Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Enzyme characterization via spectral analysis of the Laplacian associated to their contact maps

Relator:
Prof. Daniel Remondini

Presented by:
Alberto Amaduzzi

Co-relator:
Dott.Giulia Menichetti

Anno Accademico 2020/2021

Contents

Abstract

The main motivation for my thesis is the belief that global properties of enzymes are essential for a complete understanding of their behaviors. In my thesis, in particular, I investigate qualitative properties of enzymes via spectral techniques associated to the graph Laplacian. I try to apply visualization techniques to understand similarities and dissimilarities among different enzymes' structures, encoded in adjacency matrices retrieved from coordinate data in online available datasets. The purpose is to make an exploration of features and see whether these techniques, that are used extensively in literature for visual discrimination tasks, are also useful for these biological entities.

I have tried to design a size-independent analysis that would be able to differentiate among different taxonomies, different catalytic properties and different environments associated to enzymes. This attempt provided useful hints for the analysis of enzyme properties, even if as a final remark the dependence from enzyme size is still found in the Laplacian eigenvalue spectrum.

Chapter 1

Introduction

1.1 Complex systems

In the last decades there has been a significant rise in the collection of data describing features of very complex systems, most of which have yet to be analyzed. Thanks to the new technologies now available, it is possible now to approach them. A particular subset of complex systems is without any doubt provided by proteins in biological systems. There are many freely available datasets online related to proteins and their biochemical interactions (Uniprot, Swissprot, Pdb, pdbe, Swissmodel, Mint, intact ecc.). The trend is captured in fig.1.1 that shows just how PDB has increased its repositories in these years. PDB is an international repository for structural 3D data of enzymes and nucleic acids. In the analysis of these systems it has been seen a progressively increase in the use of techniques borrowed from statistical physics as in [18] and information theory on graphs as in [21] . These complex systems are suited for statistical inference, and interesting results about correlation between qualitative labels and quantitative measures on graphs are being discovered. In particular we will focus on the use of networks as suggested in [22]. A network is a set of items, which we will call vertices or sometimes nodes, with connections between them, called edges Figure 1.2.

Systems taking the form of networks (also called "graphs" in much of the mathematical literature) abound in the world. Examples include the Internet, the World Wide Web, social networks of acquaintance or other connections between individuals, organizational networks and networks of business relations between companies, neural networks, metabolic networks, food webs, distribution networks such as blood vessels or postal delivery routes, networks of citations between papers, and many others.The study of networks, in the form of mathematical graph theory, is one of the fundamental pillars of discrete mathematics. Euler's celebrated 1735 solution of the Konigsberg bridge problem is often cited as the first true proof in the theory of networks, and during the twentieth century graph theory has developed into a substantial body of knowledge. Networks have also been studied extensively in the social sciences. Typical network studies in sociology involve the circulation of questionnaires, asking respondents to detail their interactions with others. One can then use the responses to reconstruct a network in which vertices represent individuals and edges the interactions between them. Typical social network studies address issues of centrality (which individuals are best connected to others or have most influence) and connectivity (whether and how individuals are connected to one another through the network). In recent years it has been developed a line of research that focus shifting away from the analysis of single small graphs and the properties of

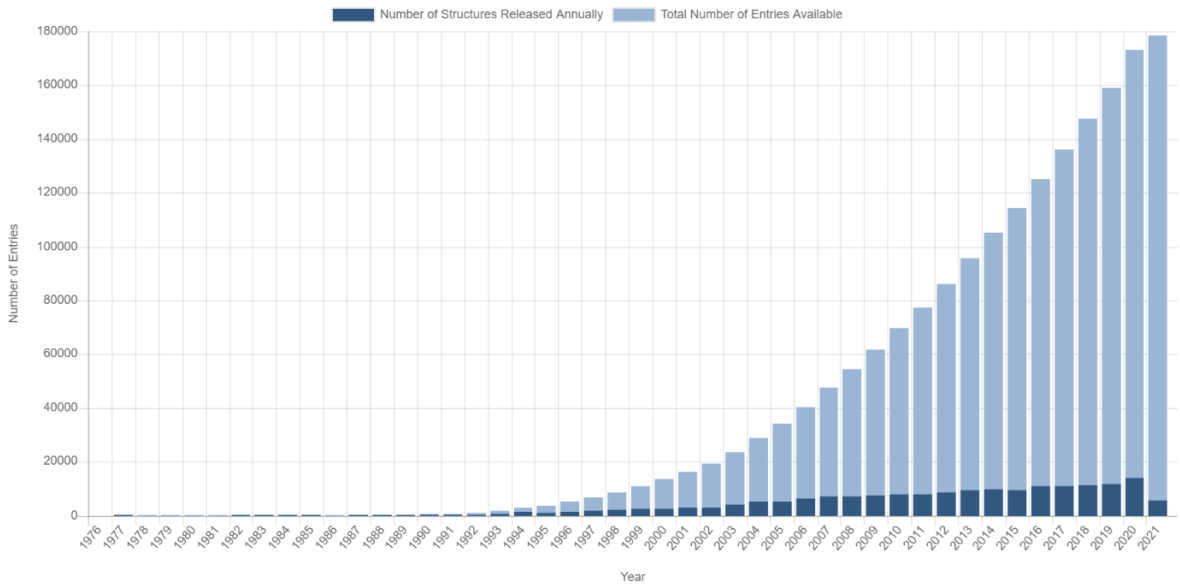


Figure 1.1: Number of new structures uploaded every year in PDB

individual vertices or edges within such graphs to consideration of large-scale statistical properties of graphs. This new approach has been driven largely by the availability of computers and communication networks that allow us to gather and analyze data on a scale far larger than previously possible. Where studies used to look at networks of maybe tens or in extreme cases hundreds of vertices, it is not uncommon now to see networks with millions or even billions of vertices. This change of scale forces upon us a corresponding change in our analytic approach. Many of the questions that might previously have been asked in studies of small networks are simply not useful in much larger networks. A social network analyst might have asked, "Which vertex in this network would prove most crucial to the network's connectivity if it were removed?" But such a question has little meaning in most networks of a million vertices—no single vertex in such a network will have much effect at all when removed. On the other hand, one could reasonably ask a question like, "What percentage of vertices need to be removed to substantially affect network connectivity in some given way?" and this type of statistical question has real meaning even in a very large network. However, there is another reason why our approach to the study of networks has changed in recent years, a reason whose importance should not be underestimated, although it often is. For networks of tens or hundreds of vertices, it is a relatively straightforward matter to draw a picture of the network with actual points and lines (Fig. 2) and to answer specific questions about network structure by examining this picture. This has been one of the primary methods of network analysts since the field began. The human eye is an analytic tool of remarkable power, and eyeballing pictures of networks is an excellent way to gain an understanding of their structure. With a network of a million or a billion vertices however, this approach is less useful, particularly at small-scale detail

One simply cannot draw a meaningful picture of a million vertices, even with modern 3D computer rendering tools, and therefore direct analysis by eye is hopeless. The recent development of statistical methods for quantifying large networks is to a large extent an attempt to find something to play the part played by the eye in the network analysis of the twentieth century. Statistical methods answer the question, "How can I tell what this network looks like, when I can't actually look at it?". There exist as already said a big

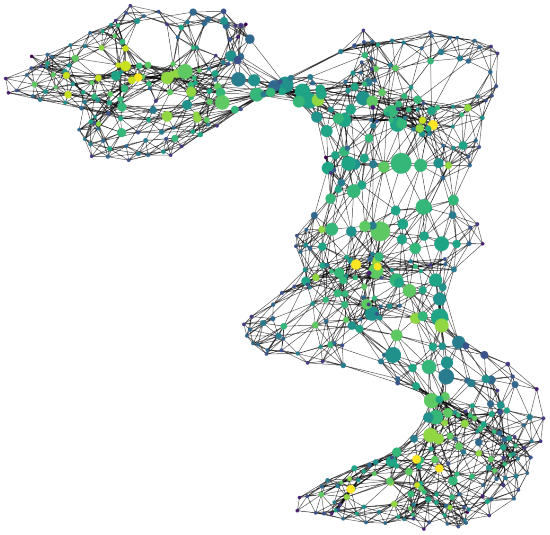


Figure 1.2: Graph associated to the contact map coming from the structure of 3YCA. In here the colors represent the betweennes centrality of the node while the size is the connectivity.

variety of different networks and I will describe them in the next sections, here, I want to introduce something about biological complex network to introduce the framework in which my study lies. A number of biological systems can be usefully represented as networks. Perhaps the classic example of a biological network is the network of metabolic pathways, which is a representation of metabolic substrates and products with directed edges joining them if a known metabolic reaction exists that acts on a given substrate and produces a given product. separate network is the network of mechanistic physical interactions between proteins (as opposed to chemical reactions among metabolites), which is usually referred to as a protein interaction network. Interaction networks have been studied by a number of authors []. Another important class of biological network is the genetic regulatory network. The expression of a gene, i.e., the production by transcription and translation of the protein for which the gene codes, can be controlled by the presence of other proteins, both activators and inhibitors, so that the genome itself forms a switching network with vertices representing the proteins and directed edges representing dependence of protein production on the proteins at other vertices. Genetic regulatory networks were in fact one of the first networked dynamical systems for which large-scale modeling attempts were made. Another much studied example of a biological network is the food web, in which the vertices represent species in an ecosystem and a directed edge from species A to species B indicates that A preys on B. Construction of complete food webs is a laborious business, but a number of quite extensive data sets have become available in recent years. Neural networks are another class of biological networks of considerable importance. Measuring the topology of real neural networks is extremely difficult, but has been done successfully in a few cases. The best known example is the reconstruction of the 282-neuron neural network of the nematode C. Elegans by White et al.. In this paper I will focus on the undirected graphs that can be extracted from 3D structures of enzymes, which is a particular set of proteins involved in catalytic activity of biochemical reactions.

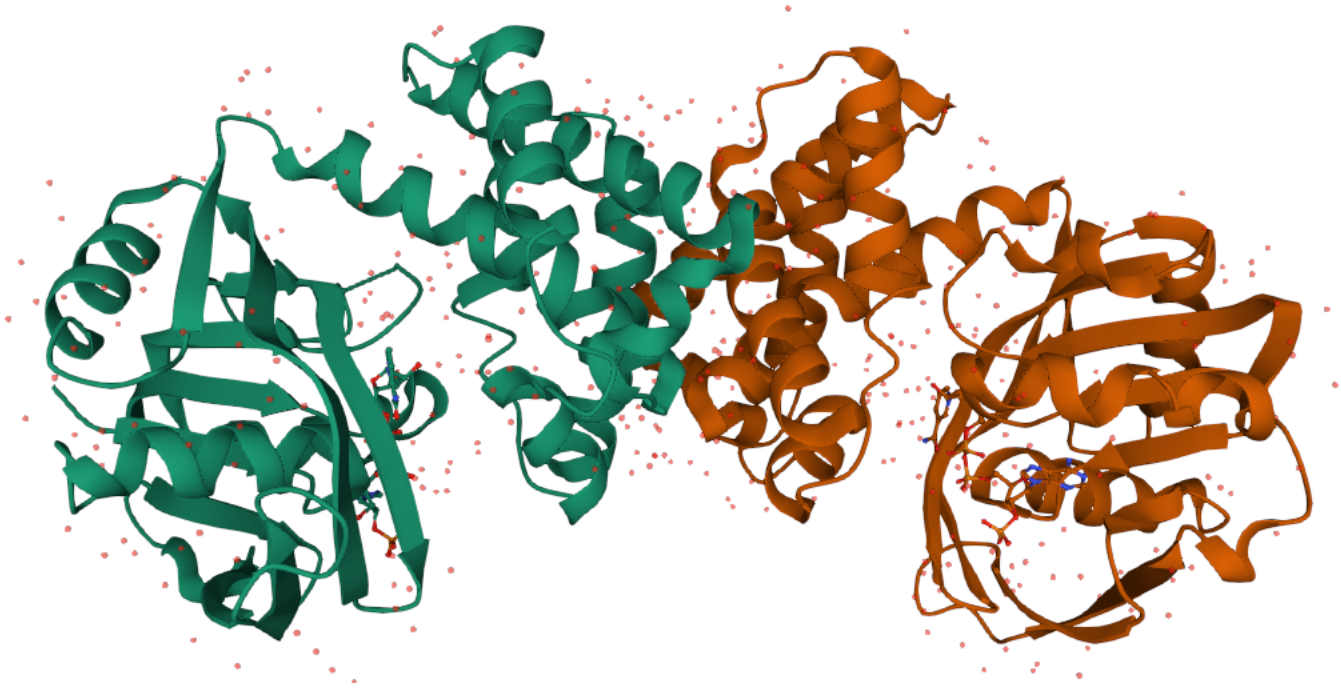


Figure 1.3: Here is represented the 3D structure coming from 3YCA. the enzyme is present in the database I have considered and is taken just as an example.

1.2 Enzymes

An enzyme is a protein that has a specific function within the cell (and also outside), formed by one or more chains of amino acids linked with peptide bonds in a special location where usually are found carbon-alpha $C\alpha$. It must be folded in the correct way to perform its function. Proteins in general are characterized by four levels of organization: primary, secondary, tertiary and quaternary structures as one can see from figure 1.3. Structures are differentiated in many ways. The first to be listed is the difference in aminoacids that form them. Many differences arise in this respect, for example they are differentiated by: polarity, charge, presence of hydroxyl group, presence of sulphur, on the basis of aromatic ring structure, acidic, basic. Also on the basis of essential amino acids and non-essential amino acids. In our work we study enzymes, that are very efficient catalysts for biochemical reactions. They speed up reactions by providing an alternative reaction pathway of lower activation energy as it is shown in figure 1.4. Like all catalysts, enzymes take part in the reaction - that is how they provide an alternative reaction pathway. But they do not undergo permanent changes and so remain unchanged at the end of the reaction. They can only alter the rate of reaction, not the position of the equilibrium. Most chemical catalysts catalyse a wide range of reactions. They are not usually very selective. In contrast enzymes are usually highly selective, catalysing specific reactions only. This specificity is due to the shapes of the enzyme molecules. Many enzymes consist of a protein and a non-protein (called the cofactor). The proteins in enzymes are usually globular. The intra- and intermolecular bonds that hold proteins in their secondary and tertiary structures are disrupted by changes in temperature and pH. This affects shapes and so the catalytic activity of an enzyme is pH and temperature sensitive. Cofactors may be:

- organic groups that are permanently bound to the enzyme (prosthetic groups)

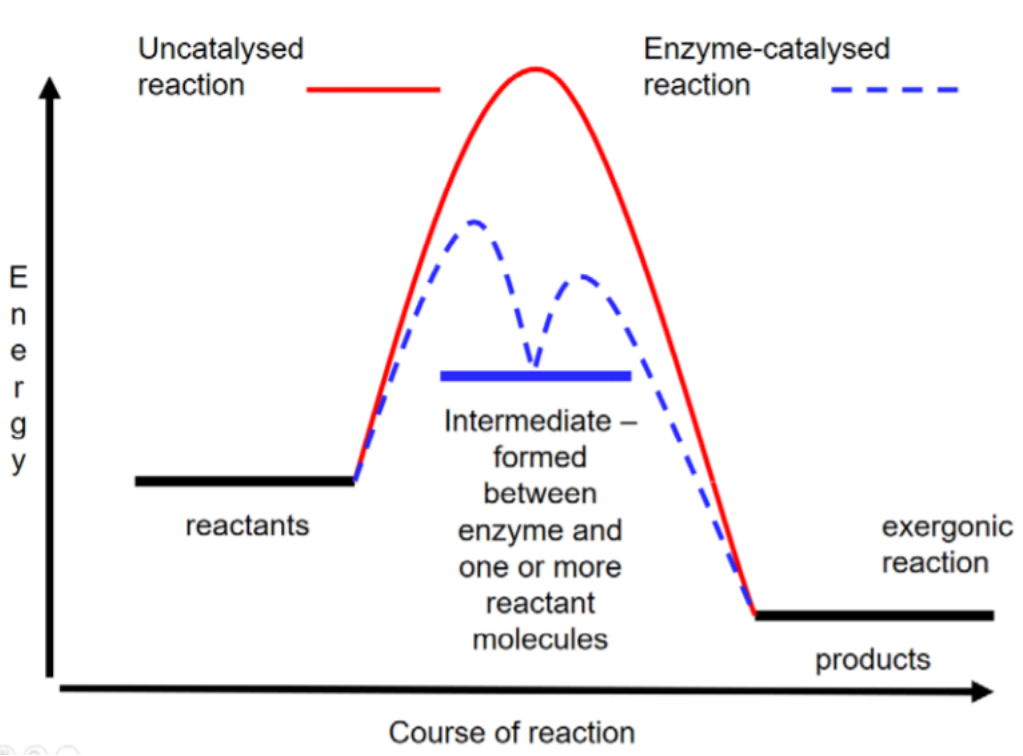


Figure 1.4: Alternative reaction pathway of lower activation energy. On the x-axis is the time and on the y-axis the energy of the system of the two reactants. In blue is the pathway that is allowed by the presence of the enzyme as a catalyst

- cations - positively charged metal ions (activators), which temporarily bind to the active site of the enzyme, giving an intense positive charge to the enzyme's protein
- organic molecules, usually vitamins or made from vitamins (coenzymes), which are not permanently bound to the enzyme molecule, but combine with the enzyme-substrate complex temporarily

1.2.1 How enzymes work

For two molecules to react they must collide with one another. They must collide in the right direction (orientation) and with sufficient energy. Sufficient energy means that between them they have enough energy to overcome the energy barrier to reaction. This is called the activation energy. Enzymes have an active site. This is part of the molecule that has just the right shape and functional groups to bind to one of the reacting molecules. The reacting molecule that binds to the enzyme is called the substrate. An enzyme-catalysed reaction takes a different 'route'. The enzyme and substrate form a reaction intermediate. Its formation has a lower activation energy than the reaction between reactants without a catalyst. A simplified picture:

$$RouteA : S_1 + S_2 \longrightarrow P \tag{1.1}$$

$$RouteB : S_1 + E + S_2 \longrightarrow intermediate + S_2 \longrightarrow P + E \tag{1.2}$$

So the enzyme is used to form a reaction intermediate, but when this reacts with another reactant the enzyme reforms. In this model the enzyme molecule changes shape as the

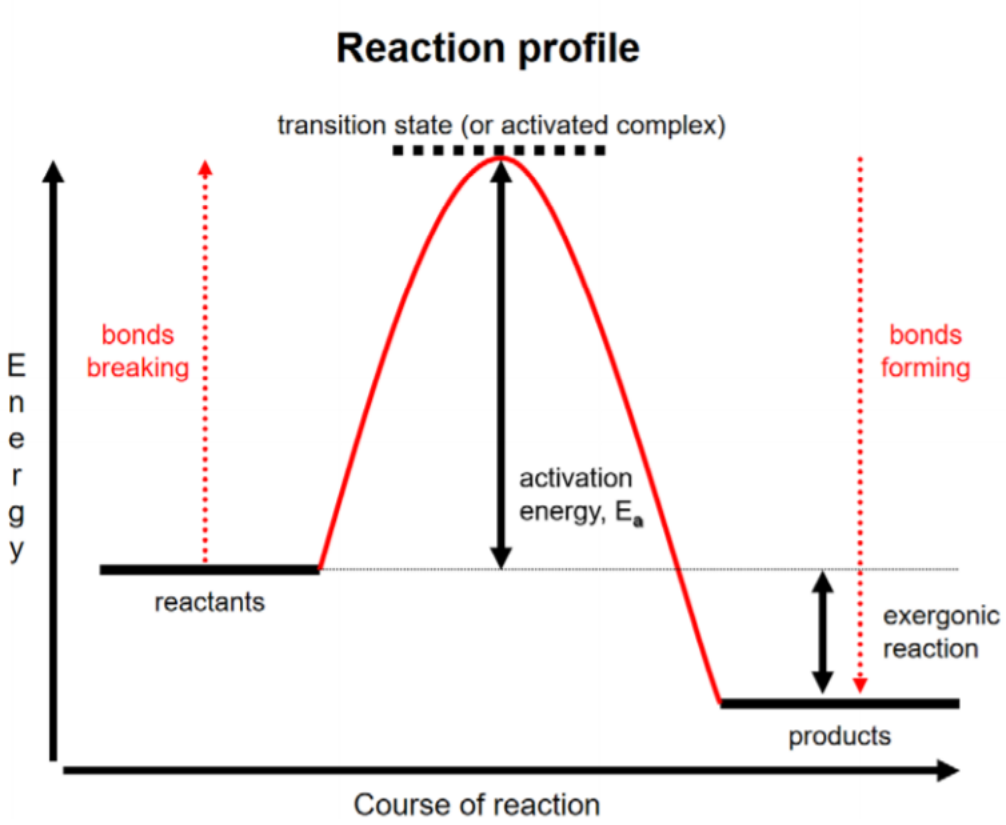


Figure 1.5: In this figure is represented the reaction profile of any two reactants in general. In particular here it is shown what is tha activation energy. On the x-axis is the time and on the y-axis the energy of the system of the two reactants

(a) (b)

Figure 1.6: In these two figures are represented the dependence of the catalitic rate in terms of temperature and pH of the environment

substrate molecules gets close. The change in shape is 'induced' by the approaching substrate molecule. This more sophisticated model relies on the fact that molecules are flexible because single covalent bonds are free to rotate. As the temperature rises, reacting molecules have more and more kinetic energy. This increases the chances of a successful collision and so the rate increases. There is a certain temperature at which an enzyme's catalytic activity is at its greatest (see figure 1.6a). This optimal temperature is usually around human body temperature (37.5 °C) for the enzymes in human cells. Above this temperature the enzyme structure begins to break down (denature) since at higher temperatures intra- and intermolecular bonds are broken as the enzyme molecules gain even more kinetic energy. Each enzyme works within quite a small pH range. There is a pH at which its activity is greatest (the optimal pH). This is because changes in pH can make and break intra- and intermolecular bonds, changing the shape of the enzyme and, therefore, its effectiveness.

One of the feature we consider in this work is the classification that this difference brings in the primary structure. This classification is encoded in uniprot codes. Uniprot codes have a one to one correspondence to primary structures.

1.3 Categorization of Enzymes

In this section I will introduce the macro categories I had at my disposal.

1.3.1 Enzyme Commission EC

The first characteristic that I want to speak about and that labels enzymes in is EC from [1]. Enzymes are proteins that allow some catalytic process to occur. There are of many types and during years it has been developed a standard way of referring at them. This standard is called enzyme commission (EC): the enzyme commission recommended a standard way to code enzymes. This will correspond to a code, made of 4 numbers: the first identify the MACRO CLASS of belonging, and then there is the organization in other sub-classes, in a hierarchical way. Until having a complete classification identifiable with 4 numbers. Here below I report the top-level of the code:

1. EC 1 (Oxidoreductases): To catalyze oxidation/reduction reactions; transfer of H and O atoms or electrons from one substance to another

2. EC 2 (Transferases): Transfer of a functional group from one substance to another. The group may be methyl-, acyl-, amino- or phosphate group

3. EC 3 (Hydrolases): Formation of two products from a substrate by hydrolysis

4. EC 4 (Lyases): Non-hydrolytic addition or removal of groups from substrates. C-C, C-N, C-O or C-S bonds may be cleaved

5. EC 5 (Isomerases): Intramolecule rearrangement, i.e. isomerization changes within a single molecule

6. EC 6 (Ligases): Join together two molecules by synthesis of new C-O, C-S, C-N or C-C bonds with simultaneous breakdown of ATP

7. EC 7 (Translocases): Catalyse the movement of ions or molecules across membranes or their separation within membranes

All these characteristics are embedded in the references to UniProt.

1.3.2 taxonomy

All enzymes we are speaking about are inserted in cellular environments of different organisms. The organisms we have considered come from different branches of taxonomy trees. We are speaking about: Mammalia, green plants, fungi, bacteria and archeas. These last ones belong to a family that has evolved at more elevated temperature averagely (and this is seen from figure 2.11b).

1.3.3 Michaelis-Menten constant

In biochemistry, Michaelis–Menten kinetics is one of the best-known models of enzyme kinetics. It is named after German biochemist Leonor Michaelis and Canadian physician Maud Menten. The model takes the form of an equation describing the rate of enzymatic reactions, by relating reaction rate of formation of product $[\boldsymbol{P}]$ to of formation of product $[\boldsymbol{S}]$.

$$\nu = \frac{d}{dt}[\boldsymbol{P}] = V_{\max}\frac{[\boldsymbol{S}]}{K_{\mathrm{M}} + [\boldsymbol{S}]} \tag{1.3}$$

This equation is called the Michaelis–Menten equation. Here, $\mathrm{V_{max}}$ represents the maximum rate achieved by the system, happening at saturating substrate concentration. The value of the Michaelis constant $\mathrm{K_M}$ (KM) is numerically equal to the substrate concentration at which the reaction rate is half of $\mathrm{V_{max}}$. Biochemical reactions involving a single substrate are often assumed to follow Michaelis–Menten kinetics, without regard to the model's underlying assumptions. In 1901, French physical chemist Victor Henri found that enzyme reactions were initiated by a bond (more generally, a binding interaction) between the enzyme and the substrate. His work was taken up by German biochemist Leonor Michaelis and Canadian physician Maud Menten, who investigated the kinetics of an enzymatic reaction mechanism, invertase, that catalyzes the hydrolysis of sucrose into glucose and fructose. In 1913, they proposed a mathematical model of the reaction.

It involves an enzyme, E, binding to a substrate, S, to form a complex, ES, which in turn releases a product, P, regenerating the original enzyme. This may be represented schematically as:

$$\mathrm{E} + \mathrm{S} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{ES} \xrightarrow{k_2} \mathrm{E} + \mathrm{P}$$

where k_1(forward rate constant), k_{-1}(reverse rate constant), and k_2 (catalytic rate constant) denote the rate constants, the double arrows between $\boldsymbol{S}$ (substrate) and $\boldsymbol{ES}$ (enzyme-substrate complex) represent the fact that enzyme-substrate binding is a reversible process, and the single forward arrow represents the formation of $\boldsymbol{P}$ (product). In general increasing the relative concentration of the enzyme with respect to the other components of the reaction increases the rate of reaction as can be seen in figure 1.7 Under certain assumptions – such as the enzyme concentration being much less than the substrate concentration – the rate of product formation is given by (1.3). The reaction order depends on the relative size of the two terms in the denominator. At low substrate concentration $[\boldsymbol{S}] \ll K_{\mathrm{M}}$ so that the reaction rate varies linearly with substrate concentration $[\boldsymbol{S}]$ as in the first part of fig 1.8. However at higher $[\boldsymbol{S}]$ with $[\boldsymbol{S}] \gg K_{\mathrm{M}}$, the reaction becomes independent of $[\boldsymbol{S}]$ (zero-order kinetics)and asymptotically approaches its maximum rate . The constant is not affected by the concentration or purity of an enzyme. The value of K_{M} is dependent on both the identity of enzyme and that of the substrate, as well as conditions such as temperature and pH.These are the characteristics we hope to spot with our analysis. The model is used in a variety of biochemical situations other than enzyme-substrate interaction.

1.3.4 ligand enzyme interaction

The ligand can be a macromolecule (DNA/RNA/proteins), elemental (< 600 Da) ions, small organic molecules or peptides. Binding this object is essential for maintaining the activity of the protein. There can be different approaches for the study of PPI, the reductionist (molecular view point) point of view (since we know that experimentally we have knowledge about complexes of proteins that are interacting) studies the features of interaction from an atomic point of view. This approach requires a high level of knowledge. Since the number of interactions for which we have knowledge is low it is necessary to look for an "higher" point of view based on understanding what protein interacts with, and so having an idea about the protein network. (all the nodes that represent the proteins, and edges that indicate that 2 proteins are in interaction, so

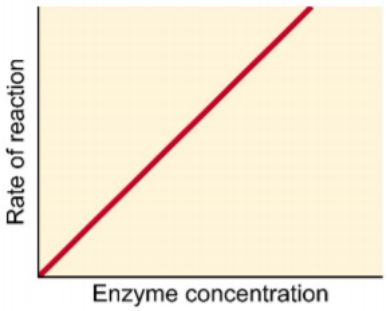


Figure 1.7: Here is represented the rate of reaction's curve of some enzyme with respect to the concentration of enzymes

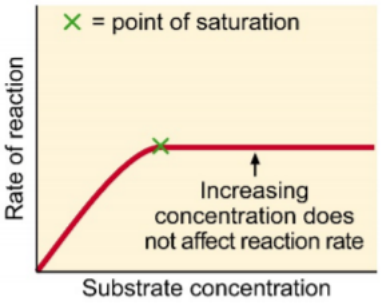


Figure 1.8: Here is represented the rate of reaction's curve of some enzyme with respect to the concentration of substrate. In particular it is interesting to see that if in a first time increasing the concentration of the substrate increases the rate of reaction. After having reached the saturation point (interpreted as the concentration at which all the allosteric places in an enzyme are occupied) the rate of reaction doesn't change anymore and remains steady

edges start from a single protein). The most complete knowledge of interaction among proteins is obtained by mixing the two levels,the result is to understand which is the function, how the protein interacts and what are the groups of proteins that cooperate of a certain function and identify pathways.

Chapter 2

Enzyme dataset description

2.1 PDB database

In this section I will speak about PDB repository focusing on just those aspects exploited in this work. For this job I am referencing what RSCB PDB guide tells about itself. I will therefore, highlight just what I needed in my work, infact, despite the great variety of data available, I have decided to consider just carbon-alpha ($C\alpha$) coordinates contained in some ENTRY for those enzymes with just one ASSEMBLY. This means that, to each Uniprot (sequence of aminoacids, contained in another repository that is however linked with PDB) I have different ENTRIES (PDB codes), and for each ENTRY I will have different INSTANCES (coordinates) of the same ENTITY ($C\alpha$) in one ASSEMBLY. This is important, as I decided not to consider those enzymes composed by multiple ASSEMBLIES, (hemogoblin has got multiple assemblies of the same protein) NOTE: when I speak about atoms ($C\alpha$), I refer to the ENTITIES that represent the position of different aminoacids in the enzyme. In practice for one uniprot I have many PDB ID's. From each PDB ID I extrct the coordinates of $C\alpha$. The PDB ID's contain just 'non-repeated' structures.

