review E* 78 (2008), p. 046110. DOI: 10.1103/PhysRevE.78.046110.

[53] Hyekyoung Lee, Moo K. Chung, Hyejin Kang, Boong-Nyun Kim, and Dong Soo Lee. "Computing the Shape of Brain Networks using Graph Filtration and Gromov-Hausdorff Metric". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2011), pp. 302–309.

[54] Qun Liu, Zhishan Dong, and En Wang. "Cut Based Method for Comparing Complex Networks". In: *Scientific Reports* 8 (2018), p. 5134. DOI: 10.1038/s41598-018-21532-5.

[55] Noel Malod-Dognin and Natasa Przulj. "GR-Align: fast and flexible alignment of protein 3D structures using graphlet degree similarity". In: *Bionformatics* 30 (2014), pp. 1259–1265. DOI: 10.1093/bioinformatics/btu020.

[56] Marina Meilă and William Pentney. "Clustering by weighted cuts in directed graphs". In: *Proceedings of the 2007 SIAM International Conference on Data Mining*. 2007, pp. 135–144. DOI: 10.1137/1.9781611972771.13.

[57] Tijana Milenković, Weng Leong Ng, Wayne Hayes, and Nataša Pržulj. "Optimal Network Alignment with Graphlet Degree Vectors". In: *Cancer Informatics* 9 (2010), pp. 121–137. DOI: 10.4137/CIN.S4744.

[58] Tijana Milenković and Nataša Przulj. "Uncovering Biological Network Function via Graphlet Degree Signatures". In: *Cancer Informatics* 6 (2008), pp. 257–273. DOI: 10.4137/CIN.S680.

[59] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. "Network Motifs: Simple Building Blocks of Complex Networks". In: *Science* 298.5594 (2002), pp. 824–827. DOI: 10.1126/science.298.5594.824.

[60] Fionn Murtagh and Pierre Legendre. "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?" In: *Journal of Classification* 31 (2014), pp. 274–295. DOI: 10.1007/s00357-014-9161-z.

[61] Erich Neuwirth. *RColorBrewer: ColorBrewer Palettes*. R package version 1.1-2. 2014.

[62] Jukka-Pekka Onnela, Daniel J.Fenn, Stephen Reid, Mason A. Porter, Peter J. Mucha, Mark D. Fricker, and Nick S. Jones. "Taxonomies of networks from community structure". In: *Physical Review E* 86 (3 2012), p. 036104. DOI: 10.1103/PhysRevE.86.036104.

[63] Panagiotis Papadimitriou, Ali Dasdan, and Hector Garcia-Molina. "Web graph similarity for anomaly detection". In: *Journal of Internet Services and Applications* 1 (2010), pp. 19–30. DOI: 10.1007/s13174-010-0003-x.

[64] Carlo Piccardi. "Finding and Testing Network Communities by Lumped Markov Chains". In: *PLOS ONE* 6 (2011), pp. 1–13. DOI: 10.1371/journal.pone.0027028.

[65] Carlo Piccardi and Lucia Tajoli. "Complexity, Centralization and Fragility in Economics Networks". In: *arXiv:1802.08575* (2018).

[66] N. Pržulj, D. G. Corneil, and I. Jurisica. "Modeling interactome: scale-free or geometric?" In: *Bioinformatics* 20 (2004), pp. 3508–3515. DOI: 10.1093/bioinformatics/bth436.

[67] Natasa Przulj. "Biological network comparison using graphlet degree distribution". In: *Bioinformatics* 23 (2007), pp. 177–183. DOI: 10.1093/bioinformatics/btl301.

[68] Pedro Ribeiro and Fernando Silva. "G-Tries: A Data Structure for Storing and Finding Subgraphs". In: *Data Mining and Knowledge Discovery* 28 (2014), pp. 337–377. DOI: 10.1007/s10618-013-0303-4.

[69] Anida Sarajlić, Noël Malod-Dognin, Ömer Nebil Yaveroğlu, and Nataša Pržulj. "Graphlet-based Characterization of Directed Networks". In: *Scientific Reports* 6 (2016), p. 35098. DOI: doi:10.1038/srep35098.

[70] Nino Shervashidze, S. V. N. Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. "Efficient graphlet kernels for large graph comparison". In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*. Vol. 5. Proceedings of Machine Learning Research. 2009, pp. 488–495.

[71] Robert R. Sokal and F. James Rohlf. "The Comparison of Dendrograms by Objective Methods". In: *Taxon* 11 (1962), pp. 33–40. DOI: 10.2307/1217208.

[72] C. Spearman. "The proof and measurment of association between two things". In: *The American Journal of Psychology* 15 (1904), pp. 72–101. DOI: 10.2307/1412159.

[73] United Nations International Trade Statistics. *Harmonized Commodity Description and Coding Systems (HS)*. 2017. URL: https://unstats.un.org/unsd/tradekb/Knowledgebase/50018/Harmonized-Commodity-Description-and-Coding-Systems-HS.

[74] *The State of Agricultural Commodity Markets 2015-2016*. Food and Agricultur Organization of the United Nations, 2015.

[75] Seinosuke Toda. "Graph Isomorphism: Its Complexity and Algorithms". In: *Foundations of Software Technology and Theoretical Computer Science*. Springer Berlin Heidelberg, 1999, pp. 341–341. DOI: 10.1007/3-540-46691-6_27.

[76] Richard C. Wilson and Ping Zhu. "A Study of Graph Spectra for Comparing Graphs and Trees". In: *Pattern Recognition* 41 (2008), pp. 2833–2841. DOI: 10.1016/j.patcog.2008.03.011.

[77] Omer Nebil Yaveroglu, Noel Malod-Dognin, Darren Davis, Zoran Levnajic, Vuk Janjic, Rasa Karapandza, Aleksandar Stojmirovic, and Natasa Przulj. "Revealing the Hidden Language of Complex Networks". In: *Scientific Reports* 4 (2014), p. 4547. DOI: 10.1038/srep04547.

[78] Omer Nebil Yaveroglu, Tijana Milenkovic, and Natasa Przulj. "Proper evaluation of alignment-free network comparison methods". In: *Bioinformatics* 31 (2015), pp. 2697–2704.