2.1.1 hierarchical structure

As we already said in the precedent sections, proteins are composed of linear chains of amino acids that (often) fold into compact subunits which then can associate into higher level assemblies with other proteins, small molecule ligands, and water or other solvent molecules. Biomolecules in the Protein Data Bank (PDB) archive are organized and represented using this hierarchy to simplify searching and exploration. Four levels of hierarchy are commonly used: Entry, Entity, Instance, and Assembly. These levels are

- An ENTRY is all data pertaining to a particular structure deposited in the PDB and is designated with a 4-character alphanumeric identifier called the PDB identifier or PDB ID (e.g., 2hbs).
- An ENTITY is a chemically unique molecule**chemically unique molecule** that may be polymeric, such as a protein chain or a DNA strand, or non-polymeric, such as a soluble ligand. Some entries may even have branched polymeric entities, such as oligosaccharides. The entities I have considered are aminoacidic chains ($C\alpha$ coordinates).

- An INSTANCE is a particular occurrence of an ENTITY. An ENTRY may contain multiple INSTANCES of an ENTITY, for example, many copies of $C\alpha$ associated to different aminoacids.
- An ASSEMBLY is a biologically relevant group of one or more INSTANCES of one or more ENTITIES that are associated with each other to form a stable complex and/or perform a function.

These four level of characterization are important not to confuse an entity for another; in this way different Cα, have different coordinates. In my work I am interested, as already said, in Cα coordinates. The PDB archive in this way uniquely maps Various identifiers to specifically indicate one atom or groups of atoms.

2.1.2 Entity level identifiers

A Protein or polypeptide (short fragment of protein) whose sequence has been mapped to UniProt includes a UniProt Accession Code (e.g., P11838) for that entity. This observation is very important in my work as (as I will explain later) I have worked with just PDBs coming from Uniprot codes. So the access I have made is just to those entities that have also a Uniprot code associated.

2.1.3 PDB Data

We have seen that the PDB archive is a repository that is structured in a hierarchical way. At the end of the hierarchy lies the entity. In the entity are contained list of atomic coordinates and other information describing proteins and other important biological macromolecules. Structural biologists use methods such as X-ray crystallography, NMR spectroscopy, and cryo-electron microscopy to determine the location of each atom relative to each other in the molecule. They then deposit this information, which is then annotated and publicly released into the archive by the wwPDB. Structures are available for many of the proteins and nucleic acids involved in the central processes of life, there you can find multiple structures for a given molecule, or partial structures, or structures that have been modified or inactivated from their native form.

The primary information stored in the PDB archive consists of coordinate files for biological molecules. These files list the atoms in each protein (we are interested just in alpha carbon atoms that are those representing the position of aminoacids), and their 3D location in space.

2.1.4 X-ray Cristallography

Most of the structures included in the PDB archive were determined using X-ray crystallography. For this method, the protein is purified and crystallized, then subjected to an intense beam of X-rays. The proteins in the crystal diffract the X-ray beam into one or another characteristic pattern of spots, which are then analyzed (with some tricky methods to determine the phase of the X-ray wave in each spot) to determine the distribution of electrons in the protein. The resulting map of the electron density is then interpreted to determine the location of each atom. X-ray crystallography can provide very detailed atomic information, showing every atom in a protein or nucleic acid along with atomic details of ligands, inhibitors, ions, and other molecules that are incorporated

into the crystal. Biological molecule crystals are finicky: some form perfect, well-ordered crystals and others form only poor crystals. The accuracy of the atomic structure that is determined depends on the quality of these crystals. In perfect crystals, we have far more confidence that the atomic structure correctly reflects the structure of the protein. Two important measures of the accuracy of a crystallographic structure are its resolution, which measures the amount of detail that may be seen in the experimental data, and the R-value, which measures how well the atomic model is supported by the experimental data found in the structure factor file. I have taken most of the data from these experiment and I didn't check the precision of the structures and so they are considered at the same level.

2.1.5 NMR spectroscopy

NMR spectroscopy may be used to determine the structure of proteins. The protein is purified, placed in a strong magnetic field, and then probed with radio waves. A distinctive set of observed resonances may be analyzed to give a list of atomic nuclei that are close to one another, and to characterize the local conformation of atoms that are bonded together. This list of restraints is then used to build a model of the protein that shows the location of each atom. The technique is currently limited to small or medium proteins, since large proteins present problems with overlapping peaks in the NMR spectra.

A major advantage of NMR spectroscopy is that it provides information on proteins in solution, as opposed to those locked in a crystal or bound to a microscope grid, and thus, NMR spectroscopy is the premier method for studying the atomic structures of flexible proteins. A typical NMR structure will include an ensemble of protein structures, all of which are consistent with the observed list of experimental restraints. The structures in this ensemble will be very similar to each other in regions with strong restraints, and very different in less constrained portions of the chain. Presumably, these areas with fewer restraints are the flexible parts of the molecule, and thus do not give a strong signal in the experiment.

In the PDB archive, you will typically find two types of coordinate entries for NMR structures. The first includes the full ensemble from the structural determination, with each structure designated as a separate model. The second type of entry is a minimized average structure. These files attempt to capture the average properties of the molecule based on the different observations in the ensemble. You can also find a list of restraints that were determined by the NMR experiment. These include things like hydrogen bonds and disulfide linkages, distances between hydrogen atoms that are close to one another, and restraints on the local conformation and stereochemistry of the chain. Just few enzymes' structures are of this kind and I treated them alike the others.

2.2 Description of dataset

In this section I will describe the repository I used jumping on much of the preprocessing did for the final pipeline. I highlight the principal results. In **starting repository** I explain how I arrived to the conclusion that I have available 494 Uniprot codes and 4061 PDB files, showing the distribution of number of nodes and connected components . In **little remark** I explain how I associated the labels (KM,temperature). In **distributions of data with respect to taxonomy** I show the tab of distribution of taxonomy

Uniprot_0	PDB_{01}	ECfirst_{01}	taxonomy_{01}	temperature_{01}	KM_{01}	$C\alpha_{01}$
...	...	...	...	...	...	...
	$\text{PDB}_{0\text{p}0}$	$\text{ECfirst}_{0\text{p}0}$	$\text{taxonomy}_{0\text{p}0}$	$\text{temperature}_{0\text{p}0}$	$\text{KM}_{0\text{p}0}$	$C\alpha_{0\text{p}0}$
...	...	...	...	...	...	...
Uniprot_{499}	$\text{PDB}_{499,1}$	$\text{ECfirst}_{499,1}$	$\text{taxonomy}_{499,1}$	$\text{temperature}_{499,1}$	$\text{KM}_{499,1}$	$C\alpha_{499,1}$
...	...	...	...	...	...	...
...	$\text{PDB}_{499\text{p}499}$	$\text{ECfirst}_{499\text{p}499}$	$\text{taxonomy}_{499\text{p}499}$	$\text{temperature}_{499\text{p}499}$	$\text{KM}_{499\text{p}499}$	$C\alpha_{499\text{p}499}$

Table 2.1: In this table is represented the structure of the data I have. Note that for each uniprot code (aminoacidic sequence) there are many PDB (structures) each one equipped with EC, taxonomy, temperature, KM and $C\alpha$

commenting those aspects that in my opinion are more relevant in the subsection below. In **Statistical comparison of different enzyme groups** I describe the principles of ANOVA and the method of judging the results in tabs (group starting from 2.12a) Then I represent the results in graphs and tabs labeled.

2.2.1 starting repository

I have started with a repository created by Giulia Menichetti. This repository contained informations about:

- uniprot codes associated to (aminoacidic sequences)
- PDB codes associated to (3D structures of enzymes)
- EC and EC first level
- organism
- KM (Michaelis Menten constant)
- substrate (ligand for catalitic process)
- units, comment
- Temperature and pH (of the experiment)
- NCBIid (code coming from National Center for Biotechnology Information) SuperClass
- taxonomy : isMammalia, isFungi, isGreenPlant, isVirus, isBacteria, isArchea, EukNotFungi

In the end, after some preprocessing, I obtained 494 Uniprot codes and 4061 PDB in a relation 1 to many. For each PDB I have associated in a relation 1-1 , EC first, taxonomy, temperature and KM. The latter two have been chosen following the criteria in **little remark**. The fact that to each Uniprot code are associated many PDB is due to the fact that for a single aminoacidic sequence have been done many measures of the structures and studied different regions of them via NMR and X-Ray cristallography . In the end I will have a repository represented as in tab 2.1.

In the next sections I will present some graphical representations of the dataset that highlights the distributions with respect to taxonomy and EC, dimension of the

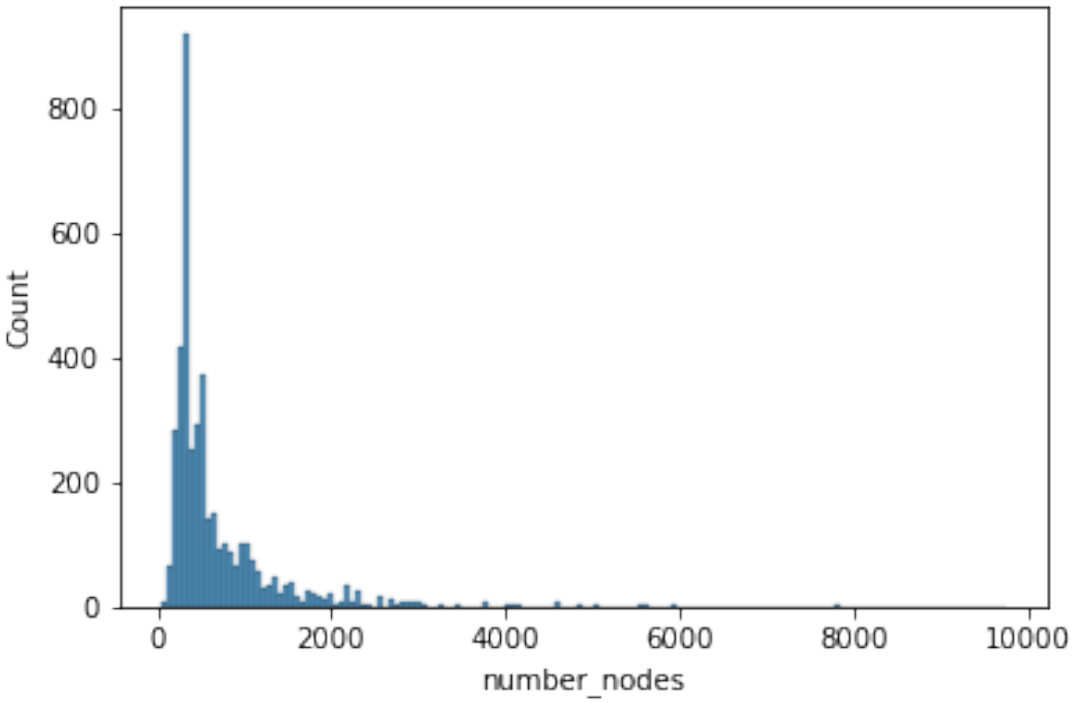


Figure 2.1: Histogram of the number of nodes per protein. The highest number of nodes is around 300, and the average is at 739 nodes. The biggest enzyme considered has 9278 nodes

enzyme, temperature of the experiment and KM. These representations give an idea of the statistics we are considering and can be useful in the interpretation of the results we obtain.

2.2.2 little remark

In the phase of preprocessing I found out that, for each uniprot, (that is already associated to different PDB) I have multiple temperature, pH, KM and ligands. For this reason I have considered just the average temperature and just one of the KM I had (chosen at random, not having any further information to the selection) and considered no ligand at all. This choice is motivated by the fact that I have seen that temperatures for one uniprot ID are reasonably similar as the different experiment are made more or less in the same environment for one sequence (as I have that sequence appear just once for each organism and taxonomy). On the other hand I have that as for the same enzyme I have multiple ligands then different ligands will have associated different Michaelis-Menden constants that infact represent the velocity of reaction at equilibrium of the ligands in their specific allosteric places. As we have different ligands I expect that these values can vary notabely as we change the chemical nature of the ligands.

2.2.3 distributions of data with respect to taxonomy

In this section I describe enzymes' labels I have. In particular I represent for each taxonomy value, how the EC first, temperature, KM and number of $C\alpha$ are distributed

(See from fig.(2.2b) to fig.(2.11a)).

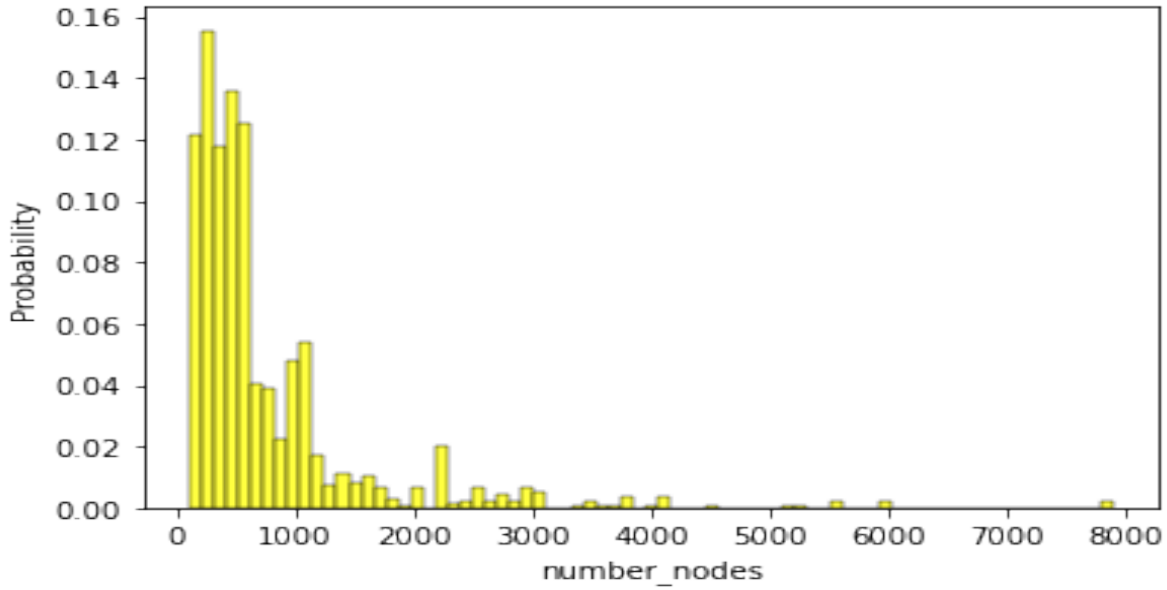
taxonomy	min $C\alpha$	max $C\alpha$
Mammalia	99	7893
Green Plants	160	4144
Fungi	114	3407
Bacteria	53	9728
Archea	205	1942

description and observations about distribution of labels

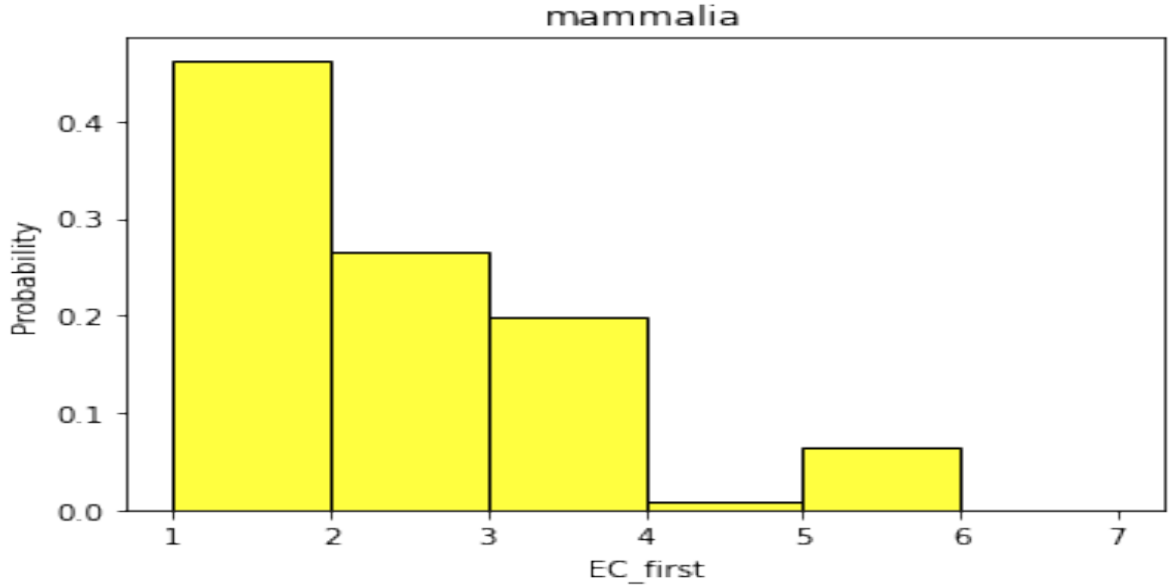
In this section I make inferences about the distribution of labels highlighting what I think it is important. The most populated sets are those belonging to mammalia and bacteria,while we have that other taxonomies are less represented. In figure **??** can be seen that the most of mammalia's enzymes have relatively small $C\alpha$. The average is 789 $C\alpha$. . We can see a similar behavior for green plants,fungi and bacteria respctively in figure 2.4a,2.6a,2.8a. The average $C\alpha$ for green plants are 679. More the 50 percent of the overall distributions has got less then 500 $C\alpha$. To be noted that fungi are particularly peaked at around 330. **Infact in our dataset there are 554 of them having exactly 330 $C\alpha$ coming from the same uniprot ID but different 3d structures.** This means that the aminoacidic structure is the same while the PDB structure are different. Looking at $C\alpha$ in figure 2.10a for archea we see a tail also at high number infact the average is 918 nodes even though the sample is not very varied. Informations about the number of enzymes for each taxonomy class are summed up in 2.2.3 . For all the taxonomy families we see that the EC distribution as in figures 2.2b ,2.4b,2.6b,2.8b,2.10b is peaked at 1 with averagely over 50 per cent belonging to that class. As shown in figures 2.3a,2.5a,2.7a,2.9a,2.11a we have that the Michaelis-Menten constants are peaked around 0 with no much dispersion but for Archea that has around 30 per cent of the enzymes whose constant is bigger then 10. I have not worried about the biological meaning of these data, I will care just about classification problem. A different picture comes from the distribution of temperatures whose distributions vary more from family to family. In mammalia as we can see from figure 2.3b the temperatures are distributed in the range 20-48 degree celsius and mode around 35 degrees. For green plants as in fig. 2.5b the lowest temperature the range is 20-40 degrees with an mode around 30. In fungi as we can see from fig. 2.7b temperature is distributed from 15 to 40 with mode around 35 but less peaked with respect to others. Bacteria live in a range from 0-80 degrees and the mode is around 35 as we see from fig. 2.9b. In the end Archea are in a range from 30 to degrees with the mode around 60 degrees as we see in fig.2.11b. This shows that Archea have evolved in hotter environments.

2.2.4 Statistical comparison of different enzyme groups

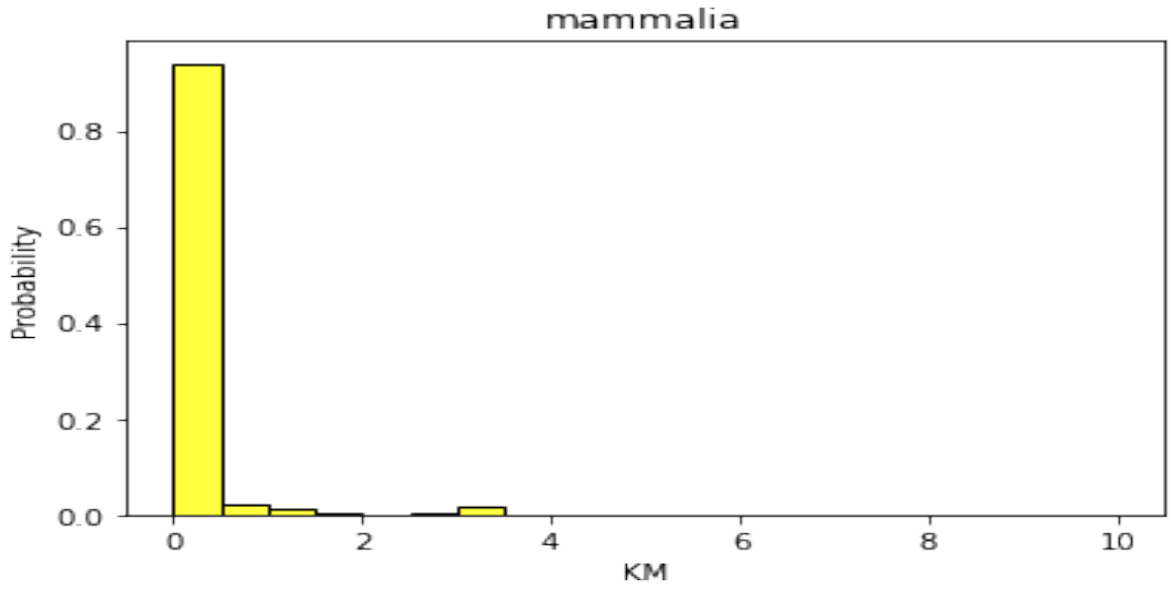
In this subsection I reproduce the results about an Anova (Analysis of Variance) to the labels of the numerical labels with respect to taxonomy and EC first. From this analysis will report just Tukey HSD of multiple comparison of means, that show via p-value considerations whether the null hypothesis (different sub-distributions belong to the same overall distribution) should be rejected (reject=True, p-adj ≤ 0.05, means that the two sub-distributions are different) or not (reject=False, p-adj ≥ 0.05, means that



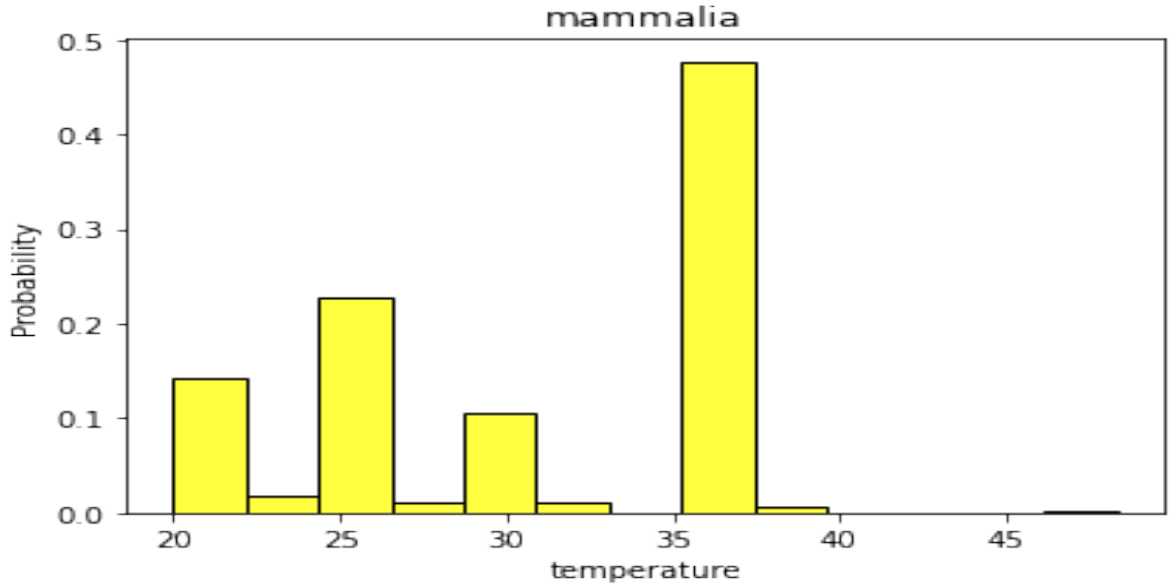
(a) Histplot picking data just from the mammalia family and whose x-axis are the number of nodes of the network associated to the enzyme, on the y-axis the probability of observing it



(b) Histplot picking data just from the mammalia family and whose x-axis is the first number of the EC code of the enzyme, on the y-axis the probability of observing it

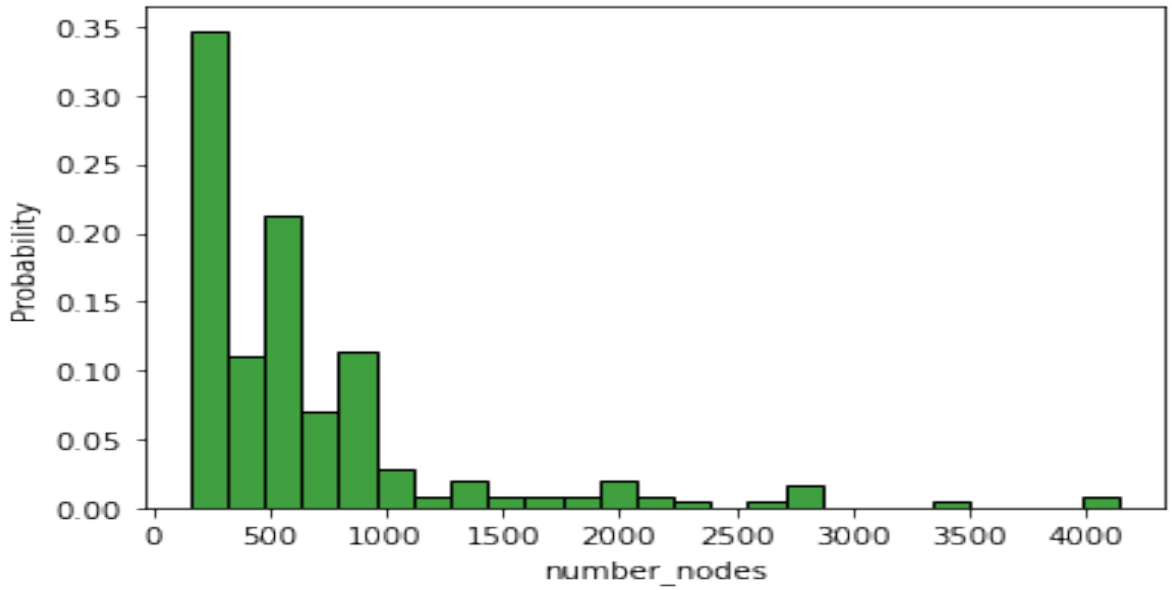


(a) Histplot picking data just from the mammalia family and whose x-axis is the Michaelis-Menten constant associated to one of the catalitic processes of the enzyme, on the y-axis the probability of observing it

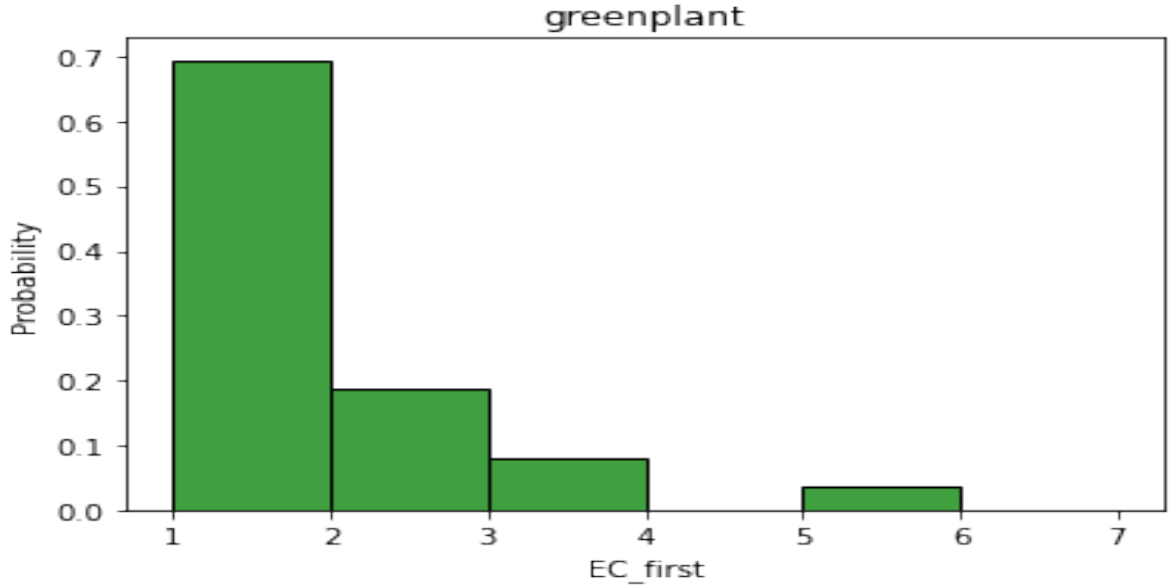


(b) Histplot picking data just from the mammalia family and whose x-axis is the temperature of the ambient of the enzyme, on the y-axis the probability of observing it

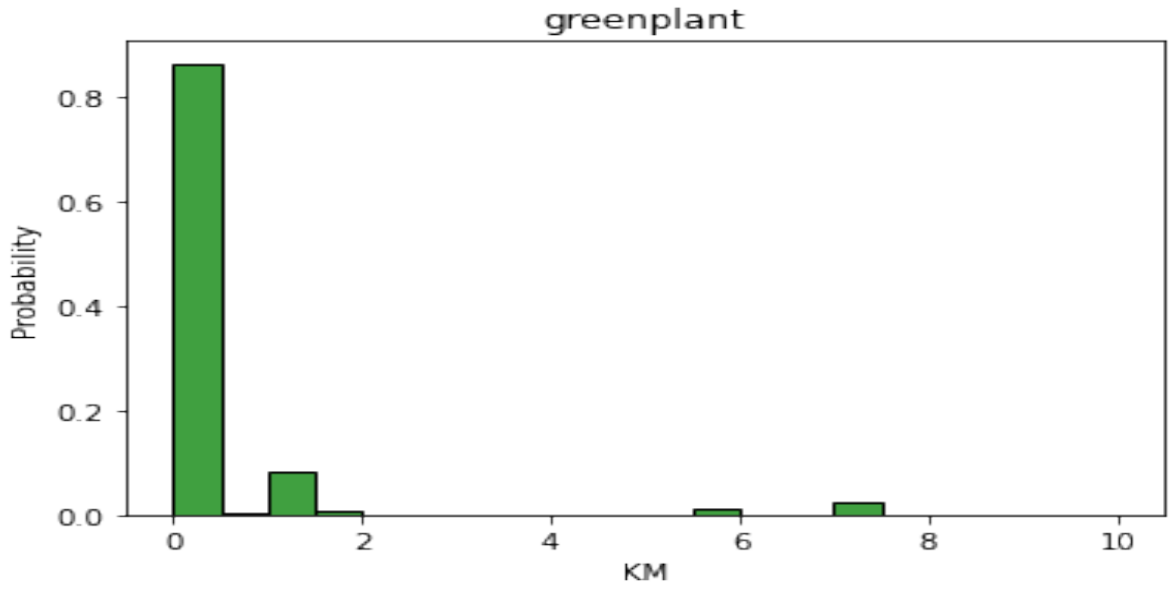
Figure 2.3: Histogram the probability distribution of enzymes that belong to the Mammalia family in its number of nodes, EC first KM and temperature. To this category belong 1639 enzymes distributed in 7 different organisms



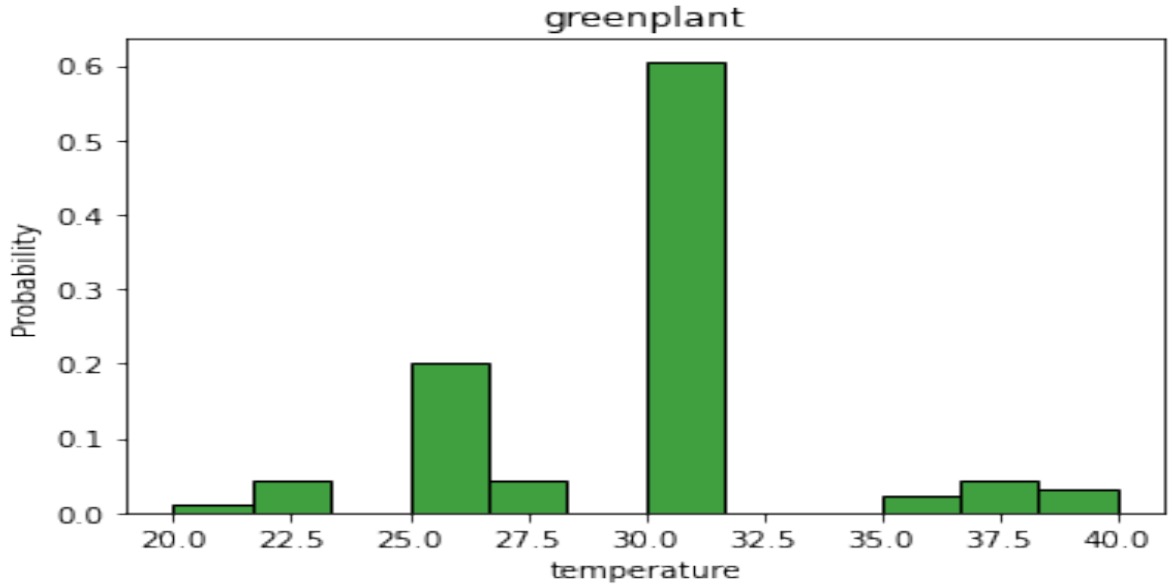
(a) Histplot picking data just from the green plants' family and whose x-axis are the number of nodes of the network associated to the enzyme, on the y-axis the probability of observing it



(b) Histplot picking data just from the green plants' family and whose x-axis is the first number of the EC code of the enzyme, on the y-axis the probability of observing it

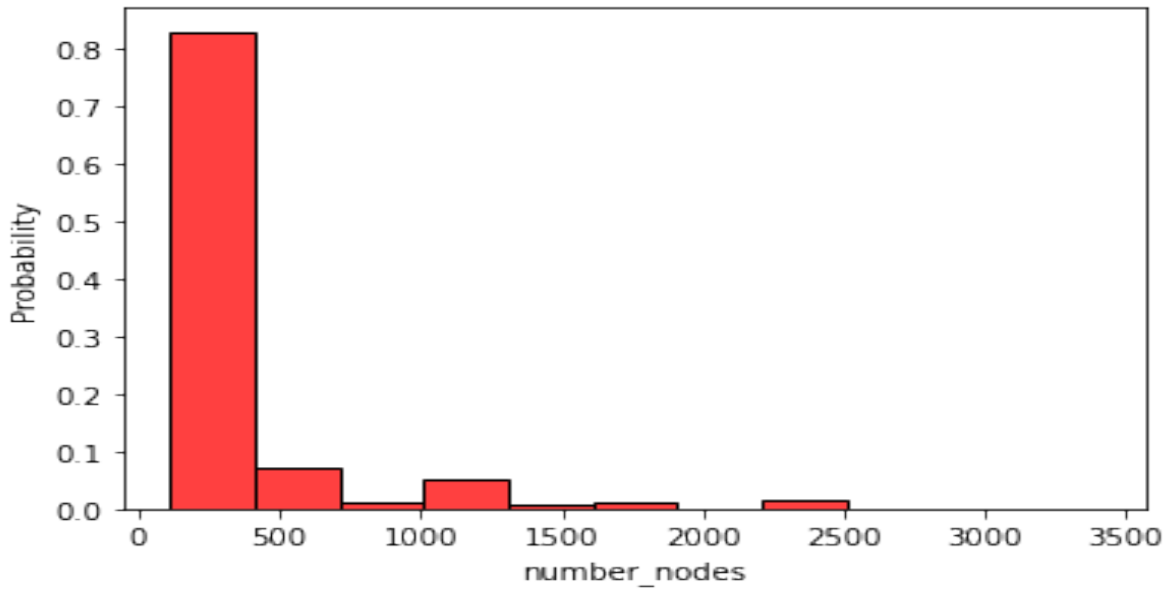


(a) Histplot picking data just from the green plants' family and whose x-axis is the Michaelis-Menten constant associated to one of the catalitic processes of the enzyme, on the y-axis the probability of observing it

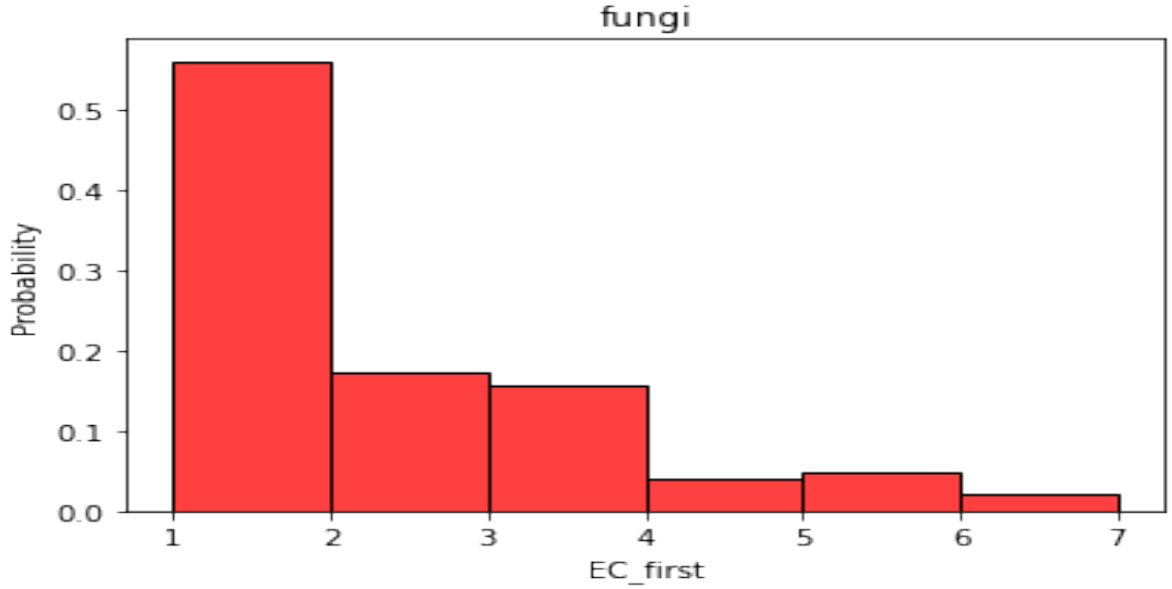


(b) Histplot picking data just from the green plants' family and whose x-axis is the temperature of the ambient of the enzyme, on the y-axis the probability of observing it

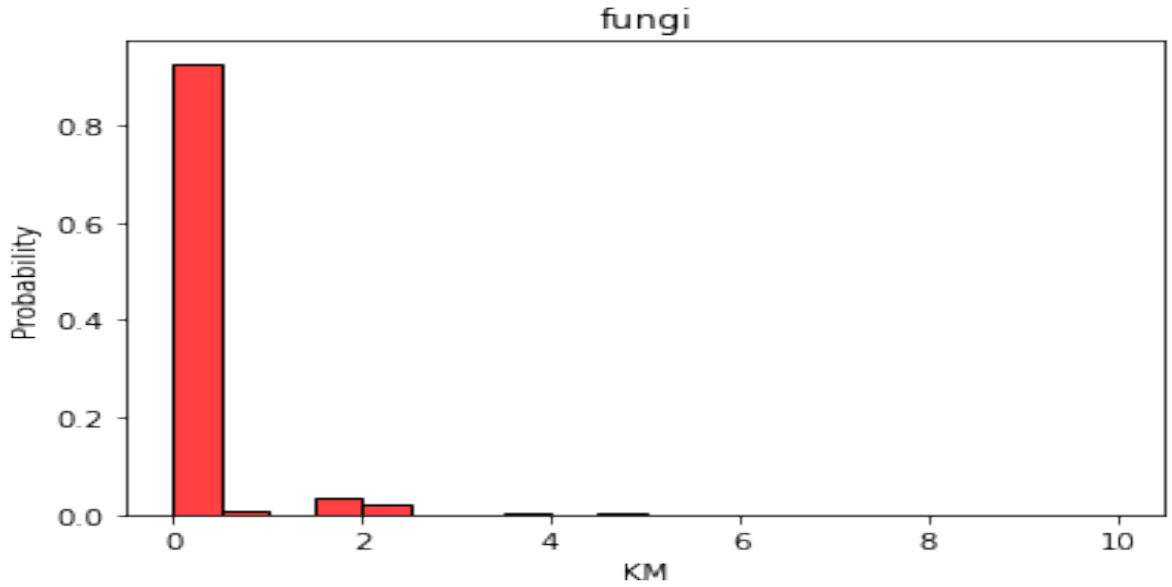
Figure 2.5: It is represented in an histogram the probability distribution of enzymes that belong to the green plants' family in its number of nodes,EC first KM and temperature. To this category belong 259 enzymes distributed in 34 different organisms



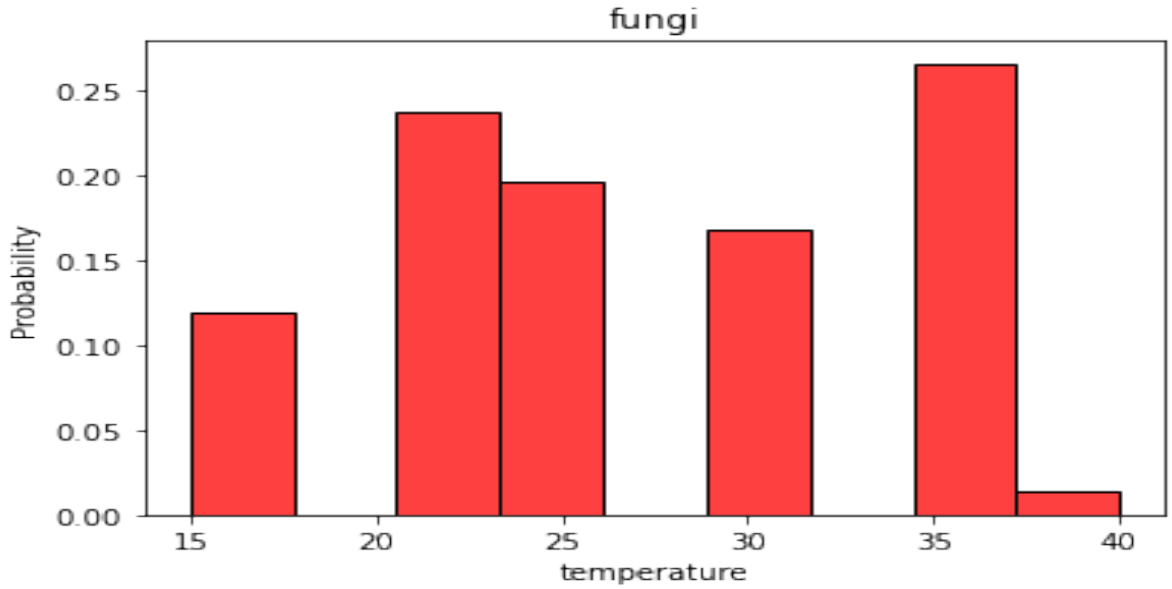
(a) Histplot picking data just from the fungi family and whose x-axis are the number of nodes of the network associated to the enzyme, on the y-axis the probability of observing it



(b) Histplot picking data just from the fungi family and whose x-axis is the first number of the EC code of the enzyme, on the y-axis the probability of observing it

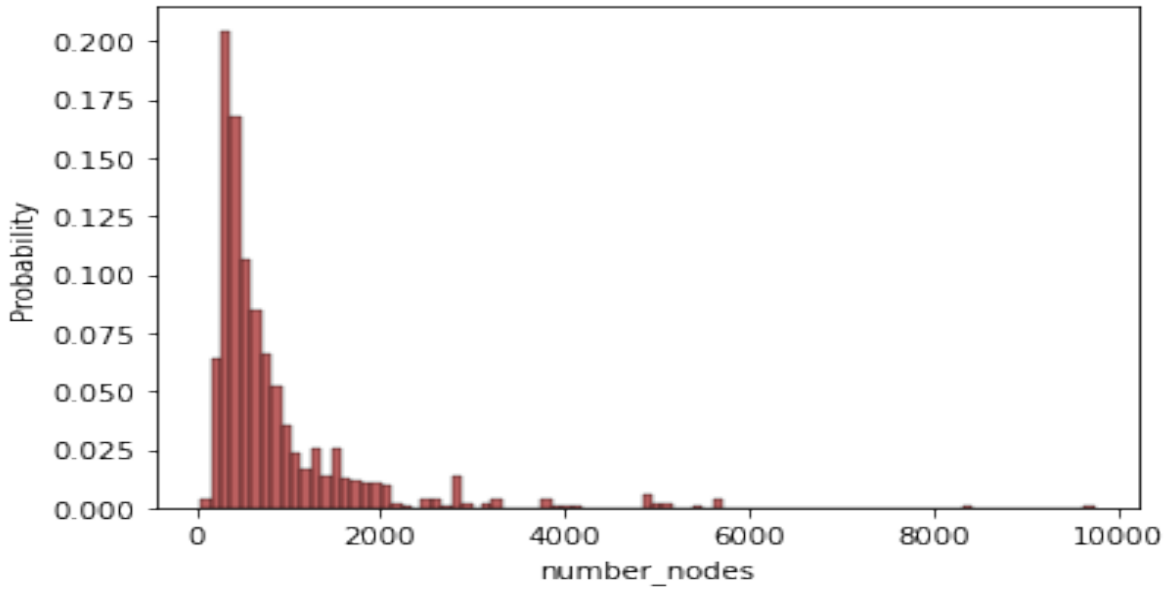


(a) Histplot picking data just from the fungi family and whose x-axis is the Michaelis-Menten constant associated to one of the catalitic processes of the enzyme, on the y-axis the probability of observing it

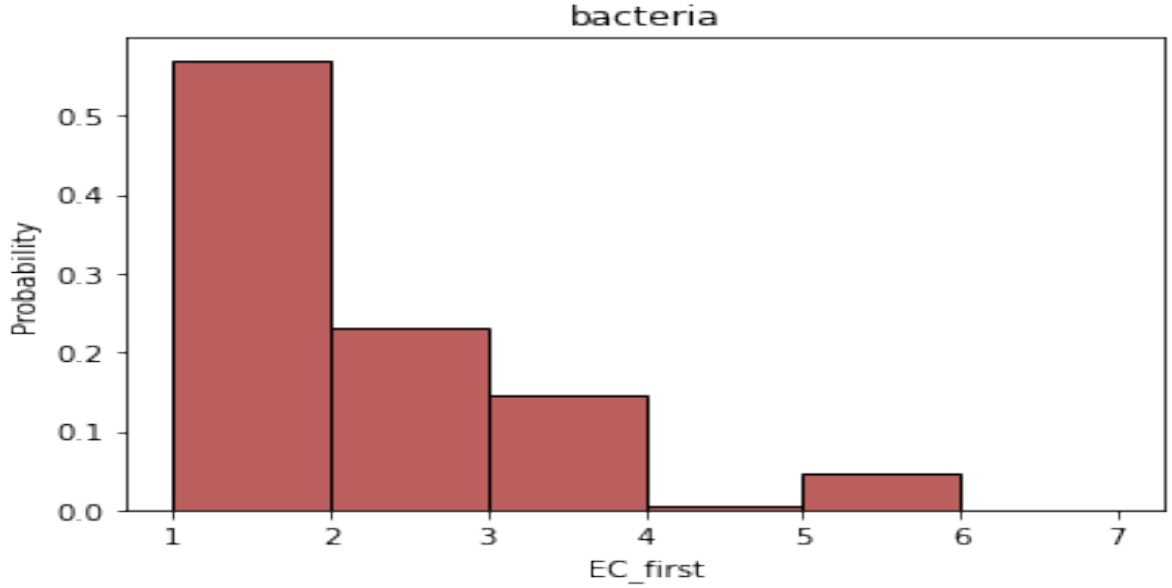


(b) Histplot picking data just from the fungi family and whose x-axis is the temperature of the ambient of the enzyme, on the y-axis the probability of observing it

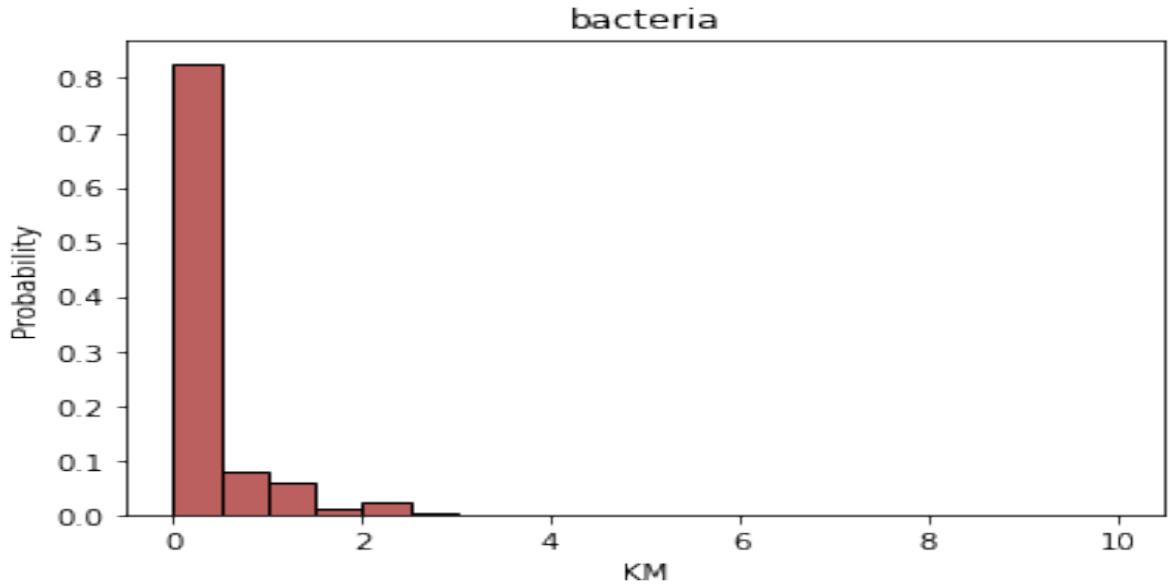
Figure 2.7: It is represented in an histogram the probability distribution of enzymes that belong to the fungi family in its number of nodes,EC first KM and temperature. To this category belong 846 enzymes distributed in 27 different organisms



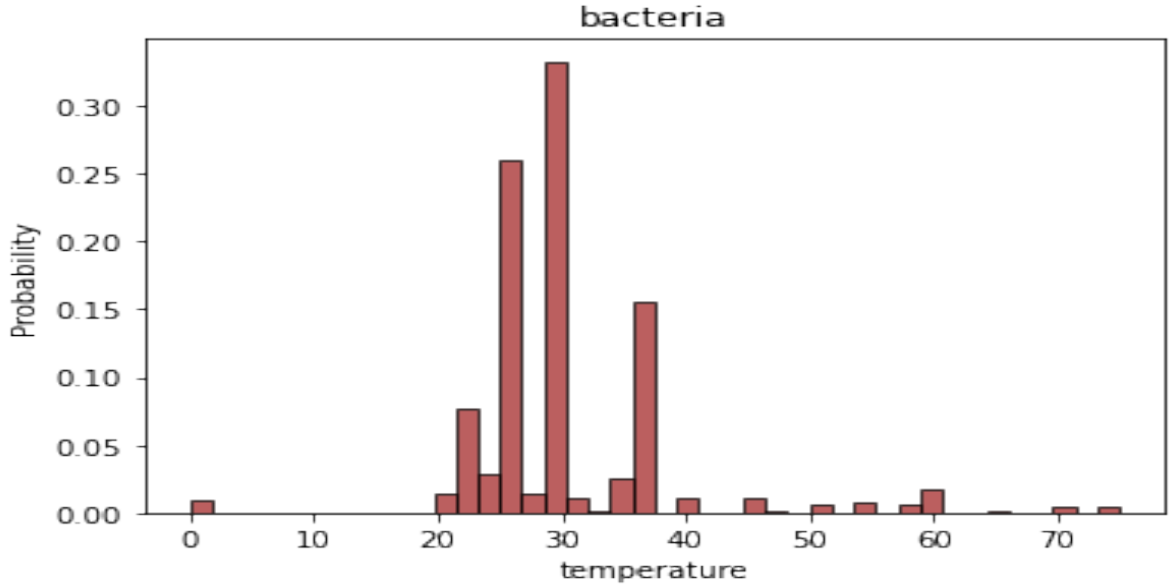
(a) Histplot picking data just from the bacteria family and whose x-axis are the number of nodes of the network associated to the enzyme, on the y-axis the probability of observing it



(b) Histplot picking data just from the bacteria family and whose x-axis is the first number of the EC code of the enzyme, on the y-axis the probability of observing it

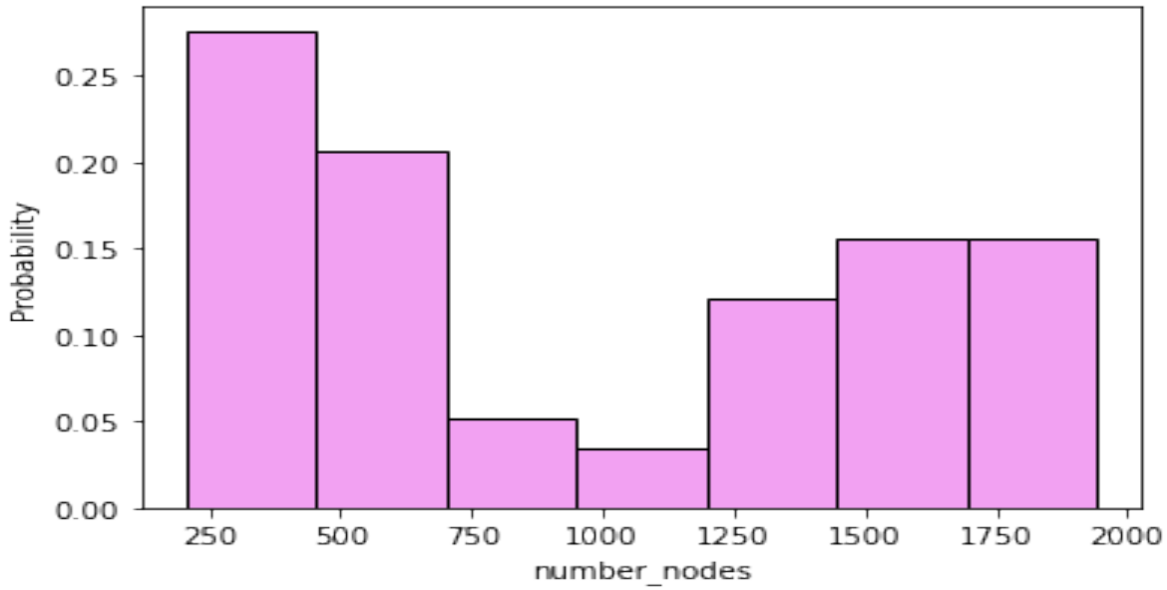


(a) Histplot picking data just from the bacteria family and whose x-axis is the Michaelis-Menten constant associated to one of the catalitic processes of the enzyme, on the y-axis the probability of observing it

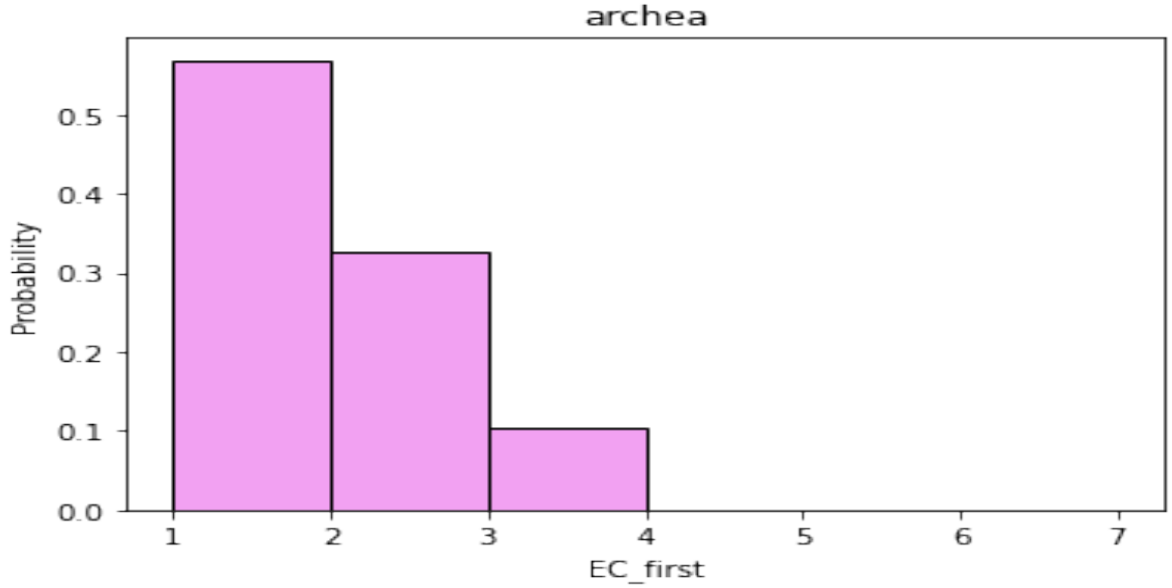


(b) Histplot picking data just from the bacteria family and whose x-axis is the temperature of the ambient of the enzyme, on the y-axis the probability of observing it

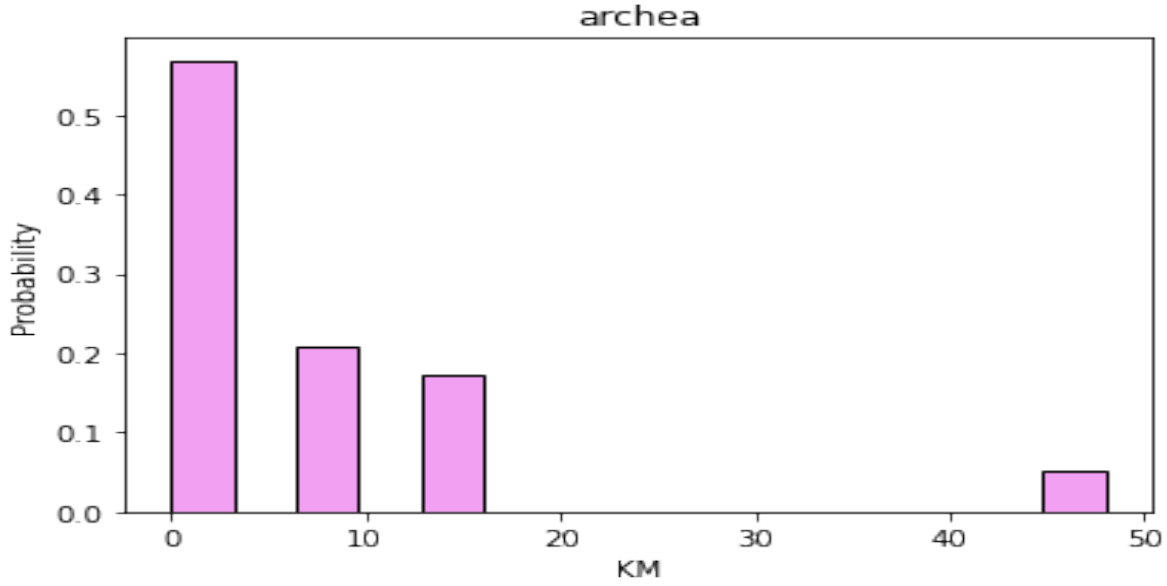
Figure 2.9: It is represented in an histogram the probability distribution of enzymes that belong to the bacteria family in its number of nodes,EC first KM and temperature. To this category belong 1057 enzymes distributed in 51 different organisms



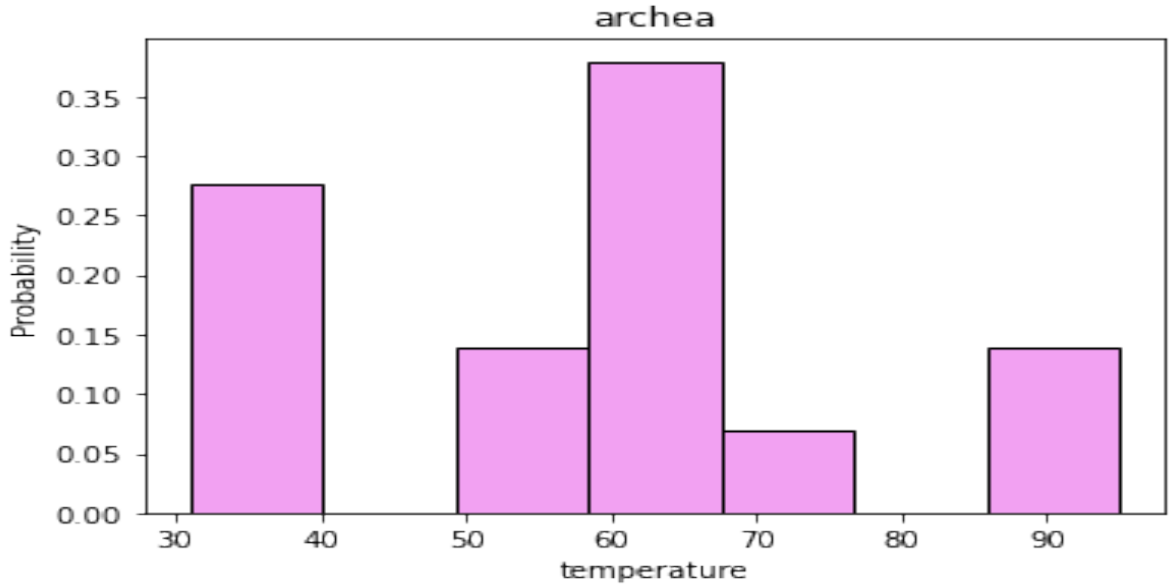
(a) Histplot picking data just from the archea family and whose x-axis are the number of nodes of the network associated to the enzyme, on the y-axis the probability of observing it



(b) Histplot picking data just from the archea family and whose x-axis is the first number of the EC code of the enzyme, on the y-axis the probability of observing it



(a) Histplot picking data just from the archea family and whose x-axis is the Michaelis-Menten constant associated to one of the catalitic processes of the enzyme, on the y-axis the probability of observing it



(b) Histplot picking data just from the archea family and whose x-axis is the temperature of the ambient of the enzyme, on the y-axis the probability of observing it

Figure 2.11: It is represented in an histogram the probability distribution of enzymes that belong to the archea family in its number of nodes,EC first KM and temperature. To this category belong 58 enzymes distributed in 9 different organisms

the two sub-distributions are the same). These results are reported in figures: 2.13b, 2.13a, 2.12b,2.12a. I comment the result obtained.

Tukey's HSD analysis

The calculations of ANOVA can be characterized as computing a number of means and variances, dividing two variances and comparing the ratio to a handbook value to determine statistical significance. Calculating a treatment effect is then trivial: "the effect of any treatment is estimated by taking the difference between the mean of the observations which receive the treatment and the general mean". In particular: Null hypothesis: Groups means are equal (no variation in means of groups) H_0: $\mu_1=\mu_2=\ldots=\mu_p$ and Residuals (experimental error) are normally distributed (Shapiro-Wilks Test), Homogeneity of variances (variances are equal between treatment groups) (Levene's or Bartlett's Test), Observations are sampled independently from each other.

$$\begin{aligned} y_{ij} &= \mu + \alpha_i + \epsilon_{ij} \\ SS_T &= SS_B + SS_E \end{aligned} \tag{2.1}$$

where:

- y_{ij} is the j^{th} observation of the i^{th} i=1,...,p (in my case will be taxonomy and EC first values separately)
- μ is the overall population mean (unknown)
- α_i is the deviation from the mean
- ϵ_{ij} Error distributed as above
- i levels of groups i=1,...,p
- j observation (or replicates) for each group (k=1,...,r)
- $SS_B = \sum_i p_i(\bar{y_{i.}} - \bar{y_{..}})^2$, weighted variances among of means of different groups
- $SS_E = \sum_i (y_{ij} - \bar{y_{i.}})^2$,variance within a group
- $SS_T = \sum_{ij} p_i(y_{ij} - \bar{y_{..}})^2$ weighted variance if I consider unique distribution.

From ANOVA analysis, we know that treatment differences are statistically significant, but ANOVA does not tell which treatments are significantly different from each other. To know the pairs of significant different treatments, we will perform multiple pairwise comparison (post hoc comparison) analysis for all unplanned comparison using Tukey-Krames's honestly significantly differenced (HSD) test. HSD test is not an equivalence test as it is not (perfectly) transitive, however it allows to capture similarities among distributions. This consists in calculating from Anova analysis:

$$HSD = q_{A,\alpha,dof}\sqrt{\frac{MS_E}{2}(\frac{1}{n_i} + \frac{1}{n_j})} \tag{2.2}$$

Where $q_{A,\alpha,dof}$ studentized range statistic with: A number of the group (i.e. Mammalia,EC first=1),α significance level (0.05) and dof degrees of freedom. MS_E means squared error from Anova, n_i and n_j the number of elements in the confronting groups.

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
EukNotFun	isArchea	30.4929	0.001	27.2963	33.6895	True
EukNotFun	isBacteria	2.712	0.001	1.0467	4.3772	True
EukNotFun	isFungi	-0.4136	0.9	-2.7418	1.9147	False
EukNotFun	isGreenPlant	1.5193	0.3407	-0.6435	3.6821	False
EukNotFun	isMammalia	3.3293	0.001	1.729	4.9295	True
isArchea	isBacteria	-27.7809	0.001	-30.7299	-24.8319	True
isArchea	isFungi	-30.9065	0.001	-34.2746	-27.5383	True
isArchea	isGreenPlant	-28.9736	0.001	-32.2295	-25.7176	True
isArchea	isMammalia	-27.1636	0.001	-30.0764	-24.2508	True
isBacteria	isFungi	-3.1255	0.001	-5.1002	-1.1509	True
isBacteria	isGreenPlant	-1.1927	0.3949	-2.9692	0.5839	False
isBacteria	isMammalia	0.6173	0.5118	-0.4022	1.6368	False
isFungi	isGreenPlant	1.9329	0.1991	-0.4762	4.342	False
isFungi	isMammalia	3.7428	0.001	1.8227	5.663	True
isGreenPlant	isMammalia	1.8099	0.0317	0.0942	3.5257	True

(a) In this figure is represented the Tukey's analysis of different taxonomy groups with respect to the temperatures. I list below the classes that can be considered having the same distribution of temperatures:

- EuknotFungi with Fungi and Bacteria
- Bacteria with Green Plants and Mammalia
- Fungi with Green Plants

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
EukNotFun	isArchea	5.2982	0.1226	-0.7288	11.3252	False
EukNotFun	isBacteria	1.5197	0.7117	-1.6201	4.6594	False
EukNotFun	isFungi	0.0283	0.9	-4.3615	4.4181	False
EukNotFun	isGreenPlant	9.6501	0.001	5.5723	13.7279	True
EukNotFun	isMammalia	-0.6085	0.9	-3.6257	2.4087	False
isArchea	isBacteria	-3.7786	0.3803	-9.3388	1.7817	False
isArchea	isFungi	-5.2699	0.1686	-11.6204	1.0805	False
isArchea	isGreenPlant	4.3519	0.33	-1.7871	10.4908	False
isArchea	isMammalia	-5.9067	0.0265	-11.3987	-0.4148	True
isBacteria	isFungi	-1.4914	0.8491	-5.2145	2.2318	False
isBacteria	isGreenPlant	8.1304	0.001	4.7808	11.48	True
isBacteria	isMammalia	-2.1282	0.02	-4.0503	-0.206	True
isFungi	isGreenPlant	9.6218	0.001	5.0795	14.1641	True
isFungi	isMammalia	-0.6368	0.9	-4.2572	2.9836	False
isGreenPlant	isMammalia	-10.2586	0.001	-13.4936	-7.0236	True

(b) In this figure is represented the Tukey's analysis of different taxonomy groups with respect to the KM. I list below the classes that can be considered having the same distribution of KM:

- EukNotFungi with Archea,Bacteria,Fungi and Mammalia
- Archea with Bacteria Fungi and Green Plants
- Bacteria with Fungi
- Fungi with mammalia

. In practice it seems that Green Plants seem to have separate distributions of KM with respect to the others.

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
1.0	2.0	2.4851	0.001	0.8915	4.0788	True
1.0	3.0	0.8576	0.2998	-0.32	2.0351	False
1.0	4.0	-1.3471	0.3513	-3.282	0.5878	False
1.0	5.0	1.8354	0.2389	-0.5464	4.2171	False
1.0	6.0	4.9732	0.0026	1.1817	8.7647	True
2.0	3.0	-1.6276	0.0391	-3.2072	-0.048	True
2.0	4.0	-3.8322	0.001	-6.035	-1.6295	True
2.0	5.0	-0.6498	0.9	-3.2539	1.9543	False
2.0	6.0	2.4881	0.4652	-1.4469	6.4231	False
3.0	4.0	-2.2046	0.0139	-4.128	-0.2813	True
3.0	5.0	0.9778	0.83	-1.3946	3.3502	False
3.0	6.0	4.1156	0.0239	0.33	7.9013	True
4.0	5.0	3.1824	0.0168	0.3566	6.0083	True
4.0	6.0	6.3203	0.001	2.2352	10.4054	True
5.0	6.0	3.1378	0.3014	-1.1769	7.4526	False

(a) In this figure is represented the Tukey's analysis of different EC groups with respect to the temperatures.I list below the classes that can be considered having the same distribution of temperature:

- '1' with '3','4' and '5'
- '2' with '5' and '6'
- '3' with '5'
- '5' with '6'

. In practice, '5' and '6' seem to have a distribution of temperatures that bond all the others

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
1.0	2.0	0.4862	0.9	-2.0939	3.0664	False
1.0	3.0	-0.2334	0.9	-2.14	1.6731	False
1.0	4.0	13.4588	0.001	10.3262	16.5914	True
1.0	5.0	3.4322	0.1137	-0.424	7.2884	False
1.0	6.0	-0.0331	0.9	-6.1717	6.1055	False
2.0	3.0	-0.7196	0.9	-3.2771	1.8378	False
2.0	4.0	12.9726	0.001	9.4062	16.5389	True
2.0	5.0	2.9459	0.347	-1.2702	7.1621	False
2.0	6.0	-0.5193	0.9	-6.8902	5.8516	False
3.0	4.0	13.6922	0.001	10.5783	16.8062	True
3.0	5.0	3.6656	0.0716	-0.1755	7.5066	False
3.0	6.0	0.2003	0.9	-5.9288	6.3294	False
4.0	5.0	-10.0266	0.001	-14.6018	-5.4515	True
4.0	6.0	-13.4919	0.001	-20.1059	-6.8779	True
5.0	6.0	-3.4653	0.6919	-10.4511	3.5205	False

(b) In this figure is represented the Tukey's analysis of different EC groups with respect to the KM. It can be seen here that the only 'EC first' that seemengly comes to a different distribution of KM with respect all the others that in constrast seem to have the same distribution, is '4'.

Figure 2.13

From figure 2.12a one can see that EukNotFungi are found in similar distribution with Fungi and Green Plants. Bactreria are similar to Mammalia and GreenPlants but not with Fungi. Fungi is similar with GreenPlants. From here it can be noted that Archeas live in much hotter environment. From figure 2.12b one can see that more or less they have all the same distributions of KM but Green Plants. From figure 2.13a one can see that '5' and '6' seem to have a distribution of temperatures that bond all the others

From figure 2.13b one can see that '4' is different in KM distribution with everybody.

2.3 Uniprot P11838

Uniprot P11838 is the endothiapepsin protein from fungi that belongs to the class of enzymes of Aspartic proteases. These are a class of enzymes that play a causative role in numerous diseases such as malaria , Alzheimer's disease, fungal infections , and hypertension . For this reason on the PDB repository I have found many strucutues.

Chapter 3

Mathematical methods

As already explained in the previous chapters, I have 4061 structures of enzymes corresponding to 494 uniprot aminoacid sequences. From these structures, I first extract the contact map, then calculate the Laplacian and the Normalized Laplacian and their eigenvectors and eigenvalues. I take the 40 biggest and 40 smallest eigenvalues as descriptors for each structure. Since it is known that some relation exists among smallest eigenvalues and the number of nodes, I tried to rescale them (to make dimensionality dependence vanish or reduce). I then proceed by applying PCA,t-SNE and UMAP to visualize 'distances' among enzymes. The principal motivation for this procedure is that, it is believed that global features of an enzyme are essential for determination of its properties (folding kinetic for example (KM) [23]). In this respect, the possibility of reconstructing the protein structure starting from a reduced representation (A) is an essential aspect for its application to the study of the protein structure. This fact is linked to [4] that suggests a representation of our enzyme as a manifold represented by same contact map (A). The contact map has got naturally associated a Laplacian (L) operator. The action of L on a N-dimensional lattice corresponds to the discretization of a N-dimensional elastic membrane, where L's eigenvalues represent the frequencies of the normal modes and L's eigenvectors represent the normal mode solutions or eigenfunctions.

3.1 Methods used for the analysis

In this chapter I expose the mathematical background needed to justify the approach I pursued in this work for the analysis of the dataset described. In **Introduction to the use of graph in the description of complex networks**, I introduce a little bit the history of networks and the usual features that one extrapolate for a description of them. In particular it is interesting to look at the clustering components, average distances among nodes, link density and degree distributions. These data are extracted to understand features of the networks we are working with. In **Contact maps** I introduce how I construct the networks I will work with. I have chosen two different thresholds (8 $\mathring{A}$,12 $\mathring{A}$), as commonly used in the literature. In **Laplacian** and subsections, I will define Laplacian and speak about some of its properties that I looked at. In particular, I state that the smallest eigenvalue of the laplacian λ_0=0, and that the number of connected components of a graph is equal to the number of eigenvalues equal to 0 (important result as I consider just connected contact maps). In **Introduction to distance geometry** and **Embedding problem** I introduce the embedding procedure. This problem will be approached in the following sections. In **PCA**, **t-SNE** and **UMAP** I will explain the

different embeddings techniques. The final goal is a classification through geometrical patterns associated to each enzyme.

3.2 Introduction to the use of graph in the description of complex networks

This section is taken from [18]. The historical development of the use of graphs in the study of complex systems is due to Erdos and Renyi, that in their work hypothesized a simple model that for each graph of N nodes and a probability of connection between two nodes p, had in average pN(N-1) edges. Their models were completely random, but in the years people started to realize that in real networks there is some order and correlation among parts of the graph, and so some effort should have been put in the realization of measures that could grasp these non random properties.

Motivated by this some concepts have arisen. Small worlds: The small-world concept in simple terms describes the fact that despite their often large size, in most networks there is a relatively short path between any two nodes. The distance between two nodes is defined as the number of edges along the shortest path connecting them. The most popular manifestation of small worlds is the "six degrees of separation" concept, uncovered by the social psychologist Stanley Milgram (1967), who concluded that there was a path of acquaintances with a typical length of about six between most pairs of people in the United States (Kochen, 1989). The small-world property appears to characterize most complex networks: the actors in Hollywood are on average within three co-stars from each other, or the chemicals in a cell are typically separated by three reactions. The small-world concept, while intriguing, is not an indication of a particular organizing principle. Indeed, as Erdos and Renyi have demonstrated, the typical distance between any two nodes in a random graph scales as the logarithm of the number of nodes. Thus random graphs are small worlds as well. Clustering: A common property of social networks is that cliques form, representing circles of friends or acquaintances in which every member knows every other member. This inherent tendency to cluster is quantified by the clustering coefficient (Watts and Strogatz, 1998), a concept that has its roots in sociology, appearing under the name "fraction of transitive triples" (Wassermann and Faust, 1994). Let us focus first on a selected node i in the network, having k_i edges which connect it to k_i other nodes. If the nearest neighbors of the original node were part of a clique, there would be $\frac{k_i(k_i-1)}{2}$ edges between them. The ratio between the number E_i of edges that actually exist between these k_i nodes and the total number $\frac{k_i(k_i-1)}{2}$ gives the value of the clustering coefficient of node i,

$$C_i = \frac{2E_i}{k_i(k_i - 1)} \tag{3.1}$$

The clustering coefficient of the whole network is the average of all individual Ci's.

$$\bar{C} = \frac{1}{N} \sum C_i \tag{3.2}$$

In a random graph, since the edges are distributed randomly, the clustering coefficient is C=p. However, in most, if not all, real networks the clustering coefficient is typically much larger than it is in a comparable random network. Degree distribution: Not all

nodes in a network have the same number of edges (same node degree). The spread in the node degrees is characterized by a distribution function P(k), which gives the probability that a randomly selected node has exactly k edges. Since in a random graph the edges are placed randomly, the majority of nodes have approximately the same degree, close to the average degree $\langle k \rangle$ of the network. The degree distribution of a random graph is a Poisson distribution with a peak at P($\langle k \rangle$). One of the most interesting developments in our understanding of complex networks was the discovery that for most large networks the degree distribution significantly deviates from a Poisson distribution.

3.3 Network properties related to distance matrix

As [16] and [18] tells us, for studying networks it can be important look for two fundamental properties of real complex networks that have attracted much attention recently: the small-world and the scale-free properties. Many naturally occurring networks are small world since one can reach a given node from another one, following the path with the smallest number of links between the nodes, in a very small number of steps. This corresponds to the so-called "six degrees of separation" in social networks[17]. It is mathematically expressed by the slow (logarithmic) increase of the average diameter of the network,$\bar{l}$, with the total number of nodes N, $\bar{l} \sim ln(N)$, where ℓ is the shortest distance between two nodes and defines the distance metric in complex networks Equivalently, we obtain:

$$N \sim e^{\bar{l}/l_0} \tag{3.3}$$

where l_0 is a characteristic length.

A second fundamental property in the study of complex networks arises with the discovery that the probability distribution of the number of links per node, P(k) (also known as the degree distribution), can be represented by a power-law (scale-free) with a degree exponent γ usually in the range $2 \leq \gamma \leq 3$:

$$P(k) \sim k^{-\gamma} \tag{3.4}$$

3.4 Contact Maps

The undirected graph I work on is the contact map. The contact map is a first coarse-grained representation of the protein. In particular, it is the representation of the interactions among its constituents,. As already said, PDB files are structured with many three-d coordinates. Each of these coordinates is associated with an atom. For constructing an adjacency matrix (A) I have extracted all the coordinates of Cα atoms that are considered as the position of the residue attached to it. The edges of A are those couples of Cα atoms that are closer then some threshold ϵ and so are considered to be interacting. The dimension of A is equal to the number of Cα we find in the protein.

$$\begin{array}{lcl} A_{ij} & = & 1 \\ (x_i - x_j)^2 & \leq & \epsilon \end{array} \tag{3.5}$$

The contact map so obtained represents a graph (G,V,E). Whose nodes are represented with i,j and degrees k_i. This will be the standard notation I am going to mantain in the following section throughout all the thesis. In my case I set $\epsilon = 8$ and 12 ($\mathring{A}$) following in my approach [23].

3.5 Laplacian

From A, the Laplacian operator of a network, is derived as:

$$L = D - A \tag{3.6}$$

$$D_{\mathrm{ij}} = \delta_{\mathrm{ij}} k_{\mathrm{i}} \tag{3.7}$$

Where $\mathrm{d_i}$ is the number of contacts for each residue. The dimension of the Laplacian is the same as the adjacency matrix. In particular, the action of L on a N-dimensional lattice corresponds to the discretization of a N-dimensional elastic membrane, where L's eigenvalues represent the frequencies of the normal modes and L's eigenvectors represent the normal mode solutions or eigenfunctions [24]. With this analogy in mind, the eigenvalue decomposition of the Laplacian operator corresponds to searching for extremal values of the Rayleigh functional, vectors x that maximize or minimize the mutual distance between nodes in the network, expressed by the following semi-positive quadratic form:

$$\vec{x}^{\mathrm{T}} L \vec{x} = \sum_{i \sim j} (x_{\mathrm{i}} - x_{\mathrm{i}})^2 \tag{3.8}$$

The trivial solution corresponds to the 0 eigenvalue, in which all nodes have the same spatial coordinates and thus $x_{\mathrm{i}} = x_{\mathrm{j}}$ for every i,j. The non-trivial solutions seek for a minimal distance by imposing the orthogonality with the constant vector. If we hypothesize that the elastic potential schematized by the Laplacian operator is an approximation around the minimum of the Lennard-Jones potential-like function, modeling the interaction between protein residues, the 3D coordinates of Cα can be estimated by the components of the 3 eigenvectors associated with the 3 smallest positive eigenvalues of the Laplacian operator, thus providing a reconstruction of the 3D protein structure up to a linear transformation.
It is useful also to introduce the normalized Laplacian, as for several classification applications it resulted to perform better:

$$\mathcal{L} = \begin{cases} 1, & \text{if } i = j \\ 1/\sqrt{k_{\mathrm{i}} k_{\mathrm{j}}}, & \text{if i,j } \textit{adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

Once diagonalized, we will call the eigenvalues $(\lambda_0, .., \lambda_{\mathrm{n}})$.

3.5.1 Laplacian's spectral properties

In this section and the next, we start with a few basic facts about eigenvalues. Some simple upper bounds and lower bounds are stated. In this section we follow the treatment of [19]. Mohar in his survey [20]. As the first property to mention of the Laplacian and the normalized laplacian is:

Theorem 1 *The smallest eigenvalue λ_0=0.*

Proof:

$$\mathbf{1}^{\mathrm{T}} L \mathbf{1} = \sum_{ij} (d_{\mathrm{i}} \delta_{\mathrm{ij}}) - A_{\mathrm{ij}} = \sum_{i} (d_{\mathrm{i}} - d_{\mathrm{i}}) = 0 \tag{3.10}$$

The algebraic. connectivity of a graph G is the second-smallest eigenvalue of the Laplacian matrix of G.

Theorem 2 *The second smallest eigenvalue is greater than 0 if and only if G is a connected graph.*

Lemma 3 *Let G be a simple graph. Then:*

$$\lambda_1 \leq max\{d_i + d_j | i, j \in E(G)\} \tag{3.11}$$

where d_i is the degree of the node i. The equality holds just for connected graphs.

Theorem 4 *Let G be a simple graph. Denote by $r = max\{d_i + d_j|(i,j) \in E(G)\}$ and $s = max\{d_i + d_j|(i,j) \in E(G) - (i,j)\} with (i,j) \in E(G)$ such that $d_i + d_j = r$. Then:*

$$\lambda_1(G) \leq 2 + \sqrt{(r-2)(s-2)} \tag{3.12}$$

this result may further be improved.

Theorem 5 *Let G be a simple connected graph. Then:*

$$\lambda(G) \leq 2 + max\{\sqrt{(d_i + d_j - 2)(d_i + d_k - 2)}\} \tag{3.13}$$

where the maximum is taken over all pairs $(i,j),(i,k) \in E(G)$. Moreover, equality holds in 3.13 if and only if G is regular bipartite graph or a semiregular graph, or a path of order four.

3.5.2 Normalized Laplacian spectral properties

In this section are reported the equivalent theorems of the previous section for the normalized laplacian $\mathcal{L}$ where some change is neede. For example, we will see that the eigenvalues of any graph lie between 0 and 2.

Lemma 6 *For a graph G on n vertices, we have:*

1. $$\sum_i \lambda_i \leq n \tag{3.14}$$

   with equality holding if and only if G has no isolated vertices.

2. *For n≥ 2 :*

$$\lambda_1 \leq \frac{n}{n-1} \tag{3.15}$$

with equality holding if and only if G is the complete graph on n vertices. Also, for a graph G without isolated vertices, we have:

$$\lambda_{n\text{-}1} \geq \frac{n}{n-1} \tag{3.16}$$

3. *For a graph which is not a complete graph, we have $\lambda_1 \leq 1$*

4. *If G is connected, then $\lambda_1 \geq 0$. If λ_i=0 and $\lambda_{i+1} \neq 0$, then G has exactly i + 1 connected components.*

5. *For all $i \leq n-1$, we have:*

$$\lambda_i \leq 2 \tag{3.17}$$

with $\lambda_{n-1}= 2$ if and only if a connected component of G is bipartite and nontrivial.

6. *The spectrum of a graph is the union of the spectra of its connected components.*

Lemma 7 *The following statements are equivalent:*

1. *G is bipartite.*
2. *G has i + 1 connected components and $\lambda_{n-j} = 2$ for $1 \leq j \leq i$*
3. *For each λ_i, the value 2- λ_i is also an eigenvalue of G*

3.6 Laplacian Eigenmaps for dimensionality reduction and Data Representation

Enzymes are here considered in a 1 to 1 correspondence to contact map, as undirected graphs. The Laplacian associated to an undireced graph is useful for classification tasks as it can be associated to a diffusion problem that gives informations about embedding an associated manifold in an Hilbert space of some dimension, and this allows looking at the classification problem through a geometrical perspective. The algorithm works in the following way as [13] explains: Given k points $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_\mathrm{k}\}$ in $\mathbb{R}^\mathrm{l}$, we construct a weighted graph with k nodes, one for each point, and a set of edges connecting neighboring points. The embedding map is now provided by computing the eigenvectors of the graph Laplacian. The algorithmic procedure is formally stated below.

1. (constructing the adjacency graph). We put an edge between nodes i and j if $\boldsymbol{x}_\mathrm{i}$ and $\boldsymbol{x}_\mathrm{j}$ are "close". ϵ -neighborhoods (parameter $\epsilon \in \mathbb{R}$). Nodes i and j are connected by an edge if $||\boldsymbol{x}_\mathrm{i} - \boldsymbol{x}_\mathrm{j}||^2 < \epsilon$ where the norm is the usual Euclidean norm in R^l .
2. Compute eigenvalues and eigenvectors for the generalized eigenvector problem.

3.6.1 Justification

Let us first show that the embedding provided by the Laplacian eigenmap algorithm preserves local information optimally in a certain sense. The following is based on standard spectral graph theory[14]. Recall that given a data set, we construct a weighted graph $G = (V, E)$ with edges connecting nearby points to each other. For the purposes of this discussion the graph is connected. Consider the problem of mapping the weighted graph G to a line so that connected points stay as close together as possible. Let $\boldsymbol{y} = (y_1, y_2, ..., y_\mathrm{n})^\mathrm{T}$ be such a map. For this map to be defined we choose the minimization of the cost function:

$$H = \sum_{i,j} (y_\mathrm{i} - y_\mathrm{j})^2 A_\mathrm{ij} \tag{3.18}$$

This minimization is equivalent to the minimization of $C = \boldsymbol{y}^{\mathrm{T}} L \boldsymbol{y}$ infact:

$$\sum_{i,j} (y_{\mathrm{i}} - y_{\mathrm{j}})^2 A_{\mathrm{ij}} = \sum_{i} D_{\mathrm{ii}} {y_{\mathrm{i}}}^2 + \sum_{j} D_{\mathrm{jj}} {y_{\mathrm{j}}}^2 - 2 \sum_{i,j} A_{\mathrm{ij}} y_{\mathrm{i}} y_{\mathrm{j}} := 2 \boldsymbol{y}^{\mathrm{T}} L \boldsymbol{y} \tag{3.19}$$

Note that this calculation also shows that L is positive semidefinite. Therefore, the minimization problem reduces to finding:

$$\underset{y, \boldsymbol{y}^{\mathrm{T}} D \boldsymbol{y}}{\operatorname{argmin}} \boldsymbol{y}^{\mathrm{T}} L \boldsymbol{y} \tag{3.20}$$

Where ${}^{\mathrm{T}} D \boldsymbol{y}$ removes arbitrary scaling factor in the embedding. Matrix D provides a natural measure on the vertices of the graph. The bigger the value D_{ii} (corresponding to the ith vertex) is, the more "important" is that vertex. As L is semidefinite positive we have that $\boldsymbol{y}$ that minimizes (3.18) is given by the minimum eigenvalue solution of the generalized eigenvalue problem:

$$L \boldsymbol{y} = \lambda D \boldsymbol{y} \tag{3.21}$$

Where we note that if $\mathbf{1}$ is the vector of all 1 then its eigenvalue $\lambda = 0$ then we further ask:

$$\boldsymbol{y}^{\mathrm{T}} \mathbf{1} = 0 \tag{3.22}$$

So the solution coincides with the eigenvector associated with the second smallest eigenvalue. Now consider the more general problem of embedding the graph into m-dimensional Euclidean space. The embedding is given by the k×m matrix $\boldsymbol{Y} = [y_1, y_2, ..., y_{\mathrm{m}}]$, where the ith row provides the embedding coordinates of the ith vertex. Similarly we have to minimize:

$$\sum_{i,j} ||\boldsymbol{y}^{(\mathrm{i})} - \boldsymbol{y}^{(\mathrm{j})}||^2 A_{\mathrm{ij}} = tr(\boldsymbol{Y}^{\mathrm{T}} L \boldsymbol{Y}), \tag{3.23}$$

Where $\boldsymbol{y}^{(\mathrm{i})} = [{y^{(\mathrm{i})}}_1, ..., {y^{(\mathrm{i})}}_{\mathrm{m}}]^{\mathrm{T}}$ is the m-dimensional representation of the ith vertex.

$$\underset{tr \boldsymbol{Y}^{\mathrm{T}} D \boldsymbol{Y} = \mathbb{I}}{\operatorname{argmin}} \ \boldsymbol{Y}^{\mathrm{T}} L \boldsymbol{Y} \tag{3.24}$$

For the one-dimensional embedding problem, the constraint prevents collapse of the node coordinates onto a point. For the m-dimensional embedding problem, the constraint presented above prevents collapse onto a subspace of dimension less than m−1 (m if, as in one-dimensional case, we require orthogonality to the constant vector). Standard methods show that the solution is provided by the matrix of eigenvectors corresponding to the lowest eigenvalues of the generalized eigenvalue problem $L \boldsymbol{y} = \lambda D \boldsymbol{y}$.

3.6.2 Using Laplace Beltrami operator for manifold analysis

In the following section we describe which properties of the Laplacian eigenvalues on a graph are desirable for embedding problems and are shared with Laplace-Beltrami operator on manifolds. Let M be a smooth, compact, m-dimensional Riemannian manifold. If the manifold is embedded in $\mathbb{R}^{\mathrm{l}}$, the Riemannian structure (metric tensor) on the

manifold is induced by the standard Riemannian structure on $\mathbb{R}^l$. As we did with the graph, we are looking here for a map from the manifold to the real line such that points close together on the manifold are mapped close together on the line. Let f be such a map. Assume that $f : M \longrightarrow \mathbb{R}$ is twice differentiable. Consider two neighboring points $\boldsymbol{x}, \boldsymbol{z} \in M$. Let's try to estimate the distance between two embedded points in terms of the distance of two points in the manifold:

$$|f(x) - f(z)| \leq dist_{\mathrm{M}}(x, z)||\nabla f(\boldsymbol{x})|| + o(dist_{\mathrm{M}}(x, z)) \tag{3.25}$$

The gradient $\nabla f(\boldsymbol{x})$ is a vector in the tangent space $\mathrm{TM_x}$, such that given another vector $\boldsymbol{v} \in \mathrm{TM_x}$, $df(\boldsymbol{v}) = \langle \nabla f(\boldsymbol{x}), \boldsymbol{v} \rangle_{\mathrm{M}}$. Let $\mathrm{l} = \mathrm{dist_M(x,\ z)}$. Let c(t) be the geodesic curve parameterized by length connecting $\mathrm{x} = \mathrm{c}(0)$ and $\mathrm{z} = \mathrm{c(l)}$. Then:

$$f(z) = f(x) + \int_0^l df(c(t))\, dt \tag{3.26}$$

and so we have that:

$$f(z) = f(x) + \int_0^l \langle \nabla f(c(t)), c'(t) \rangle dt \tag{3.27}$$

Now by Schwartz inequality:

$$\langle \nabla f(c(t)), c'(t) \rangle \leq ||\nabla f(c(t))||||c'(t)|| \tag{3.28}$$

As $||c'(t)|| = 1$ then if we expand in Taylor series at the first order:

$$\langle \nabla f(c(t)), c'(t) \rangle \leq ||\nabla f(x)|| + o(t) \tag{3.29}$$

integrating:

$$|f(z) - f(x)| \leq l||\nabla f(x)|| + o(l) \tag{3.30}$$

as l is the geodesic distance in M between x,z then the statement is verified. Thus, we see that $||\nabla f||$ provides us with an estimate of how far apart f maps nearby points.We therefore look for a map that best preserves locality on average by trying to find:

$$\underset{||f||^2=1}{\operatorname{argmin}} \int_M ||\nabla f||^2 \tag{3.31}$$

where the integral is taken with respect to the standard measure on a Riemannian manifold. Note that minimizing $\int_{\mathcal{M}} ||\nabla f(x)||^2$ corresponds to minimizing on a graph $Lf = \sum_{i,j} (f_{\mathrm{i}} - f_{\mathrm{j}})^2 W_{\mathrm{ij}}$. Here, f is a function on vertices, and $\mathrm{f_i}$ is the value of f on the i-th node of the graph. It turns out that minimizing the objective function reduces to finding eigenfunctions of the Laplace Beltrami operator $\mathcal{L}$.

$$\mathcal{L}f := -div\nabla(f), \tag{3.32}$$

where div is the divergence of the vector field. It follows from the Stokes' theorem that $-$ div and ∇ are formally adjoint operators, that is, if f is a function and $\boldsymbol{X}$ is a vector field, then:

$$\int_M \langle \nabla f, \boldsymbol{X} \rangle = \int_M div\boldsymbol{X} f \tag{3.33}$$

and so:

$$\int_M ||\nabla f||^2 = \int_M \mathcal{L}(f) f \tag{3.34}$$

We see that $\mathcal{L}$ is positive semidefinite. f that minimizes $\int_M ||\nabla f||^2$ 2 has to be an eigenfunction of $\mathcal{L}$. The spectrum of L on a compact manifold M is known to be discrete (Rosenberg, 1997). Let the eigenvalues (in increasing order) be $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq ..., \leq \lambda_\mathrm{n}$ and let $\mathrm{f_i}$ be the eigenfunction corresponding to eigenvalue λ_i. It is easily seen that $\mathrm{f_0}$ is the constant function that maps the entire manifold to a single point. To avoid this eventuality, we require (just as in the graph setting) that the embedding map f be orthogonal to $\mathrm{f_0}$. It immediately follows that $\mathrm{f_1}$ is the optimal embedding map. Following the arguments of the previous section, we see that:

$$x \longrightarrow (f_1(x), ..., f_\mathrm{m}(x)) \tag{3.35}$$

provides the optimal m-dimensional embedding

3.7 PCA

Given a set $\{\boldsymbol{x}_1, ..., x_\mathrm{n}\}$ with $\boldsymbol{x}_\mathrm{i} \in \mathbb{R}^\mathrm{l}$ such that vector space and having some distribution whose covariance matrix

$$\Sigma^{(\mathrm{ij})} = \sum_k \frac{\boldsymbol{x}^{(\mathrm{i})}{}_\mathrm{k} \boldsymbol{x}^{(\mathrm{j})}{}_\mathrm{k}}{n^2} - \sum_k \frac{\boldsymbol{x}^{(\mathrm{i})}{}_\mathrm{k}}{n} \sum_l \frac{\boldsymbol{x}^{(\mathrm{j})}{}_\mathrm{l}}{n} \tag{3.36}$$

Then PCA is a dimensional reduction of my space that performs a rotation of the axis of my l-dimensional space, suche that the covariance matrix becomes diagonal and its eigenvectors are the direction in which the distribution is more spread making now our variables indipendently distributed and its eigenvector represent how much spread data are. In this process we want to use less memory loosing the least information as possible and so the least precision. One way to encode these points is to represent a lower-dimensional version of them. For each $\boldsymbol{x}_\mathrm{i} \in \mathbb{R}^\mathrm{n}$ we find $\boldsymbol{y}_\mathrm{i} \in \mathbb{R}^\mathrm{l}$ with $l \leq n$. We want to find an encoding function $f : \mathbb{R}^\mathrm{n} \longrightarrow \mathbb{R}^\mathrm{l}$ and a decoding function $g : \mathbb{R}^\mathrm{l} \longrightarrow \mathbb{R}^\mathrm{n}$ such that $\boldsymbol{x} \approx g(f(\boldsymbol{x}))$. PCA is defined by our choice of the decoding function. Let $\boldsymbol{x} = g(\boldsymbol{y}) = \boldsymbol{D}\boldsymbol{y}$ where $\boldsymbol{D} \in \mathbb{R}^{\mathrm{nxl}}$. The l-columns are orthogonal one another and normalized to unity suche that the encoding function is unique. In the costruction of the algorithm for PCA we first have to choose $\boldsymbol{c}$ such that it minimizes the distance between $\boldsymbol{x}$ and its decoded counterpart. We call it $\boldsymbol{c}^*$. The minimization is with respect to the L^2 norm:

$$\boldsymbol{c}^* = \underset{\boldsymbol{c}}{\mathrm{argmin}} [\boldsymbol{x} - g(\boldsymbol{c})]_2{}^2 \tag{3.37}$$

The equation can be rephrased as

$$c^* = \underset{\boldsymbol{c}}{\mathrm{argmin}} (\boldsymbol{x} - g(\boldsymbol{c}))^\mathrm{T} (\boldsymbol{x} - g(\boldsymbol{c})) \tag{3.38}$$

Then we find that:

$$\boldsymbol{c}^* = \underset{\boldsymbol{c}}{\mathrm{argmin}} g(\boldsymbol{c})^\mathrm{T} g(\boldsymbol{c}) - 2\boldsymbol{x}^\mathrm{T} g(\boldsymbol{c}) \tag{3.39}$$

As already stated g($\boldsymbol{c}$)=$\boldsymbol{D}\boldsymbol{c}$, then we can rephrase as:

$$\boldsymbol{c}^* = \underset{\boldsymbol{c}}{\mathrm{argmin}} \boldsymbol{c}^\mathrm{T} \boldsymbol{D}^\mathrm{T} \boldsymbol{D}\boldsymbol{c} - 2\boldsymbol{x}^\mathrm{T} \boldsymbol{D}\boldsymbol{c} \tag{3.40}$$

We solve the problem via vector calculus:

$$\nabla_{\boldsymbol{c}}(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{c} - 2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{c}) = 0 \tag{3.41}$$

then:

$$\boldsymbol{c} = \boldsymbol{D}^{\mathrm{T}}\boldsymbol{x} \tag{3.42}$$

This makes the algorithm efficient and one can make the encription via matrix-vector moltiplication. So the encoder function:

$$f(\boldsymbol{x}) = \boldsymbol{D}^{\mathrm{T}}\boldsymbol{x} \tag{3.43}$$

having that the decoding function is:

$$r(\boldsymbol{x}) = f(g(\boldsymbol{x})) = \boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{x} \tag{3.44}$$

Now is time to find $\boldsymbol{D}$. To do so, we minimize the L^2 distance between inputs and reconstructions. Having that this must be valid for all the data we must now minimize the Frobenius norm of the matrix of all errors computed over all dimensions and points:

$$\boldsymbol{D}^{*} = \underset{\boldsymbol{D}}{\operatorname{argmin}} \sqrt{\sum_{i,j} ({x_{\mathrm{j}}}^{(\mathrm{i})} - r(\boldsymbol{x}^{(\mathrm{i})})_{\mathrm{j}})^2} \tag{3.45}$$

Where as already said $\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}$=$\boldsymbol{I}_{\mathrm{l}}$. To derive the algorithm let's consider the case for l=1.

$$\boldsymbol{d}^{*} = \underset{\boldsymbol{d}}{\operatorname{argmin}} = \sum_{i} ||\boldsymbol{x}^{(\mathrm{i})} - \boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}\boldsymbol{x}^{(\mathrm{i})}||_2{}^2 \tag{3.46}$$

This formulation is not pleasing statistically and so we rephrase it considering $\boldsymbol{d}^{\mathrm{T}}\boldsymbol{x}^{(\mathrm{i})}$ as a scalar and exploiting the commutativity of the scalar product:

$$\boldsymbol{d}^{*} = \underset{\boldsymbol{d}}{\operatorname{argmin}} = \sum_{i} ||\boldsymbol{x}^{(\mathrm{i})} - \boldsymbol{x}^{(\mathrm{i})\mathrm{T}}\boldsymbol{d}\boldsymbol{d}||_2{}^2 \tag{3.47}$$

With always the constraint on $\boldsymbol{d}$ to be unitary.

If I then define as $\boldsymbol{X}$ being the $p \times n$ then I can rephrase the problem as:

$$\boldsymbol{d}^{*} = \underset{\boldsymbol{d}}{\operatorname{argmin}} Tr(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}) \tag{3.48}$$

and so it is valid

$$\underset{\boldsymbol{d}}{\operatorname{argmin}} Tr(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}) - Tr(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}) - Tr(\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}) + Tr(\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}) \tag{3.49}$$

that after noticing that minimization doesn't depend on some factors and for the symmetric nature of scalar product:

$$\underset{\boldsymbol{d}}{\operatorname{argmin}} - 2Tr(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}) + Tr(\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}) \tag{3.50}$$

exploiting the ciclicity invariance of the and the normalization constraint $\boldsymbol{d}^{\mathrm{T}}\boldsymbol{d} = 1$, we arrive at the equivalent formulation of the encoding problem as:

$$\underset{\boldsymbol{d}}{\operatorname{argmax}} Tr(\boldsymbol{d}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{d}) \tag{3.51}$$

with $\boldsymbol{d}^{\mathrm{T}}\boldsymbol{d}$=1 .

3.8 t-SNE

In this section we describe the algorithm as described in [11]. t-SNE is an algorithm to visualize high dimensional data in lower dimensional spaces (2 or 3). In contrast with techniques that represent data preserving structure at high distances t-SNE provides a method for preserving local neighborhood in points mapped onto a new (lower-dimensional) space. This method provides a map $f : \mathcal{X} \longrightarrow \mathcal{Y}$ where $\mathcal{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ and $\mathcal{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_n\}$ where $\boldsymbol{x}_i \in \mathbb{R}^l$ $(l >> 1)$and $\boldsymbol{y}_i \in \mathbb{R}^m$ where $m = 2, 3$. The aim of dimensionality reduction is to preserve as much of the significant structure of the high-dimensional data as possible in the low-dimensional map. For high-dimensional data, that lie on a low-dimensional non-linear manifold, it is usually more important to keep the low-dimensional representations of very similar datapoints close together, which is typically not possible with a linear mapping. t-SNE is capable of capturing much of the local structure of the high-dimensional data very well, while also revealing global structure such as the presence of clusters at several scales.

Let's consider $\mathcal{X}$ and define the Euclidean distance $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2$ between points in it.

t-SNE as a first step transforms the distance into conditional probability distribution.The similarity of datapoint $\boldsymbol{x}_j$ to datapoint $\boldsymbol{x}_i$ is the conditional probability, $p_{j|i}$, that $\boldsymbol{x}_i$ would pick $\boldsymbol{x}_j$ as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at $\boldsymbol{x}_i$.

$$p_{j|i} = \frac{exp\{-\frac{d(\boldsymbol{x}_i, \boldsymbol{x}_j)^2}{2{\sigma_i}^2}\}}{\sum_{k \neq i} exp\{\frac{-d(\boldsymbol{x}_k, \boldsymbol{x}_i)^2}{2{\sigma_i}^2}\}} \tag{3.52}$$

. Because we are only interested in modeling pairwise similarities, we set the value of $p_{i|i}$ to zero. For the lower dimensional counterpart whose distance between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ is $d(\boldsymbol{y}_i, \boldsymbol{y}_j) = ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2$ it is possible to compute a similar conditional probability which we denote with $q_{j|i}$

$$q_{j|i} = \frac{(1 + ||\boldsymbol{y}_j - \boldsymbol{y}_i||^2)^{-1}}{\sum_{k \neq i}(1 + ||y_k - y_i||^2)^{-3}} \tag{3.53}$$

. Again, since we are only interested in modeling pairwise similarities, we set $q_{i|i} = 0$. The t-Student distribution has been choosen as it has heavier tails and so bigger distances are put far apart and the problem of overcrowding in low dimensional spaces is overcome. A second step consists on symmetrizing the probabilities.

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2} q_{ij} = \frac{q_{j|i} + q_{i|j}}{2}$$

$$p_{ij} = \frac{exp\{\frac{d(\boldsymbol{x}_i, \boldsymbol{x}_j)^2}{2\sigma^2}\}}{\sum_k \sum_{l \neq k} exp\{\frac{d(\boldsymbol{x}_l, \boldsymbol{x}_k)^2}{2\sigma^2}\}} \tag{3.54}$$

Note that in these way we have weights different from zero just for a couple of different elements,i.e. $p_{ii} = 0$. Let $\boldsymbol{Y}$ be $\{y_1,...,y_n\}$ the set of 2 dimensional vectors, and

$$q_{ij} = \frac{(1 + ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + ||\boldsymbol{y}_k - \boldsymbol{y}_l||^2)^{-1}}$$

(3.55)

If the map points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ correctly model the similarity between the high-dimensional datapoints $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the conditional probabilities p_{ji} and q_{ji} will be equal. Motivated by this observation,t-SNE aims to find a low-dimensional data representation that minimizes the mismatch between p_{ji} and q_{ji}.

We have that t-SNE produces a map $f : X \longrightarrow Y$ by minimizing the cost function :

$$H(p_{ij}|q_{ij}) = -\sum_{ij} p_{ij} log(\frac{p_{ij}}{q_{ij}}) \tag{3.56}$$

that is called Kullback-Leibner divergence that is not convex neither symmetric . . For this reason the minimization problem is addressed via greatest descent algorithm and so choosing different starting points we will have different results corresponding to different minima. By using a heavy-tailed distribution to measure similarities in the lowdimensional map, t-SNE allows points that are only slightly similar to be visualized much further apart in the map. This typically leads to very good visualizations compared with other techniques.However, SNE suffers from a crowding problem that is the result of the exponential volume difference between high and low-dimensional spaces .

Although the simple algorithm produces visualizations that are often much better than those produced by other non-parametric dimensionality reduction techniques, the results can be improved further by using either of two tricks. The first trick, which we call “early compression”, is to force the map points to stay close together at the start of the optimization. When the distances between map points are small, it is easy for clusters to move through one another so it is much easier to explore the space of possible global organizations of the data. Early compression is implemented by adding an additional L2-penalty to the cost function that is proportional to the sum of square distances of the map points from the origin. The magnitude of this penalty term and the iteration at which it is removed are set by hand, but the behavior is fairly robust across variations in these two additional optimization parameters. A less obvious way to improve the optimization, which we call “early exaggeration”, is to multiply all of the p_{ij}'s by, for example, 4, in the initial stages of the optimization. This means that almost all of the q_{ij}'s, which still add up to 1, are much too small to model their corresponding p_{ij}'s. As a result, the optimization is encouraged to focus on modeling the large p_{ij}'s by fairly large q_{ij}'s. The effect is that the natural clusters in the data tend to form tight widely separated clusters in the map. This creates a lot of relatively empty space in the map, which makes it much easier for the clusters to move around relative to one another in order to find a good global organization.

3.9 UMAP

Another dimensional reduction can be tried via UMAP and see whether it brings better results. UMAP is an algorithm based in manifold theory and topological data analysis.

UMAP formulation relies on topological arguments and **constructions of fuzzy simplicial sets, that contain uniformly data.** UMAP has no computational restrictions on embedding dimension, making it viable as a general purpose dimension reduction technique for machine learning. We now expose the computational implementation of the algorithm careless of the theoretical motivations that lay on category theory, and management of simplicial sets (outside the scope of this thesis). UMAP can be put in the class of k-neighbour based graph learning algorithms such as Laplacian Eigenmaps, Isomap and t-SNE. As with other k-neighbour graph based algorithms, UMAP can be described in two phases. In the first phase a particular weighted k-neighbour graph is constructed. In the second phase a low dimensional layout of this graph is computed. The differences between all algorithms in this class amount to specific details in how the graph is constructed and how the layout is computed. The formulations of the algorithm is based on the following axioms about data:

- there exists a manifold on which the data would be uniformly distributed.
- the underlying manifold of interest is locally connected.
- Preserving the topological structure of this manifold is the primary goal.

Any algorithm that attempts to use a mathematical structure akin to a k-neighbour graph to approximate a manifold must follow a similar basic structure.

- Graph construction:
  1. Construct a weighted k-neighbour graph
  2. Apply some transform on the edges to ambient local distance.
  3. Deal with the inherent asymmetry of the k-neighbour graph.
- Graph layout:
  1. Define an objective function that preserves desired characteristics of this k-neighbour graph.
  2. Find a low dimensional representation which optimizes this objective function.(in t-SNE KL divergence and UMAP cross-entropy)

The idea here is to construct fuzzy simplicial sets that are covering the points in the manifold. We can construct the fuzzy simplicial set local to a given point x by finding the k nearest neighbors, generating the appropriate normalised distance on the manifold, and then converting the finite metric space to a simplicial set via the functor FinSing, which translates into exponential of the negative distance. Infact, given X be $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_\mathrm{n}\}$ with a metric $d : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}^+$. Given an input hyperparameter k, for each $\boldsymbol{x}_\mathrm{i}$ we compute the set $\{\boldsymbol{x}_\mathrm{i}, ..., \boldsymbol{x}_{\mathrm{i_k}}\}$ of the k nearest neighbors of $\boldsymbol{x}_\mathrm{i}$ under the metric d. This computation can be performed via any nearest neighbour or approximate nearest neighbour search algorithm. For each $\boldsymbol{x}_\mathrm{i}$ exist ρ_i and σ_i such that:

$$\rho_\mathrm{i} = min\{d(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i_j}}) | 1 \leq j \leq k\} \tag{3.57}$$

and σ_i such that :

$$\sum_{j=1} exp\{\frac{-max(0, d(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i_j}}))}{\sigma_\mathrm{i}}\} = log_2(k) \tag{3.58}$$

The graph associated is =(V,E,w) where the vertices V of are simply the set X the edges ρ_{i}= $\{(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{i_j}})|1 \leq j \leq k\}$ and weights:

$$w_{\mathrm{i_j}} = exp\{\frac{-max(0, d(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{i_j}}) - \rho_{\mathrm{i}})}{\sigma_{\mathrm{i}}}\} \tag{3.59}$$

. In practice UMAP uses a force directed graph layout algorithm in low dimensional space. A force directed graph layout utilizes of a set of attractive forces applied along edges and a set of repulsive forces applied among vertices. Any force directed layout algorithm requires a description of both the attractive and repulsive forces. The algorithm proceeds by iteratively applying attractive and repulsive forces at each edge or vertex. This amounts to a non-convex optimization problem. Convergence to a local minima is guaranteed by slowly decreasing the attractive and repulsive forces in a similar fashion to that used in simulated annealing. In UMAP the attractive force between two vertices i and j at coordinates $\boldsymbol{y}_{\mathrm{i}}$ and $\mathrm{y_j}$ respectively, is determined by:

$$\frac{(-2ab + ||\boldsymbol{y}_{\mathrm{i}} - \boldsymbol{y}_{\mathrm{j}}||^{2(\mathrm{b\text{-}1})}{}_2)}{(1 + ||\boldsymbol{y}_{\mathrm{k}} - \boldsymbol{y}_{\mathrm{l}}||^2{}_2)} w_{\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}}}(\boldsymbol{y}_{\mathrm{i}} - \boldsymbol{y}_{\mathrm{j}}) \tag{3.60}$$

where a and b are hyper-parameters. Repulsive forces are computed via sampling due to computational constraints. Thus, whenever an attractive force is applied to an edge, one of that edge's vertices is repulsed by a sampling of other vertices.The algorithm can be initialized randomly but in practice, since the symmetric Laplacian of the graph G is a discrete approximation of the Laplace Beltrami operator of the manifold, we can use a spectral layout to initialize the embedding. [U+008C]is provides both faster convergence and greater stability within the algorithm. UMAP is an algorithm that tries to preserve global structures, however there are severe drawbacks due to the lack of measures for global structures or even definitions of what global structures are.

3.9.1 Comparison between UMAP and t-SNE

The problem is always to look for relationships between two points in high dimensional space X and low dimensional embedded space Y. t-SNE defines input probabilities in three stages. First, for each pair of points, i and j, in X, a pair-wise similarity, $\mathrm{p_{ij}}$, is calculated, Gaussian with respect to the Euclidean distance between $\boldsymbol{x}_{\mathrm{i}}$ and $\boldsymbol{x}_{\mathrm{j}}$ as in (3.54), this is the definition of the distribution in high-dimension space. Following the same steps in t-SNE section and using the same notation we now spot the differences between the two algorithm traced in the differences among the different definitions of cost functions and probability distributions. The starting distribution in both cases is a Gaussian whose variables are the distances defined in the graph. In t-SNE:

$$v_{j|i} = exp\{\frac{d(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}})^2}{2\sigma^2}\} \tag{3.61}$$

and

$$w_{j|i} = (1 + ||\boldsymbol{y}_{\mathrm{i}} - \boldsymbol{y}_{\mathrm{j}}||^2)^{\text{-}1} \tag{3.62}$$

$$p_{\mathrm{ij}} = \frac{v_{j|i} + v_{i|j}}{\sum_k \sum_{l \neq k} (v_{l|k} + v_{k|l})} \tag{3.63}$$

and,

$$q_{\mathrm{ij}} = \frac{w_{j|i} + w_{i|j}}{\sum_k \sum_{l \neq k} (w_{l|k} + w_{k|l})} \tag{3.64}$$

with

$$H(p_{\mathrm{ij}}|q_{\mathrm{ij}}) = -\sum_{ij} p_{\mathrm{ij}} log(\frac{p_{\mathrm{ij}}}{q_{\mathrm{ij}}}) \tag{3.65}$$

as the cost function to be minimized thanks to gradient descent methods. In UMAP we on the other hand we stop at the definition of distances not caring about normalization to probability distribution and work with them as the functions that define simplicial sets. As we said in the previous section:

$$v_{j|i} = exp[\frac{-d_{\mathrm{ij}} - \rho_{\mathrm{i}}}{\sigma_{\mathrm{i}}}] \tag{3.66}$$

Calculated just for k-nearest neighbours and $v_{j|i}$ for all the rest of couples.. $\mathrm{d_{ij}}$ is the distance between $\boldsymbol{x}^{\mathrm{i}}$ and $\boldsymbol{x}^{\mathrm{j}}$, which UMAP does not require to be Euclidean. ρ_{i} is the distance to the nearest neighbor of i. σ_{i} is the normalizing factor, that plays a similar role to the perplexity-based calibration of σ_{i} in t-SNE. Symmetrization is carried out by fuzzy set union can be expressed as:

$$v_{ij} = v_{j|i} + v_{i|j} + v_{j|i}v_{i|j} \tag{3.67}$$

The low dimensional similarities are given by:

$$w_{j|i} = (1 + a||\boldsymbol{y}_{\mathrm{i}} - \boldsymbol{y}_{\mathrm{j}}||_2{}^{2\mathrm{b}})^{\text{-}1} \tag{3.68}$$

where a and b are user-defined positive value. The cost function to be minimized via gradient descent algorithm is:

$$\sum_{i \neq j} v_{\mathrm{ij}} log(\frac{v_{\mathrm{ij}}}{w_{\mathrm{ij}}}) + (1 - v_{\mathrm{ij}}) log(\frac{1 - v_{\mathrm{ij}}}{1 - w_{\mathrm{ij}}}) \tag{3.69}$$

3.9.2 Spectral techniques for embedding and clustering

While there is a large body of work on dimensionality reduction in general, most existing approaches do not explicitly take into account the structure of the manifold on which the data may possibly reside. The justification comes from the role of the Laplacian operator in providing an optimal embedding. The Laplacian of the graph obtained from the data points may be viewed as an approximation to the Laplace-Beltrami operator defined on the manifold.The embedding maps for the data come from approximations to a natural map that is defined on the entire manifold. Laplacian preserves the local structure of a manifold so data that are outliers don't weight much. A byproduct of this is that the algorithm implicitly emphasizes the natural clusters in the data.

3.10 Description of dataset

As already mentioned, I have 4061 different PDB structures related to 494 different UniProt sequences. For each PDB I have extracted the contact map and calculated the Laplacian,its eigenvalues, eigenvectors, number of nodes, number of components and number of links. Besides these quantitative labels there are also labels for taxonomy, KM, organism and EC values. Here below some features of interest about the dataset.

Chapter 4

Description of contact maps and Laplacian

The scope of this chapter is the description of the networks obtained. In the first section, I will describe the contact maps obtained from the dataset using the tools introduced in section **network properties related to distance matrix**. Then I will describe laplacian's (L) and normalized laplacian's $\mathcal{L}$ spectra for all enzymes for both the cases of 8 and 12 $\mathring{A}$. Using these informations, I prepare the reader for the rest of the analysis contained in the next chapter. When I speak about smallest eigenvalues, I refer to the smallest eigenvalues after the (λ_0=0) eigenvalue, which I always neglect as it is trivial and all contact maps are made by one unique component. After this analysis, I find that the distributions of 40 smallest and 40 biggest eigenvalues of the laplacian, and properties of contact maps obtained with threshold at 8 $\mathring{A}$ seem to contain enough information for PCA,t-SNE,UMAP representation described in next chapter. For this reason, in the next chapter I will not show the results for the the normalized laplacian at 8 $\mathring{A}$ and for the laplacian and normalized laplacian at 12 $\mathring{A}$. Two important results of this chapter are the linear dependence of the logarithm of the 40 smallest eigenvalues with the logarithm of the number of nodes, and the logarithm of the 40 biggest eigenvalues with the logarithms of the link density for both thresholds (thus a polynomial relationship). Another important result of this section is the observation that the dataset contains families of enzymes having the same uniprot code (aminoacidic sequence) but different PDBs (structure) that seem to share similar laplacian spectra and network's properties, confirming the hypothesis of similar spectral representation for similar proteins. It is a general fact from which I highlight just the most striking redundance, the 'P11838' protein in Fungi.

4.1 Contact maps

I here report the description of the contact maps obtained with threshold at 8 $\mathring{A}$ and 12 $\mathring{A}$. Their description will be brought on the three aspects I have already spoken about:

- small world properties described via relation 3.3
- scale free invariance described by relation 3.4
- average clustering coefficient described via formula 3.2

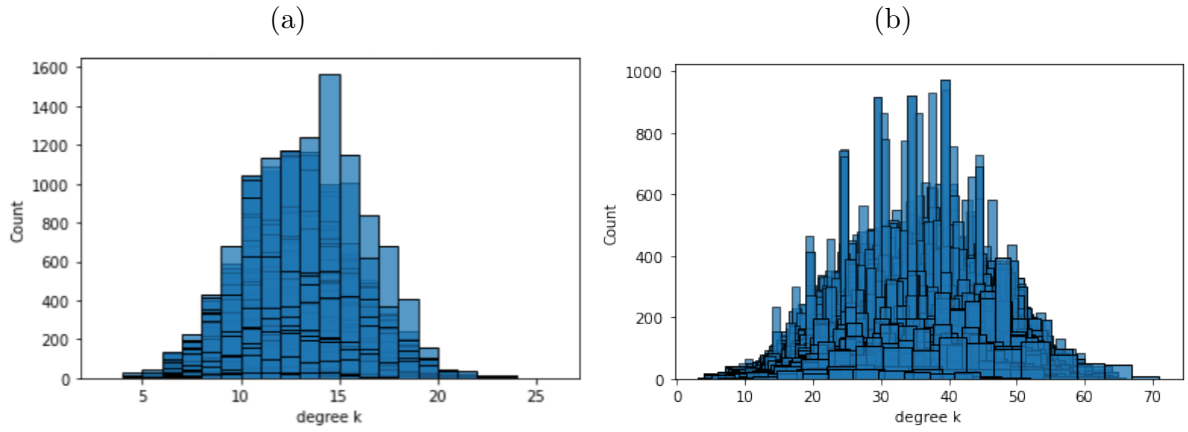


Figure 4.1: In these figures in the x-axis are the degrees for each node of one enzyme , and on the y-axis the number counts for each enzyme in the case of threshold 8 $\mathring{A}$ (a) and 12 $\mathring{A}$ (b). Here are represented superposed all the enzymes to show that the degree distributions are not scale invariant in the sense of equation (3.4) in neither case. In that case I would have had a decaying graph but that is not the case and infact there is an initial increase of the probability and then a fall.

As for the first item of the above list, it can be seen that,from fig.[4.7a] and [4.7b] these enzyme adhere (3.3) for both 8 and 12 $\mathring{A}$, and so they have the small world property with l_0=0.695 and l_0=0.708 . Infact from a fit I obtain the relation:

$$\begin{aligned} log(\bar{l}) &= 0.695 log(N) + 1.117 \\ log(\bar{l}) &= 0.708 log(N) + 1.405 \end{aligned} \tag{4.1}$$

Where:

$$\bar{l} = \frac{1}{N} \sum_{i \leq j}^{N} d_{\mathrm{ij}} \tag{4.2}$$

And the parameters 0.694 and 1.117 and 0.708, are found by fitting data and the relative $\boldsymbol{R}^2$ measure is 0.9. The scale invariance, however, is not matched, indeed, looking at the distribution of the degrees for each enzyme of figure 4.1a and 4.1b, no one has the $k^{-\gamma}$. The first evidence of structure in the dataset, comes from the distribution of link density in fig 4.2a for 8 $\mathring{A}$ contact maps, and less evidently in 4.2b for 12 $\mathring{A}$ contact maps. In the former it can be seen that there are three peaks, in the latter, the three peaks smear. I interpret this behavior as the fact that different aminoacidic sequences result more different one another at 8 $\mathring{A}$. The complete description in terms of the label distributions of the three peaks of figure 4.2a is contained in figures from 4.3a to 4.5d. I have decided not to go further in the analysis of the details. One of these three peaks contains the 554 fungi 3d structures belonging to the same uniprot 'P11838', that affect peak size and shape. Shifting to clustering coefficients, we start from figure 4.2a, where there is a peak around 0.53 with 1198 enzymes. It turns out that also this peak is formed, for the majority, by P11838 structures. It is by now clear that P11838 is constituted by very similar structures. This behavior is found for other uniprot sequences too.

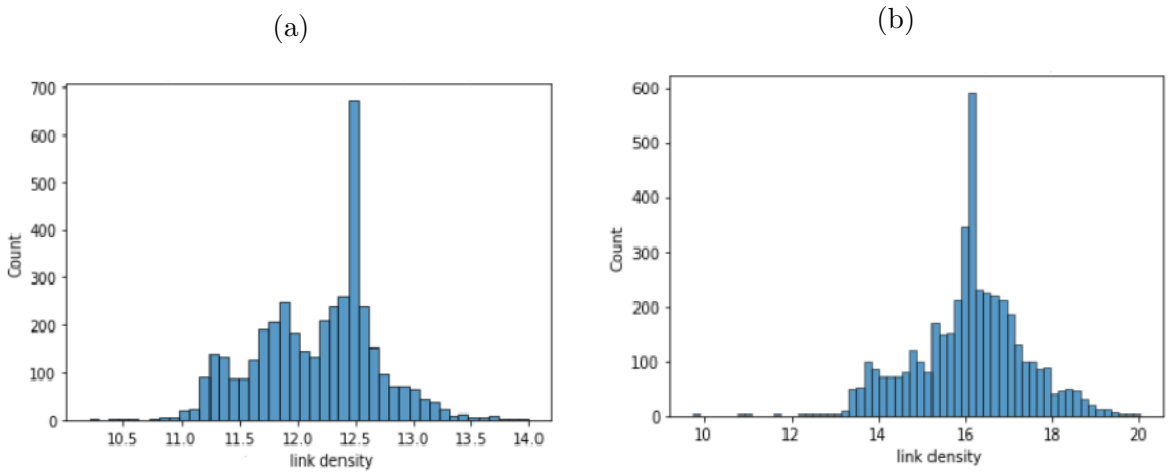


Figure 4.2: In these two figures are represented the density of links in the distribution of the contact maps calculated with threshold 8 $\mathring{A}$ (a), 12 $\mathring{A}$ (b). The three peaks in figure (a) are mostly due to redundancies of PDB code for the same uniprot as in the case of the highest peak where there is an over-representation of 554 structures from P11838. In figure (b) the peaks are still there even though,the distribution is more smeared. The above observations suggest that at coarser grained scales enzymes that don't belong to the same uniprot maybe more different and so, the capability of spotting some cluster is diminished.

4.1.1 considerations about biggest and smallest eigenvalues distributions

In the following subsection I am going to comment the figures that describe the distribution of biggest and smallest eigenvalues for both normalized and not normalized laplacian with threshold 8 and 12 $\mathring{A}$. These comments are made in the perspective of the use of these data for comparison of networks with PCA,t-SNE and UMAP representation. From tables 4.2, 4.1, 4.5 and 4.6, it can be seen (for both L and $\mathcal{L}$ and both thresholds) that the logarithm of the 40 smallest eigenvalues have a linear dependence on the logarithm of number of nodes (with different coefficients), while, from figures 4.10a and 4.10b, the biggest eigenvalues show some dependence on the number of nodes that is however too noisy to try to be removed by scaling. For this reason one part of the analysis will be brought on by rescaling the smallest eigenvalues with a function of node number. From the set of figures related to 4.15b (biggest eigenvalues-link density) and 4.14b (smallest eigenvalues-link density) for both Laplacian at 8 $\mathring{A}$ and 12 $\mathring{A}$, it can be seen that biggest eigenvalues seem to depend on the link density while smallest eigenvalues don't, (that is, they do in a very noisy way). This would suggest a rescaling with respect to the link density of the biggest eigenvalues for further analysis (which I didn't have the time to test).

4.2 description of distributions of eigenvalues

In this section I am going to take a look on the eigenvalues of the laplacian $\{L\}$ for threshold at 8 $\mathring{A}$ and 12 $\mathring{A}$. Having verified that information are similar as in the case

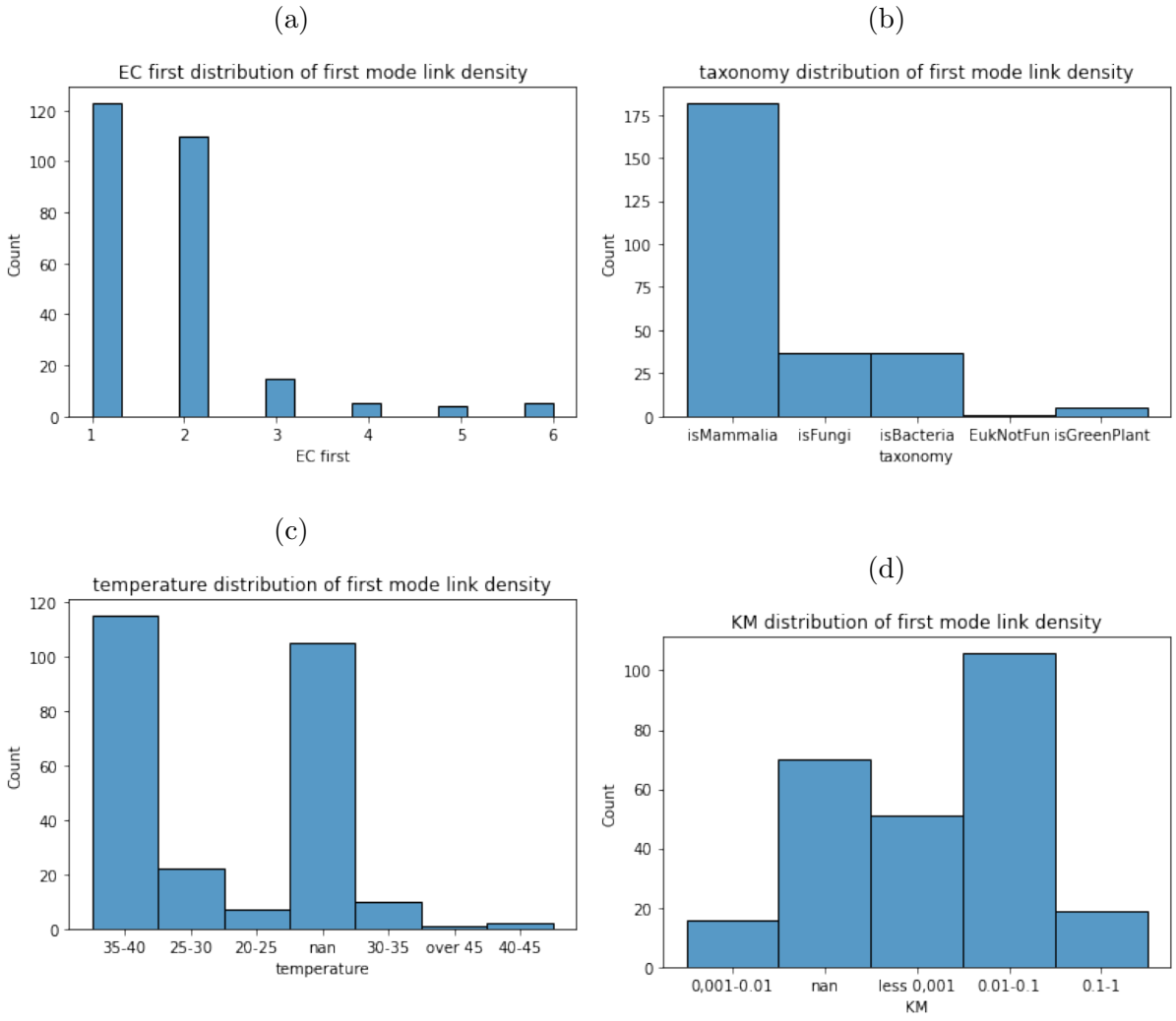


Figure 4.3: In these figure it is represented the composition of the first peak (11.3-11.4) in figure 4.2a are represented in terms of the classes EC first (a), taxonomy (b), temperature (c) and KM (d). It can be seen that a biggest chunks comes from mammalia. Also in this case we find an uniprot code 'P00374' containing 51 PDB'studied in different ways, the EC second '1' everywhere, the KM however vary and the temperature is 35. They form the biggest chunk unified of the peak as in total we have 264 elements. It is difficult to locate this in the t-SNE-UMAP analysis.

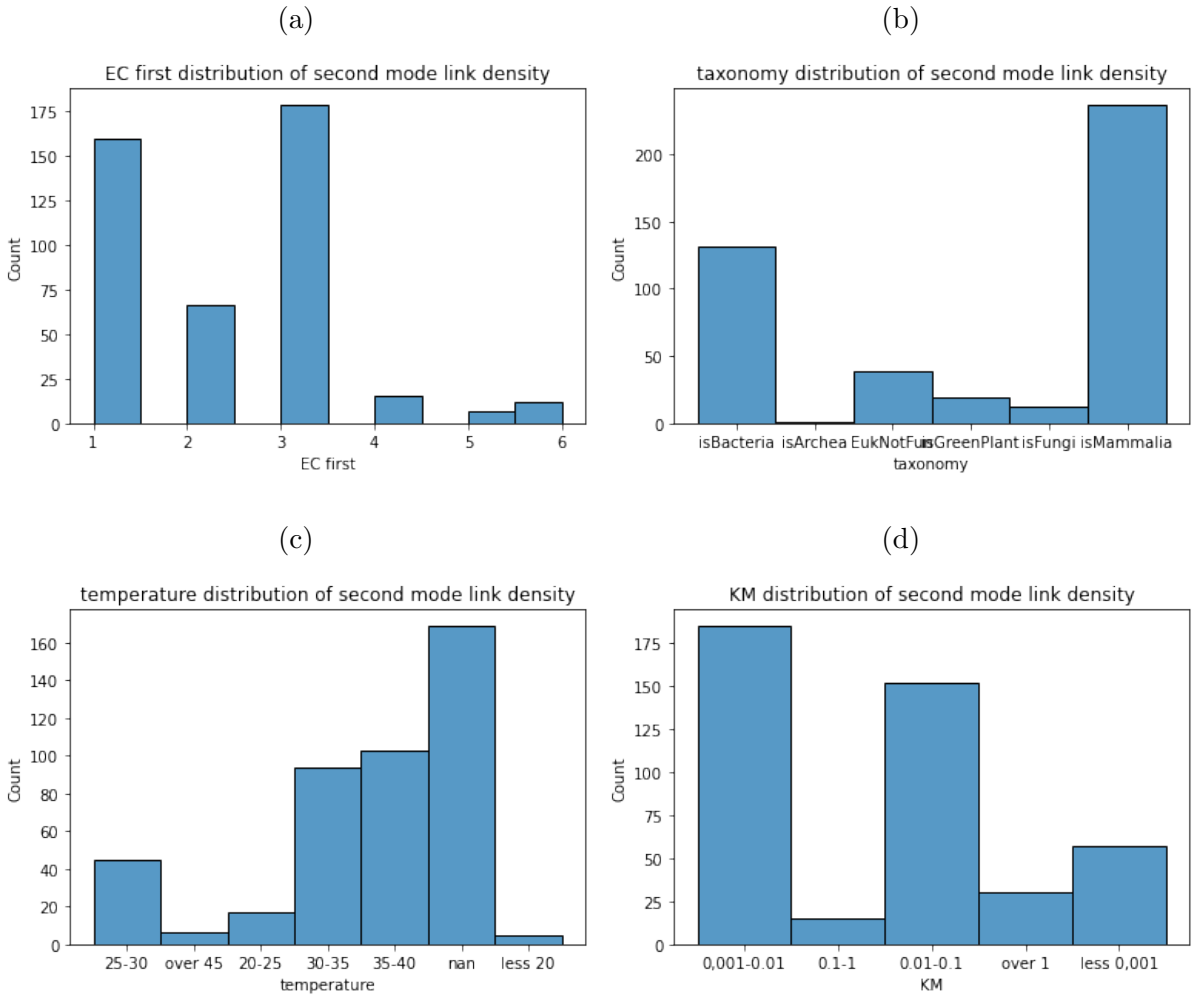


Figure 4.4: In these figure it is represented the composition of the second peak (11.3-11.4) in figure 4.2a are represented in terms of the classes EC first (a),taxonomy (b), temperature (c) and KM (d). It can be seen that a biggest chunks come from mammalia and bacteria. Also in this case we find an uniprot code 'P34913' containing 61 PDB'studied in different ways, the EC second '3'everywhere, the KM however vary and the temperature is 35. They form the biggest chunk unified of the peak as in total we have 439 elements. It is difficult to locate this in the t-SNE-UMAP analysis.

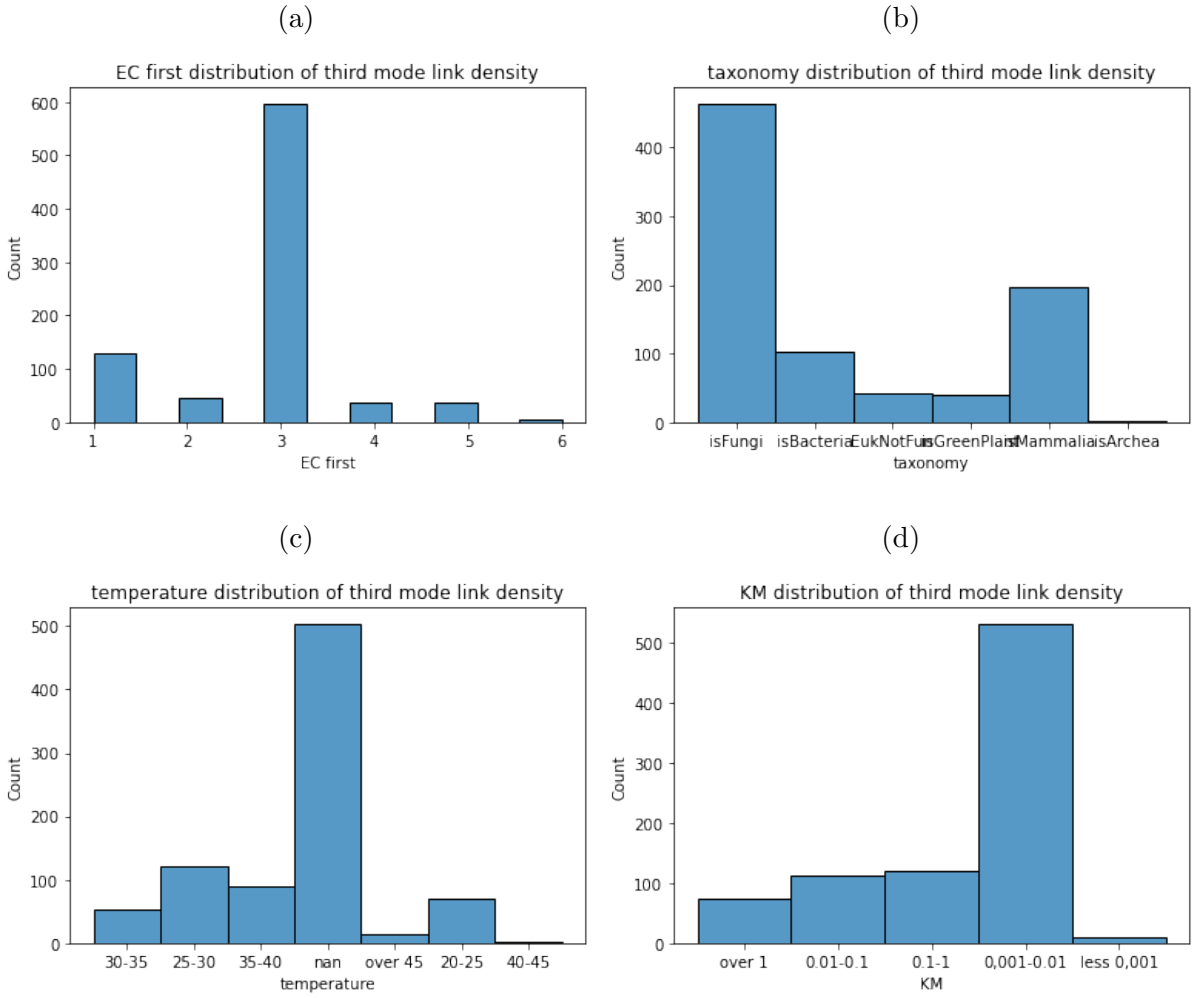


Figure 4.5: In these figure it is represented the composition of the third peak (12.4-12.6) in figure 4.2a are represented in terms of the classes EC first (a),taxonomy (b), temperature (c) and KM (d). It can be seen that a biggest chunks come from mammalia and bacteria. Also in this case we find an uniprot code 'P11838' containing 436 PDB's studied in different ways, the EC second '3'everywhere, the KM however vary and the temperature is 35. This is a part of the chunk coming also from figure 4.8a of 554. They form the biggest chunk unified of the peak as in total we have 864 elements. This chunk is pretty visible in t-SNE-UMAP analysis.

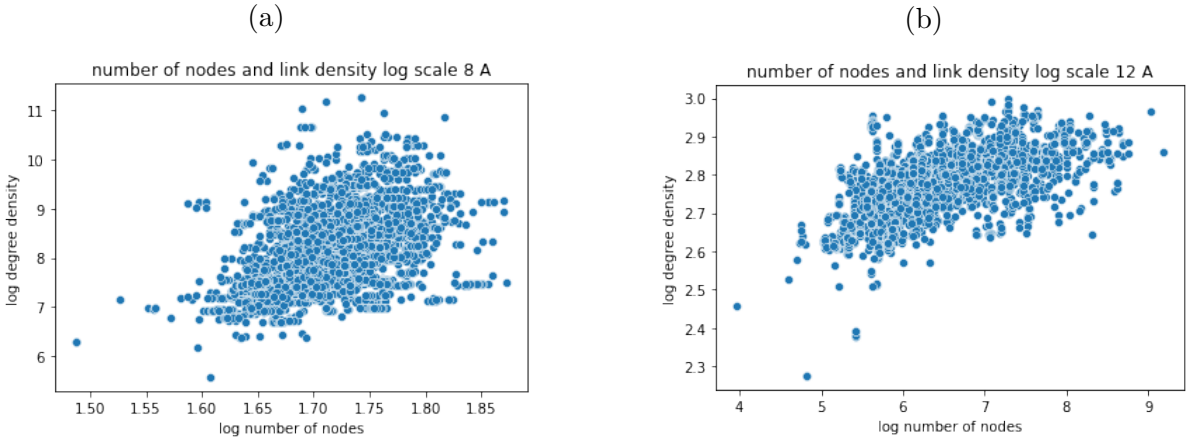


Figure 4.6: In these scatterplots are represented the logarithms of the number of nodes per enzyme against the logarithm of the link density per enzyme for contact maps with threshold of 8 $\mathring{A}$ (a) and 12 $\mathring{A}$ (b). From both the figures there is an hint of dependence on the number of nodes that however is hidden by much noise. In general both the figures seem compatible to the fact suggested in [25], that all the enzymes share similar backbone structure, and more or less the enzymes have the same sparseness of long range contact that seems not to have any structure

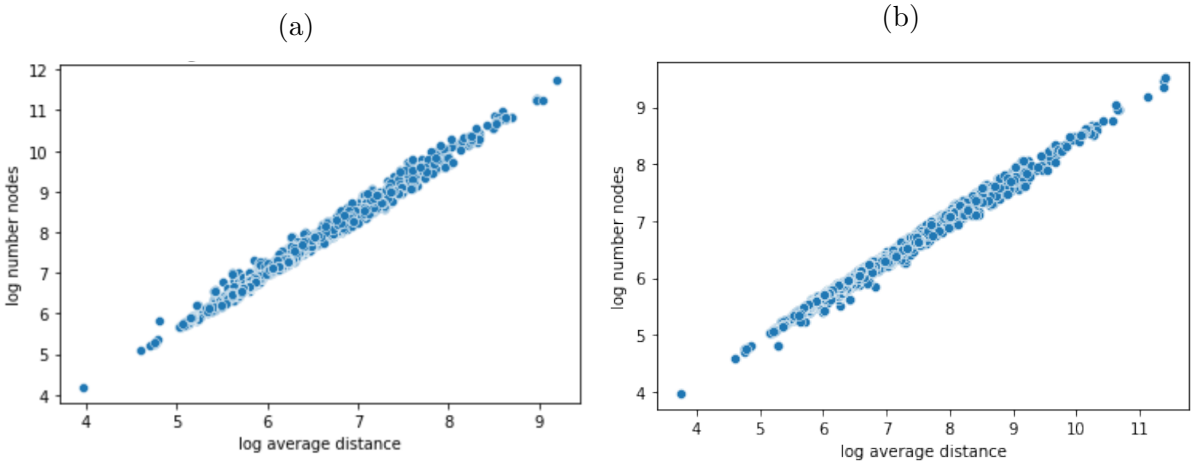


Figure 4.7: In these figures are represented on the x-axis, the average distance among the nodes on each enzyme in log scale and on the y-axis the number of nodes for each enzyme in log scale, for both the cases of threshold 8 $\mathring{A}$ (a) and 12 $\mathring{A}$ (b). The fitting equations are: (a) $\log(\bar{l})$=0.695 log(N) + 1.117 (d) $\log(\bar{l})$=0.708 log(N) + 1.405

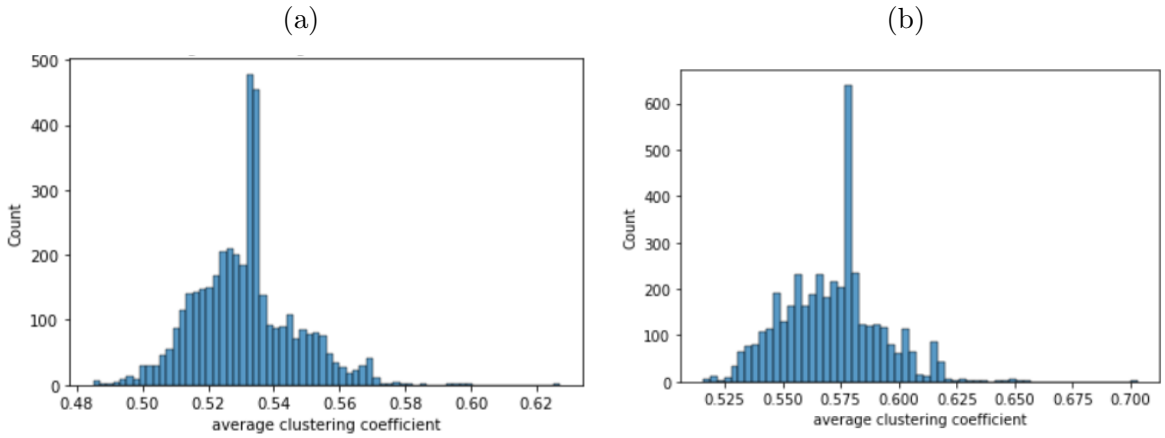


Figure 4.8: In these figures in the x-axis are the average clustering coefficients calculated with the nodes on each enzyme as in equation 3.2 for both the threshold 8 $\mathring{A}$ (a) and 12 $\mathring{A}$ (b). There is a peak of counts in the interval [0.53,0.537] of 1198 enzymes. These 'anomalies' are analyzed in figures 4.9a,4.9b,4.9d,4.9c. It can be seen that the peak is formed principally by those enzymes of 'P11838'

of link density, I decide to show the moments of distribution of the eigenvalues in tabs 4.8 and 4.9 of the laplacian L and normalized laplacian $\mathcal{L}$ for threshold at 8 $\mathring{A}$ only.

It seems from tab.4.8 that all the distributions are close, that is are similar. However one would need to make further analysis to tell that. This analysis will be made in via t-SNE and UMAP that try to represent some aspects (specified better in the following sections and chapters) of distribution distances.

4.3 Effect of rescaling on Laplacian smallest eigenvalues

As already said , in the next section I will show the analysis of just the laplacian obtained from the contact map with threshold 8 $\mathring{A}$. In this section, exploiting the results of 4.1, I try to extract the dependence on the smallest eigenvalues in a process that I call 'rescaling'. Let λ_{i} (i=1,..,40) be one of the smallest eigenvalue then:

$$\lambda_{\mathrm{s,i}} = \lambda_{\mathrm{i}} n^{-\mathrm{a_i}} \tag{4.3}$$

where $\lambda_{\mathrm{s,i}}$ is the eigenvalue rescaled, n the number of nodes and a_{i} the fitting coefficient I have found in tab 4.1. It is in our interest to look if the behavior of the rescaled eigenvalues have some interesting feature, and can spot similarities among enzymes independently of dimension.

4.4 eigenvalues and link density

In this section I am going to evaluate the correlation between the biggest and smallest eigenvalues with respect to the link density. This choice is motivated by the lack of relation among number of nodes and link density, and so it adds a degree of freedom whose dependencies on eigenvalues may be important . It turns out that smallest eigenvalues

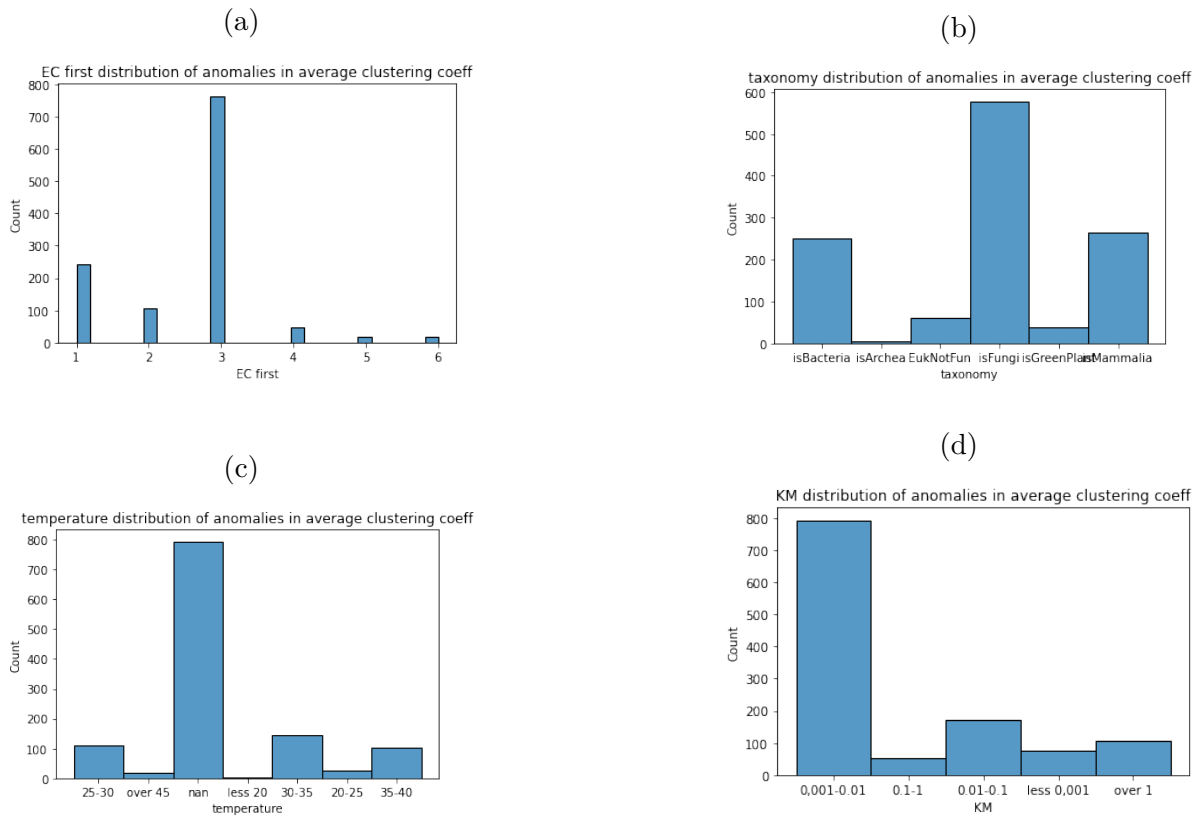


Figure 4.9: In these figure it is represented how the two 'peaks' in figure 4.8a are represented in terms of the classes EC first (a), taxonomy (b), temperature (c) and KM (d). It can be seen that a big chunk comes from fungi. That set is composed by 554 elements whose uniprot code is P11838, it is endothiapepsin crystal studied in different ways, the EC first '3', the KM are all 0.0016 and the temperature is not defined. These 554 graphs, have 330 nodes. They are very similar one another and it can be considered as a redundance. This group of enzymes is going to be recognized also later in the t-SNE-UMAP analysis.

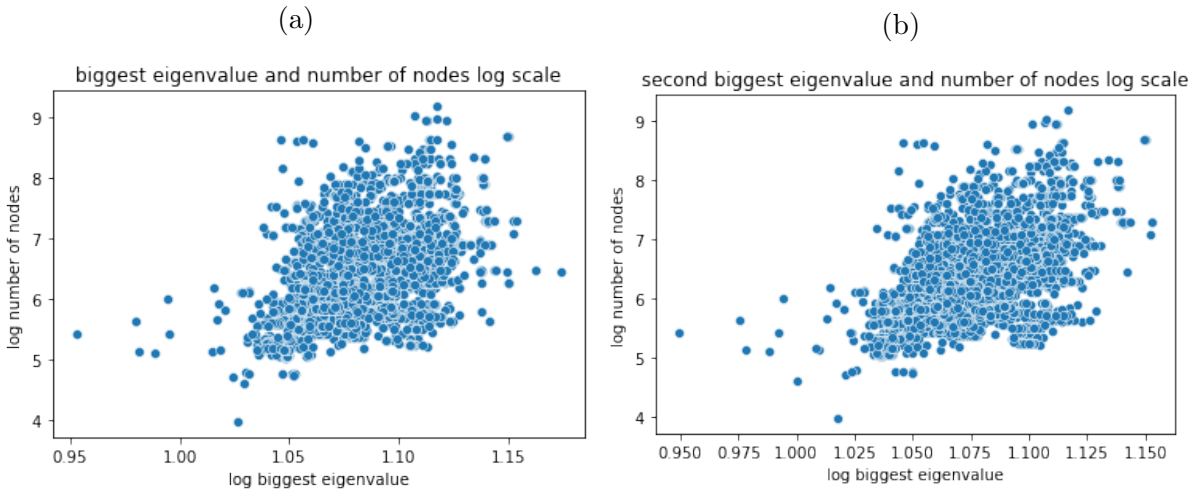


Figure 4.10: These figures represent the logarithm of the biggest (a) and the second biggest eigenvalue (b) with respect to the logarithm of the number of nodes for laplacian (L) obtained with a threshold of 8 $\mathring{A}$. Even though a dependence on the number of nodes is suggested for both the eigenvalues, it is very noisy, thus rescaling by node number might be not much effective. This behavior is similar up to the 40-th biggest eigenvalue.

seem not to depend on the link density, while linear dependence is visible for the biggest. I won't show the coefficients of dependence as I didn't use them for rescaling. The $\mathbf{R}^2$ ranges from 0.6 to 0.9.

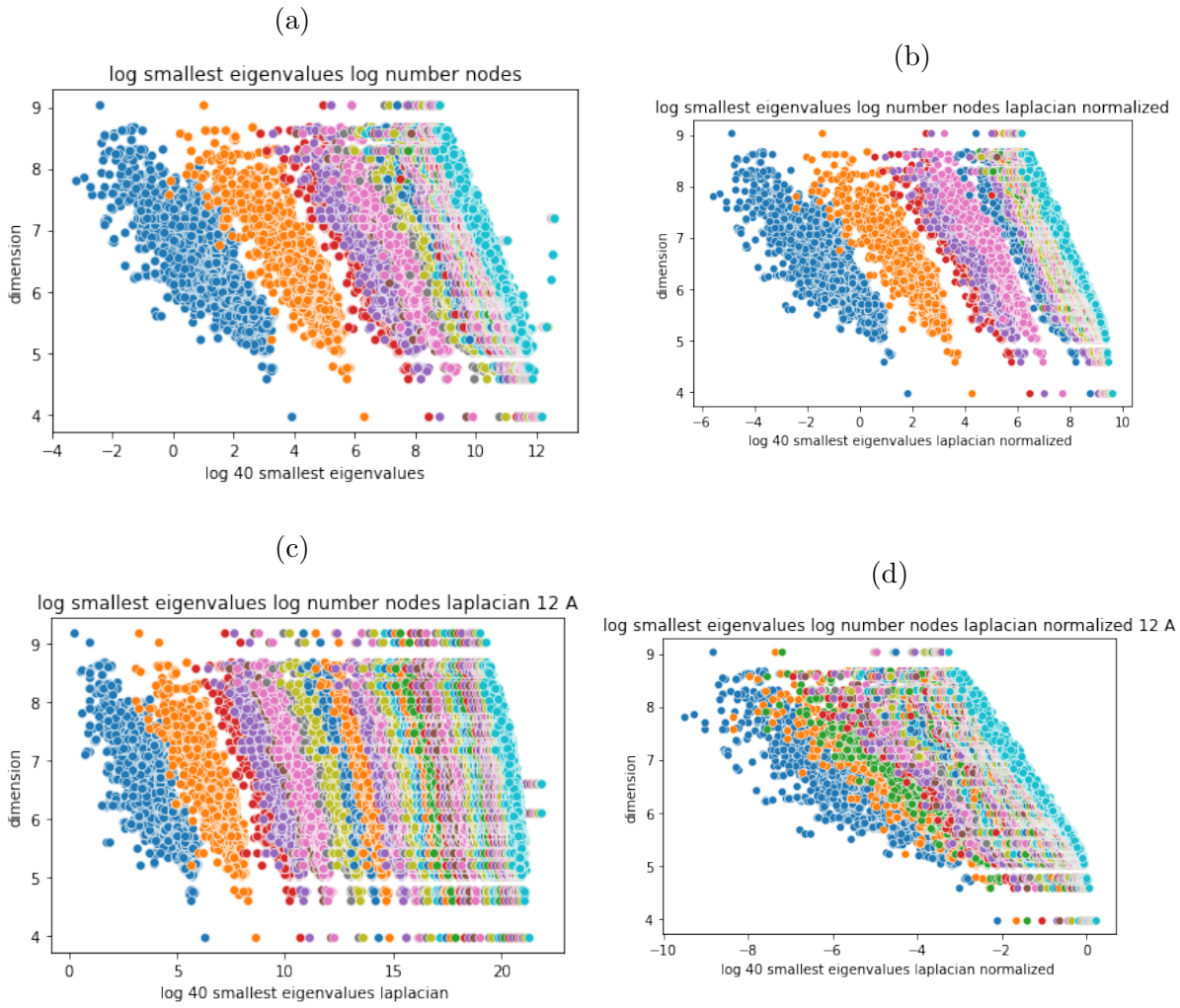


Figure 4.11: Logarithm of the 40 smallest eigenvalues of the laplacian (L) in the configurations from 8 $\mathring{A}$ and 12 $\mathring{A}$ in (a),(b), and of the the 40 smallest eigenvalues of the normalized laplacian ($\mathcal{L}$) in the configurations from 8$\mathring{A}$ and 12 $\mathring{A}$ in (c),(d) respectively, as a function of protein size (node number).

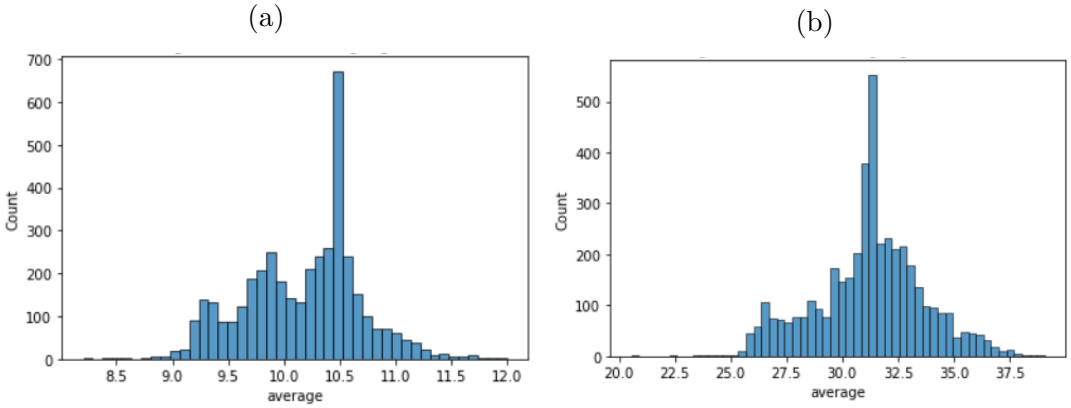


Figure 4.12: These two figures represent histplots containing the means (μ_{L}) of the distribution of the eigenvalues of the laplacian L for each enzyme whose contact map are obtained with threshold 8 $\mathring{A}$ (a) and threshold 12 $\mathring{A}$ (b). They are very similar to figure 4.2a and 4.2b. It is infact the case that the trace of the diagonalized laplacian (sum of eigenvalues) is = to the trace of the non diagonalized (sum of degrees). The coarse graining of the structure,again, corresponds to a smear of the distribution.

Table 4.1

$log(\lambda_i) = alog(n) + b$	a	b	$\boldsymbol{R^2}$
$log(\lambda_1)$	-0.48	6.94	0.34
$log(\lambda_2)$	-0.71	9.36	0.62
$log(\lambda_3)$	-0.77	10.33	0.11
$log(\lambda_4)$	-0.91	12.29	0.72
$log(\lambda_5)$	-0.94	12.86	0.79
$log(\lambda_6)$	-1.02	14.31	0.82
$log(\lambda_7)$	-1.01	14.35	0.84
$log(\lambda_8)$	-1.10	16.02	0.85
$log(\lambda_9)$	-1.12	16.43	0.87
$log(\lambda_{10})$	-1.15	17.12	0.89
$log(\lambda_{11})$	-1.15	17.14	0.90
$log(\lambda_{12})$	-1.17	17.76	0.91
$log(\lambda_{13})$	-1.17	17.73	0.91
$log(\lambda_{14})$	-1.14	17.23	0.91
$log(\lambda_{15})$	-1.14	17.32	0.92
$log(\lambda_{16})$	-1.15	17.59	0.92
$log(\lambda_{17})$	-1.15	17.66	0.92
$log(\lambda_{18})$	-1.16	18.00	0.93
$log(\lambda_{19})$	-1.17	18.07	0.94
$log(\lambda_{20})$	-1.17	18.24	0.94
$log(\lambda_{21})$	-1.18	18.37	0.94
$log(\lambda_{22})$	-1.17	18.42	0.94
$log(\lambda_{23})$	-1.17	18.4	0.95
$log(\lambda_{24})$	-1.17	18.49	0.95
$log(\lambda_{25})$	-1.17	18.46	0.95
$log(\lambda_{26})$	-1.16	18.38	0.95
$log(\lambda_{27})$	-1.16	18.33	0.95
$log(\lambda_{28})$	-1.17	18.68	0.95
$log(\lambda_{29})$	-1.17	18.7	0.95
$log(\lambda_{30})$	-1.18	18.85	0.96
$log(\lambda_{31})$	-1.18	18.81	0.96
$log(\lambda_{32})$	-1.16	18.57	0.95
$log(\lambda_{33})$	-1.16	18.5	0.95
$log(\lambda_{34})$	-1.16	18.53	0.95
$log(\lambda_{35})$	-1.16	18.57	0.95
$log(\lambda_{36})$	-1.16	18.65	0.95
$log(\lambda_{37})$	-1.16	18.64	0.95
$log(\lambda_{38})$	-1.16	18.78	0.95
$log(\lambda_{39})$	-1.17	18.87	0.95
$log(\lambda_{40})$	-1.17	18.88	0.95

Table 4.2

$log(\lambda_i) = alog(n) + b$	a	b	$\boldsymbol{R^2}$
$log(\lambda_1)$	-0.54	6.13	0.37
$log(\lambda_2)$	-0.77	6.73	0.59
$log(\lambda_3)$	-0.83	7.06	0.67
$log(\lambda_4)$	-0.99	7.51	0.67
$log(\lambda_5)$	-1.03	7.72	0.74
$log(\lambda_6)$	-1.14	8.04	0.75
$log(\lambda_7)$	-1.16	8.18	0.77
$log(\lambda_8)$	-1.28	8.53	0.79
$log(\lambda_9)$	-1.31	8.69	0.8
$log(\lambda_{10})$	-1.36	8.88	0.8
$log(\lambda_{11})$	-1.38	9.01	0.81
$log(\lambda_{12})$	-1.45	9.23	0.8
$log(\lambda_{13})$	-1.47	9.34	0.8
$log(\lambda_{14})$	-1.53	9.53	0.81
$log(\lambda_{15})$	-1.53	9.62	0.81
$log(\lambda_{16})$	-1.59	9.82	0.81
$log(\lambda_{17})$	-1.6	9.88	0.8
$log(\lambda_{18})$	-1.64	10.03	0.8
$log(\lambda_{19})$	-1.65	10.12	0.8
$log(\lambda_{20})$	-1.7	10.29	0.8
$log(\lambda_{21})$	-1.73	10.39	0.8
$log(\lambda_{22})$	-1.75	10.48	0.8
$log(\lambda_{23})$	-1.78	10.58	0.8
$log(\lambda_{24})$	-1.83	10.74	0.8
$log(\lambda_{25})$	-1.85	10.83	0.79
$log(\lambda_{26})$	-1.87	10.93	0.79
$log(\lambda_{27})$	-1.91	11.05	0.79
$log(\lambda_{28})$	-1.96	11.2	0.79
$log(\lambda_{29})$	-1.97	11.28	0.79
$log(\lambda_{30})$	-2.0	11.39	0.79
$log(\lambda_{31})$	-2.03	11.48	0.79
$log(\lambda_{32})$	-2.05	11.56	0.79
$log(\lambda_{33})$	-2.04	11.57	0.77
$log(\lambda_{34})$	-2.06	11.65	0.77
$log(\lambda_{35})$	-2.08	11.72	0.77
$log(\lambda_{36})$	-2.1	11.81	0.77
$log(\lambda_{37})$	-2.12	11.87	0.77
$log(\lambda_{38})$	-2.13	11.94	0.77
$log(\lambda_{39})$	-2.15	12.0	0.77
$log(\lambda_{40})$	-2.16	12.06	0.77

Table 4.3: In these tabs are represented the fitting parameters between the logarithms of the 40 smallest eigenvalues of the laplacian and the logarithm of the number of nodes, for the operators whose contact map has been calculated with the threshold of 8 $\mathring{A}$ (a) and 12 $\mathring{A}$ (b)

Table 4.4: In these tabs are represented the fitting parameters between the logarithms of the 40 smallest eigenvalues of the normalized laplacian and the logarithm of the number of nodes, for the operators whose contact map has been calculated with the threshold of 8 $\mathring{A}$ (a) and 12 $\mathring{A}$ (b)

Table 4.5

$log(\lambda_i) = alog(n) + b$	a	b	$\boldsymbol{R}^2$
$log(\lambda_1)$	-0.47	4.0	0.43
$log(\lambda_2)$	-0.71	3.54	0.71
$log(\lambda_3)$	-0.76	3.66	0.79
$log(\lambda_4)$	-0.91	3.48	0.83
$log(\lambda_5)$	-0.93	3.53	0.87
$log(\lambda_6)$	-1.0	3.51	0.89
$log(\lambda_7)$	-0.99	3.68	0.9
$log(\lambda_8)$	-1.07	3.62	0.91
$log(\lambda_9)$	-1.08	3.69	0.92
$log(\lambda_{10})$	-1.08	3.82	0.93
$log(\lambda_{11})$	-1.07	3.92	0.94
$log(\lambda_{12})$	-1.12	3.9	0.95
$log(\lambda_{13})$	-1.11	4.0	0.95
$log(\lambda_{14})$	-1.11	4.09	0.96
$log(\lambda_{15})$	-1.09	4.19	0.96
$log(\lambda_{16})$	-1.12	4.21	0.96
$log(\lambda_{17})$	-1.11	4.27	0.96
$log(\lambda_{18})$	-1.12	4.32	0.97
$log(\lambda_{19})$	-1.11	4.38	0.97
$log(\lambda_{20})$	-1.13	4.41	0.97
$log(\lambda_{21})$	-1.13	4.46	0.97
$log(\lambda_{22})$	-1.12	4.53	0.98
$log(\lambda_{23})$	-1.12	4.57	0.98
$log(\lambda_{24})$	-1.12	4.62	0.98
$log(\lambda_{25})$	-1.12	4.66	0.98
$log(\lambda_{26})$	-1.11	4.72	0.98
$log(\lambda_{27})$	-1.11	4.75	0.98
$log(\lambda_{28})$	-1.12	4.78	0.98
$log(\lambda_{29})$	-1.12	4.81	0.98
$log(\lambda_{30})$	-1.13	4.84	0.98
$log(\lambda_{31})$	-1.12	4.88	0.98
$log(\lambda_{32})$	-1.12	4.92	0.98
$log(\lambda_{33})$	-1.12	4.95	0.98
$log(\lambda_{34})$	-1.12	4.97	0.98
$log(\lambda_{35})$	-1.12	5.0	0.99
$log(\lambda_{36})$	-1.13	5.02	0.99
$log(\lambda_{37})$	-1.13	5.05	0.99
$log(\lambda_{38})$	-1.13	5.07	0.99
$log(\lambda_{39})$	-1.13	5.1	0.99
$log(\lambda_{40})$	-1.13	5.13	0.99

Table 4.6

$log(\lambda_i) = alog(n) + b$	a	b	$\boldsymbol{R}^2$
$log(\lambda_1)$	-0.54	4.35	0.53
$log(\lambda_2)$	-0.77	4.22	0.77
$log(\lambda_3)$	-0.81	4.4	0.84
$log(\lambda_4)$	-0.98	4.35	0.88
$log(\lambda_5)$	-0.98	4.49	0.89
$log(\lambda_6)$	-1.07	4.53	0.91
$log(\lambda_7)$	-1.08	4.63	0.92
$log(\lambda_8)$	-1.16	4.66	0.94
$log(\lambda_9)$	-1.17	4.76	0.94
$log(\lambda_{10})$	-1.19	4.85	0.95
$log(\lambda_{11})$	-1.21	4.9	0.95
$log(\lambda_{12})$	-1.28	4.92	0.96
$log(\lambda_{13})$	-1.27	5.0	0.96
$log(\lambda_{14})$	-1.31	5.04	0.96
$log(\lambda_{15})$	-1.31	5.09	0.96
$log(\lambda_{16})$	-1.36	5.12	0.97
$log(\lambda_{17})$	-1.37	5.16	0.96
$log(\lambda_{18})$	-1.41	5.18	0.97
$log(\lambda_{19})$	-1.41	5.23	0.96
$log(\lambda_{20})$	-1.44	5.26	0.96
$log(\lambda_{21})$	-1.45	5.3	0.96
$log(\lambda_{22})$	-1.47	5.33	0.96
$log(\lambda_{23})$	-1.49	5.36	0.96
$log(\lambda_{24})$	-1.53	5.37	0.96
$log(\lambda_{25})$	-1.54	5.4	0.96
$log(\lambda_{26})$	-1.56	5.42	0.96
$log(\lambda_{27})$	-1.58	5.45	0.96
$log(\lambda_{28})$	-1.61	5.47	0.95
$log(\lambda_{29})$	-1.62	5.49	0.95
$log(\lambda_{30})$	-1.65	5.51	0.95
$log(\lambda_{31})$	-1.66	5.53	0.94
$log(\lambda_{32})$	-1.69	5.54	0.94
$log(\lambda_{33})$	-1.71	5.56	0.94
$log(\lambda_{34})$	-1.74	5.57	0.93
$log(\lambda_{35})$	-1.75	5.59	0.93
$log(\lambda_{36})$	-1.79	5.6	0.93
$log(\lambda_{37})$	-1.8	5.61	0.92
$log(\lambda_{38})$	-1.82	5.62	0.92
$log(\lambda_{39})$	-1.84	5.64	0.91
$log(\lambda_{40})$	-1.87	5.65	0.9

Table 4.7: In these tab I have represented the average,variance, max value and min value of the first four principal moments $(\mu, \sigma, \gamma, kurt)$ for all the eigenvalues of each Laplacian L (4.2) average and variance, max value and min value of the first four principal moments $(\sigma, \gamma, kurt)$ for all the eigenvalues of each normalized laplacian $\mathcal{L}$ (4.3).

Table 4.8

	mean	variance	max	min
μ_{L}	10.173342	0.520610	8.201439	12.000000
σ_{L}	18.230391	1.568598	11.482742	26.352493
γ_{L}	-0.332704	0.081392	-0.683385	-0.050474
$\mathrm{kurt}_{\mathrm{L}}$	-0.446208	0.098481	-0.816198	-0.069437

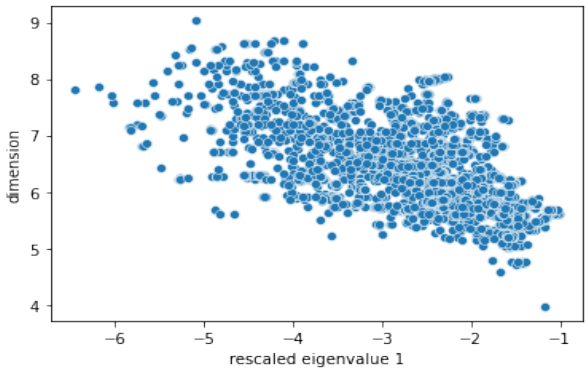
Table 4.9

	mean	variance	max	min
$\mu_{\{\mathcal{L}\}}$	1	0	1	1
$\sigma_{\{\mathcal{L}\}}$	0.10230	0.3 e-04	0.08713	0.12934
$\gamma_{\{\mathcal{L}\}}$	-1.373970	0.001561	-1.522011	-1.212919
$\mathrm{kurt}_{\{\mathcal{L}\}}$	1.257559	0.019161	0.804302	1.764981

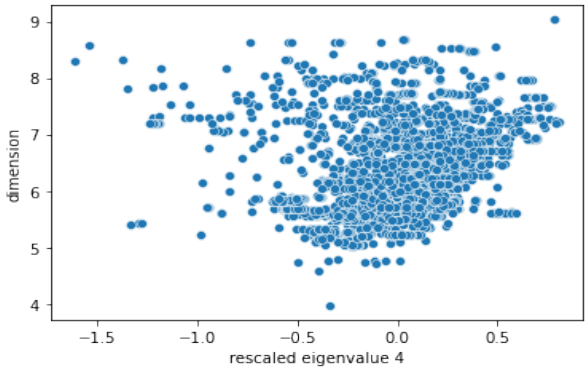
Table 4.10: In this tab I represent the fit of the rescaled eigenvalues. It can be seen that in all the cases $\mathbf{R}^2$ has decreased, however the dependence is still there and can be seen from fig. 4.13a to fig 4.13d

$\log(\lambda_{\mathrm{s,i}}) = a log(n_{\mathrm{i}}) + b$	a	b	$\mathbf{R}^2$
$log(\lambda_{\mathrm{s},1})$	-0.47	5.07	-0.7
$log(\lambda_{\mathrm{s},2})$	-0.62	5.91	-3.3
$log(\lambda_{\mathrm{s},3})$	0.21	6.25	-105.9
$log(\lambda_{\mathrm{s},4})$	0.33	6.24	-52.86
$log(\lambda_{\mathrm{s},5})$	1.17	6.29	-2.58
$log(\lambda_{\mathrm{s},6})$	1.15	6.28	-3.37
$log(\lambda_{\mathrm{s},7})$	1.65	6.69	0.05
$log(\lambda_{\mathrm{s},8})$	1.73	6.84	0.24
$log(\lambda_{\mathrm{s},9})$	1.8	7.12	0.5
$log(\lambda_{\mathrm{s},10})$	1.87	7.15	0.5
$log(\lambda_{\mathrm{s},11})$	1.85	7.39	0.63
$log(\lambda_{\mathrm{s},12})$	1.92	7.41	0.63
$log(\lambda_{\mathrm{s},13})$	2.0	7.19	0.47
$log(\lambda_{\mathrm{s},14})$	2.06	7.24	0.5
$log(\lambda_{\mathrm{s},15})$	2.01	7.34	0.55
$log(\lambda_{\mathrm{s},16})$	2.05	7.38	0.57
$log(\lambda_{\mathrm{s},17})$	2.06	7.55	0.66
$log(\lambda_{\mathrm{s},18})$	2.11	7.61	0.68
$log(\lambda_{\mathrm{s},19})$	2.1	7.69	0.7
$log(\lambda_{\mathrm{s},20})$	2.11	7.75	0.72
$log(\lambda_{\mathrm{s},21})$	2.14	7.79	0.73
$log(\lambda_{\mathrm{s},22})$	2.17	7.79	0.73
$log(\lambda_{\mathrm{s},23})$	2.18	7.84	0.74
$log(\lambda_{\mathrm{s},24})$	2.23	7.84	0.74
$log(\lambda_{\mathrm{s},25})$	2.25	7.79	0.72
$log(\lambda_{\mathrm{s},26})$	2.28	7.77	0.7
$log(\lambda_{\mathrm{s},27})$	2.22	7.93	0.76
$log(\lambda_{\mathrm{s},28})$	2.26	7.96	0.77
$log(\lambda_{\mathrm{s},29})$	2.24	8.04	0.79
$log(\lambda_{\mathrm{s},30})$	2.27	8.03	0.78
$log(\lambda_{\mathrm{s},31})$	2.27	7.86	0.72
$log(\lambda_{\mathrm{s},32})$	2.31	7.83	0.7
$log(\lambda_{\mathrm{s},33})$	2.31	7.84	0.7
$log(\lambda_{\mathrm{s},34})$	2.32	7.86	0.71
$log(\lambda_{\mathrm{s},35})$	2.31	7.89	0.72
$log(\lambda_{\mathrm{s},36})$	2.32	7.88	0.71
$log(\lambda_{\mathrm{s},37})$	2.3	7.95	0.73
$log(\lambda_{\mathrm{s},38})$	2.28	8.0	0.74
$log(\lambda_{\mathrm{s},39})$	2.22	7.94	0.72

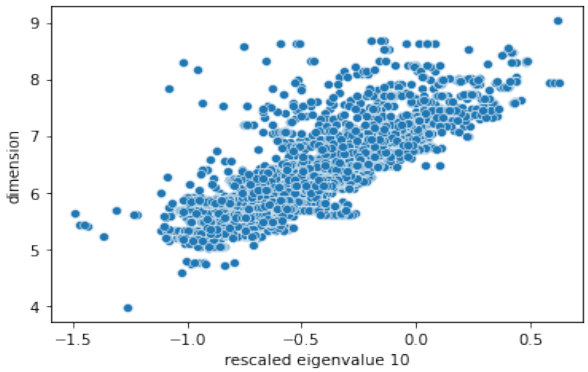
(a)



(b)



(c)



(d)

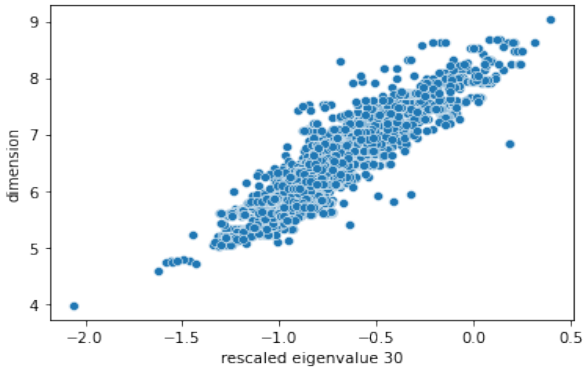


Figure 4.13: In these figures are represented the 'rescaled' eigenvlaues 1 (a), 4 (b), 10 (c), 30 (d). It can be seen that the dependence is always there. However on the smallest eigenvalues it is less evident.

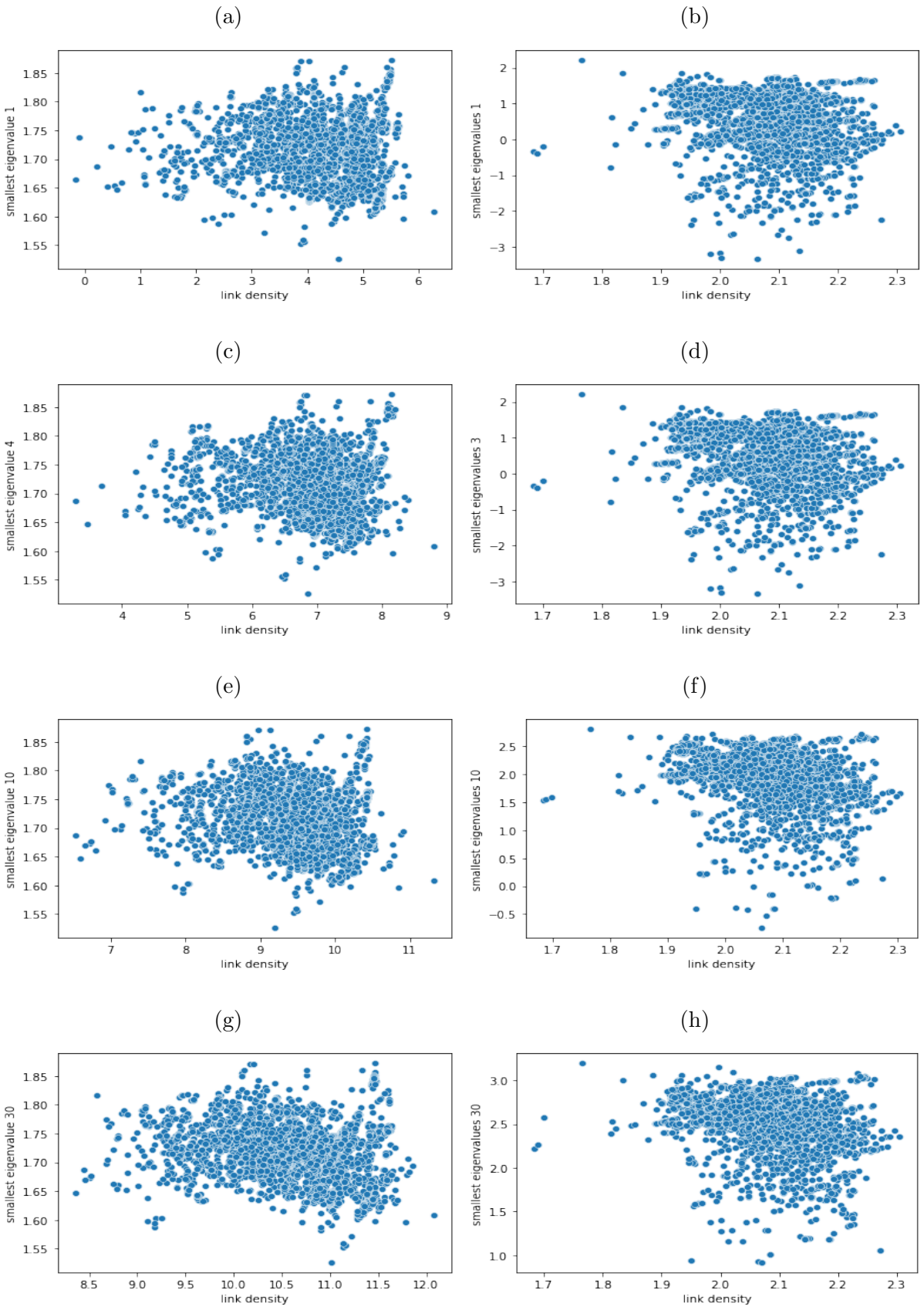


Figure 4.14: In these figures I represent the logarithms of λ_1 (a,b), λ_4 (c,d), λ_{10} (e,f), λ_{30} (g,h) with respect to the logarithm of link density in the laplacian with threshold 8 $\mathring{A}$ (on the left) and 12 $\mathring{A}$ (on the right). I have chosen these 4 to show that the there seems to be no clear dependence.

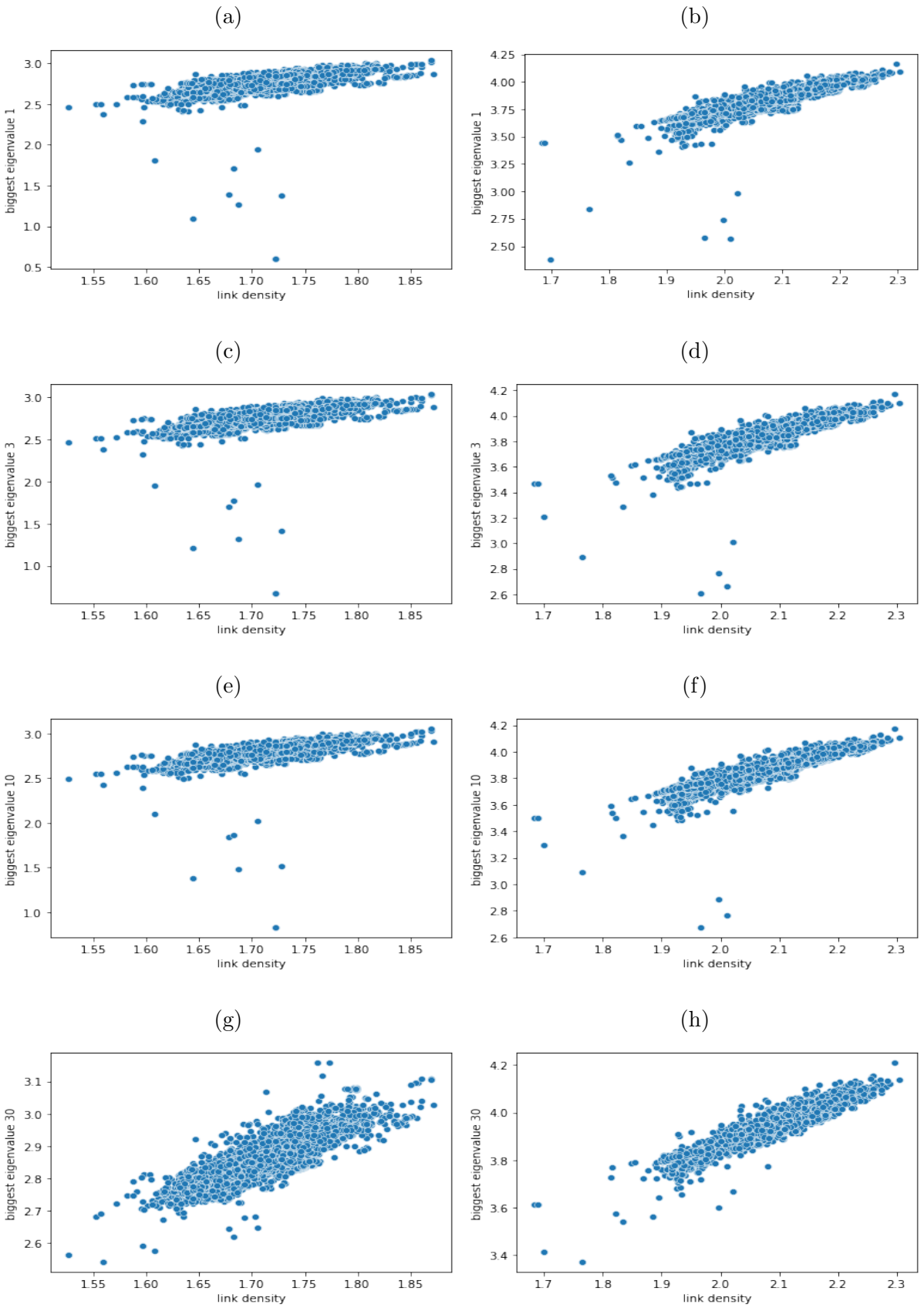


Figure 4.15: In these figures I represent the logarithm of $\lambda_{\text{n-1}}$ (a,b), $\lambda_{\text{n-4}}$ (c,d), $\lambda_{\text{n-10}}$ (e,f), $\lambda_{\text{n-30}}$ (g,h) (biggest eigenvalues) with respect to the logarithm of link density in the laplacian with threshold 8 $\mathring{A}$ (on the left) and 12 $\mathring{A}$ (on the right). We start to see a dependence that becomes more marked going up to 30-th eigenvalue.

Chapter 5

Results

In this section I will show the results of the analysis I have pursued using the methods introduced so far. I will show the results for the laplacian extracted from contact maps whose threshold is 8 $\mathring{A}$. I will, besides, show the results of the analysis with: 40 biggest eigenvalues, 40 smallest eigenvalues and 40 smallest eigenvalues rescaled by node number (see Table 4.1). In the Appendix, the other cases are shown, and they are not significantly different from that of the laplacian at 8 $\mathring{A}$. T he principal results that I describe in sections **PCA analysis**,**t-SNE analysis** and **UMAP analysis** are:

- PCA embedding is not able to spot differences related to EC first, taxonomy, temperature and KM labels. Biggest eigenvalues show dependence on the number of nodes and also the smallest. The rescaled smallest eigenvalues still show significant dependence on the number of nodes
- t-SNE creates many clusters. These are not able to recognize EC first, taxonomy, temperature and KM labels. However, clusters are related to the uniprot codes, and so that different PDB structures belonging to the same uniprot sequence are very close in this space. Furthermore, the number of nodes still appears as an identifiable feature
- UMAP create little less clusters than t-SNE. These clusters, again, are not able to recognize EC first, taxonomy, temperature and KM labels. However, similar to t-SNE, clusters are related to the uniprot codes. I have set the parameter 'number of neighbors' to 5 for the biggest and the smallest eigenvalue and to 10 for the rescaled smallest eigenvalues, since it provided the best visual structure on the data.

5.1 Summary of analysis procedure and goals

So far I have calculated the contact maps for each enzyme setting the distance threshold to both 8 and 12 $\mathring{A}$ and seeing that there is no much difference among them. From these contact maps I have extracted the laplacian and normalized laplacian. I have seen that both from the analysis on the eigenvalues, link density and clustering coefficient, and PCA,t-SNE and UMAP, the laplacian obtained from contact maps of 8 $\mathring{A}$ gives enough information. For this reason I have decided just to show it in my results, and postpone to the Appendix the analysis of both the normalized laplacian (8 $\mathring{A}$) and the laplacian and normalized laplacian (12 $\mathring{A}$).

The goal of this work, is to see whether Laplacian observables (eigenvalues) are able to spot differences among different enzymes independently on their dimension. The differences between enzymes are represented by different labels such as EC, KM, temperature and taxonomy. The principal problem here is that the dataset (figure 2.10a,2.8a,2.6a,2.2a)) is composed by enzymes of different size (thus networks with a different number of nodes). Thus I tried a way to construct a semi-metric space from the spectrum datathrough UMAP,t-SNE and PCA, independently on network size. The observable space is composed by the 40 biggest and the 40 smallest eigenvalues of the Laplacian operator (L). As it is shown by Table 4.2 and **??**, for the smallest eigenvalues, there is a polynomial dependence with node number. This means that in line of principle, it could be possible to remove from these eigenvalues the dependence on the number of nodes, but it turns out this is not satisfactorily possible. For the biggest eigenvalues this was not possible, as no neat relation with node number has been found. So, here, we will look if PCA,t-SNE and UMAP, applied to these sets in 40 dimensional spaces, are able to spot differences in taxonomy, temperature, KM and EC indexes for L. I will try this analysis for 'rescaled' smallest eigenvalues.

As no previous knowledge on enzyme similarity is known, the parameters of t-SNE and UMAP were chosen in order to find maximal similarity (i.e. closeness) among the embedding of the different enzyme structures related to the same protein sequence 'P11838'.

5.2 PCA analysis

5.2.1 40 biggest and smallest eigenvalues laplacian

In this section, Figures 5.1a, 5.1b, 5.1c, 5.1d show the PCA analysis for 40 biggest eigenvalues of the laplacian L. Figures 5.2a, 5.2b, 5.3a, 5.3b show the PCA analysis for 40 smallest eigenvalues of the laplacian L and Figures 5.4a, 5.4b, 5.5a, 5.5b show the PCA analysis for 40 smallest eigenvalues rescaled by node number. In all the PCA plots, there is not a clear separation of the different labels, even though the amount of information preserved in the dimensionality reduction (PCA explained variance) is high, close to 1. This suggests us that features represented by taxonomy, EC first, KM and temperature might be non linearly related to the eigenvalue space of L, thus cannot be caught by the PCA analysis (producing linear projections onto this space). We then moved to algorithms that try to reconstruct more complex manifolds (i.e. not simple linear projections of the original space). We applied t-SNE and UMAP: these two algorithms generate non linear trasforms of initial data space by minimizing some cost function for optimal representation of data as explained in [6].

5.3 t-SNE analysis

In this section I apply t-SNE analysis to the 40 biggest, smallest and rescaled smallest eigenvalues of the laplacian coming from contact maps with threshold 8 $\mathring{A}$. I have tested different parameter values: the perplexity and the number of nearest neighbors (data not shown). Finally, I have set the parameters as follows: perplexity=30 . Figures (5.8a, 5.8b, 5.8c, 5.8d) show the t-SNE analysis for 40 biggest eigenvalues of the laplacian L, (5.9a, 5.9b, 5.9c, 5.9d) show the t-SNE analysis for 40 smallest eigenvalues of the

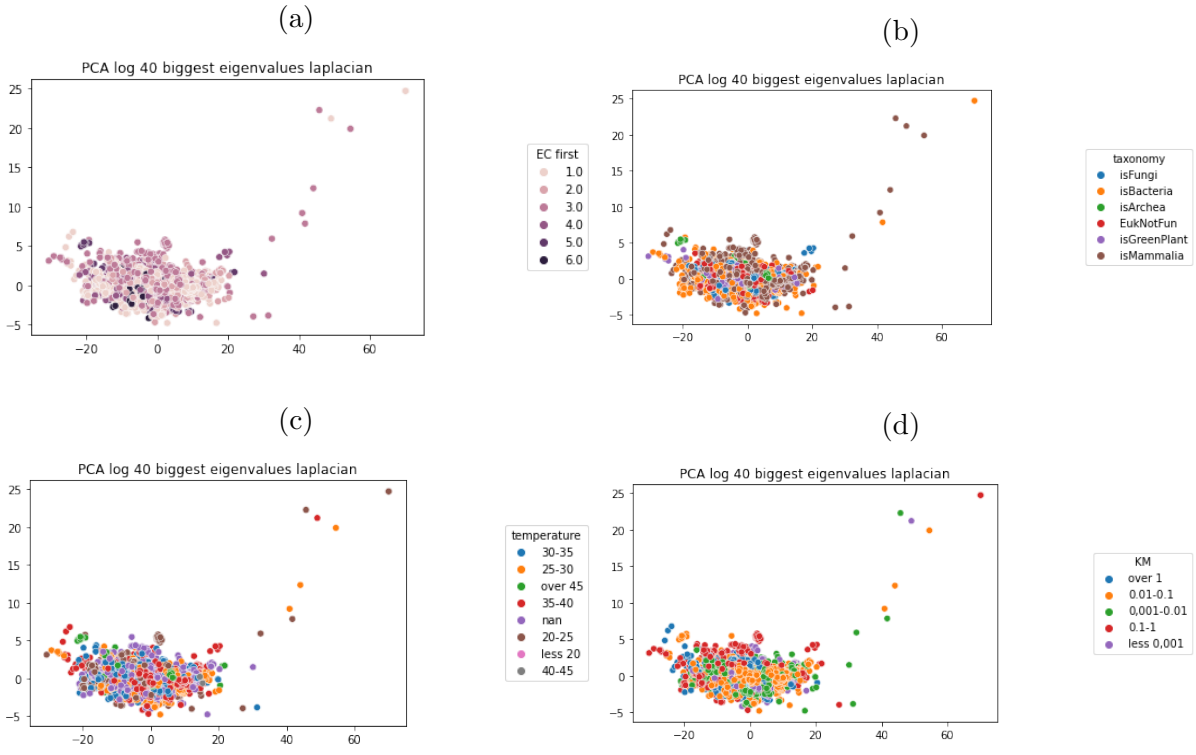
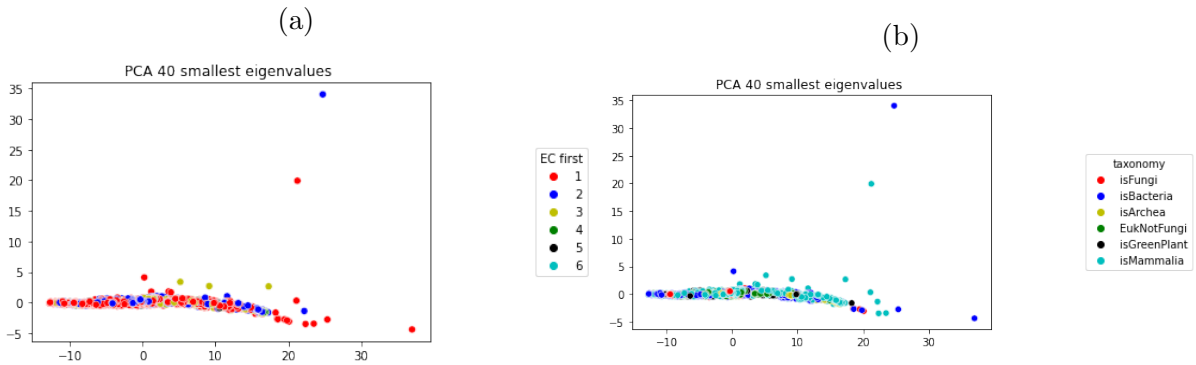


Figure 5.1: Are here represented the scatterplots of the PCA of the 40 biggest eigenvalues of the laplacian (L) obtained with the threshold set to 8 $\mathring{A}$. Figure (a) is colored by EC first label, figure (b) by taxonomy, figure (c) by temperature and in the end, figure (d) by KM. Apparently there is no possibility of conducting discriminant classification of the different lables. All the classes are infact spread along the figure and mixed without any sharp distinction among them. The explained variance is [0.95, 0.04]



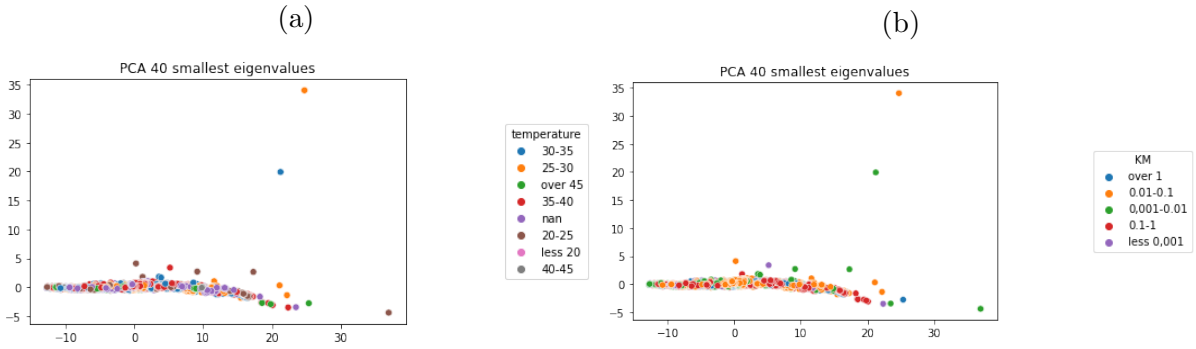


Figure 5.3: Are here represented the scatterplots of the PCA of the 40 smallest eigenvalues of the laplacian (L) obtained with the threshold set to 8 $\mathring{A}$. Figure (a) is colored by EC first label, figure (b) by taxonomy, figure (c) by temperature and in the end, figure (d) by KM. Apparently there is no possibility of conducting discriminant classification of the different lables. All the classes are infact spread along the figure and mixed without any sharp distinction among them. The explained variance is [0.98, 0.01]

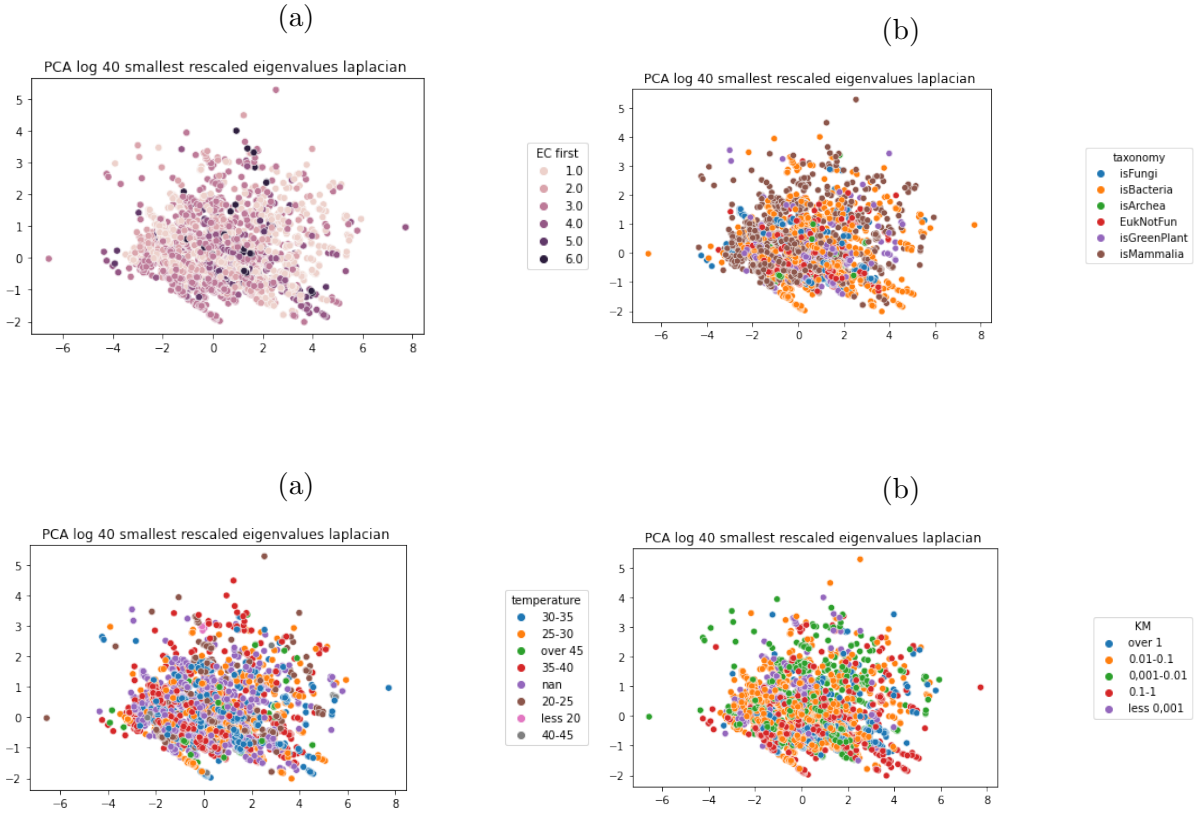
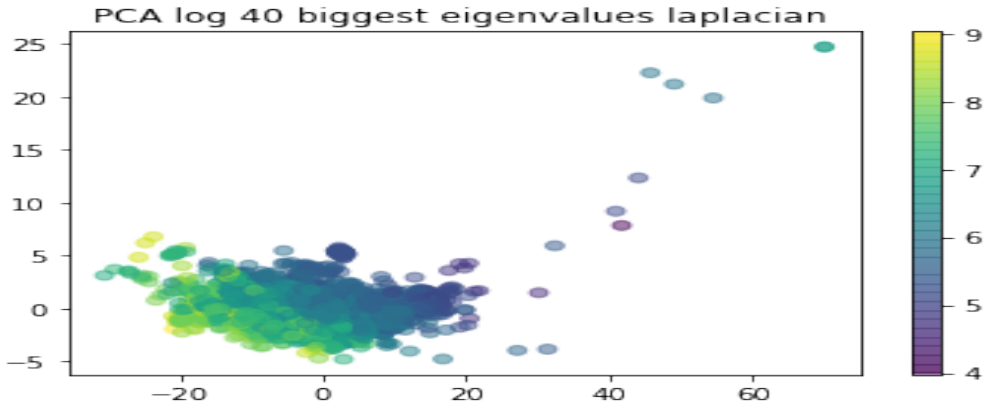
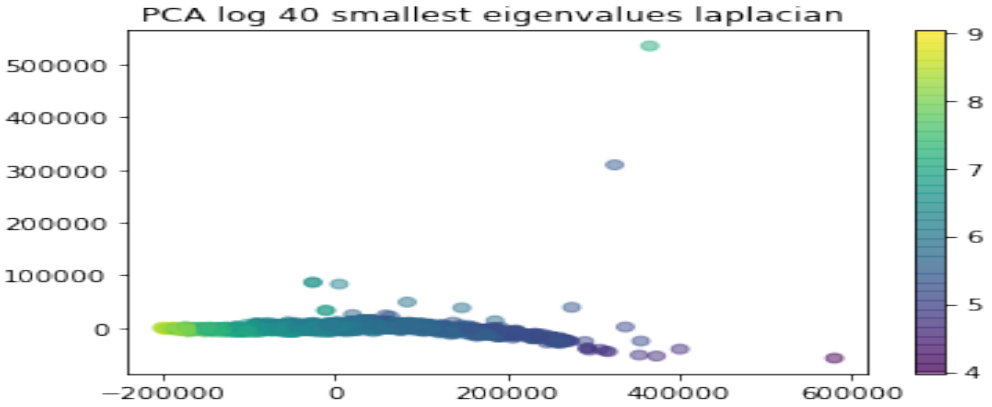


Figure 5.5: Are here represented the scatterplots of the PCA of the 40 smallest eigenvalues rescaled of the laplacian (L) obtained with the threshold set to 8 $\mathring{A}$. Figure (a) is colored by EC first label, figure (b) by taxonomy, figure (c) by temperature and in the end, figure (d) by KM. Apparently there is no possibility of conducting discriminant classification of the different lables. All the classes are infact spread along the figure and mixed without any sharp distinction among them. The explained variance is [0.92, 0.03]

(a)



(b)



(c)

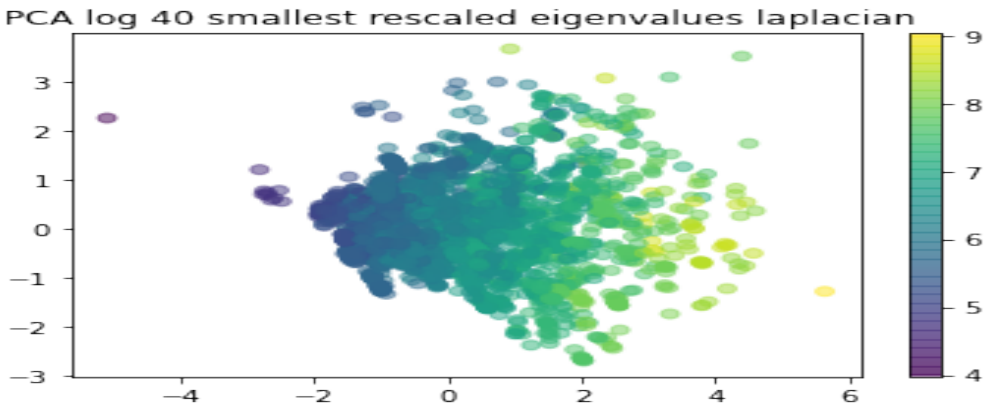
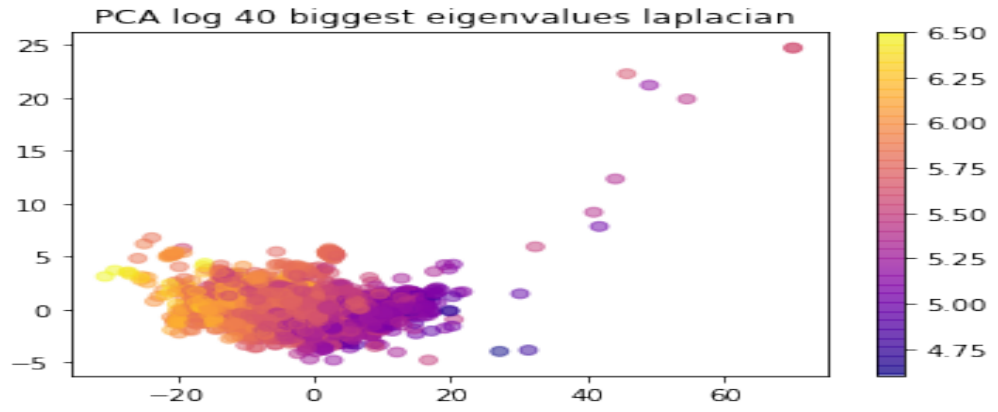
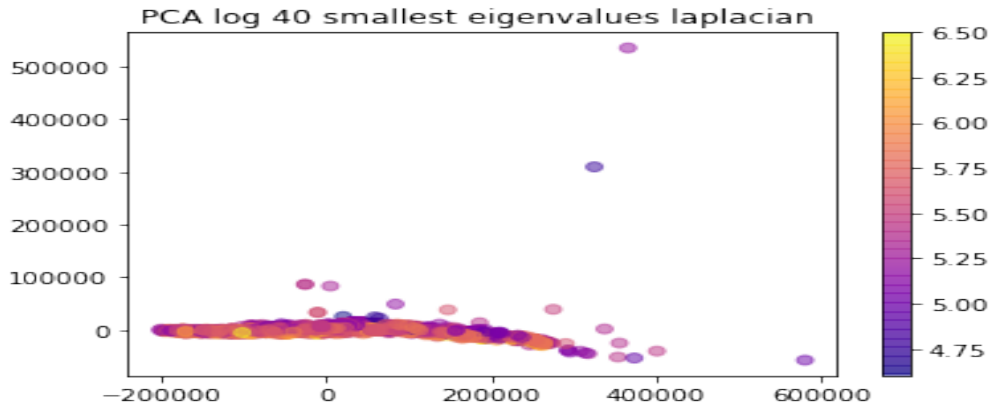


Figure 5.6: In these figures are represented the PCA analysis for the log of the 40 biggest (a), the log of the 40 smallest (b) and the smallest srescaled eigenvalues of the laplacian obtained from contact maps of threshold 8 $\mathring{A}$, with respect to the logarithm of the number of nodes. It can be seen from them, a neat distinction among number of nodes. I have applied the logarithm for both the eigenvalues and the number of nodes as they show better the dependence. The gradient of color is along the horizontal direction.

(a)



(b)



(c)

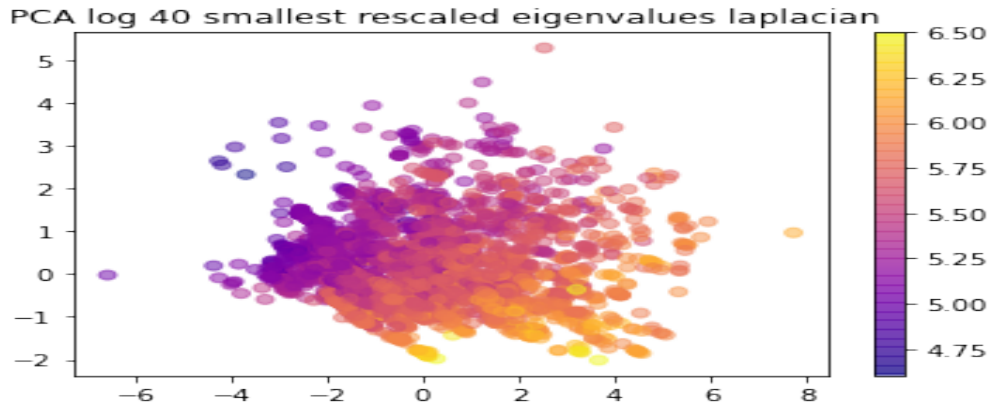


Figure 5.7: In these figures are represented the PCA analysis for the log of the 40 biggest (a), the log of the 40 smallest (b) and the smallest srescaled eigenvalues of the laplacian obtained from contact maps of threshold 8 $\mathring{A}$, with respect to the logarithm of the link density. It can be seen from them, a neat distinction among link density. I have applied the logarithm for both the eigenvalues and the number of nodes as they show better the dependence. It is curious that the gradient is along the vertical direction while for number of nodes it was on the horizontal direction

laplacian L and (5.10a, 5.10b, 5.10c, 5.10d) show it for the smallest rescaled eigenvalues. Figures (5.11a, 5.11b, 5.11c) show the t-SNE analysis for 40 biggest,smallest and smallest rescaled with respect the number of nodes. Figures (5.12a, 5.12b, 5.12c) show the t-SNE analysis for 40 biggest, smallest and smallest rescaled with respect the link density.

In the end I have chosen one representant for each uniprot code and reapeted the analysis. In figures (5.13a, 5.13b, 5.13c) show the t-SNE analysis for 40 biggest,smallest and smallest rescaled with respect the number of nodes. In Figures (5.14a, 5.14b, 5.14c) show the t-SNE analysis for 40 biggest, smallest and smallest rescaled with respect the link density. We can see from all these figures many different clusters; however these clusters don't represent the classification given by EC first, taxonomy, temperature and KM labels. Despite the goal of rescaling was to render the embedding independent of the number of nodes, we could not remove this dependence in this way. Link density seems to be in general uniformly clustered even though there is some mixing of colors in the different clusters that need to be investigated further. It seems the case that many groups of the same uniprot are held together (maybe this reflects the fact that inside each uniprot there is no much variety in number of nodes) at least in the smallest and smallest rescaled case. A peculiar fact is showed in the analysis of biggest eigenvalues that show that the 554 Fungi belonging to the same uniprot code 'P11838' are split into two subgroups. I think, that even though some difference in the distribution of the 40 eigenvalues is spotted, that distance doesn't picture 'real distance' and is induced by the dimensionality reduction (I think it is exaggerated in this case). In any case, this this distinction (not hinted by any measure so far done) would need some further analysis, even though UMAP doesn't show this effect. A little comment is deserved by pictures of the analysis for 496 structures chosen as representant of each uniprot. I have decided to put them to see if some cluster of enzymes could have formed. Just the smallest rescaled case shows some clustering.

5.4 UMAP analysis

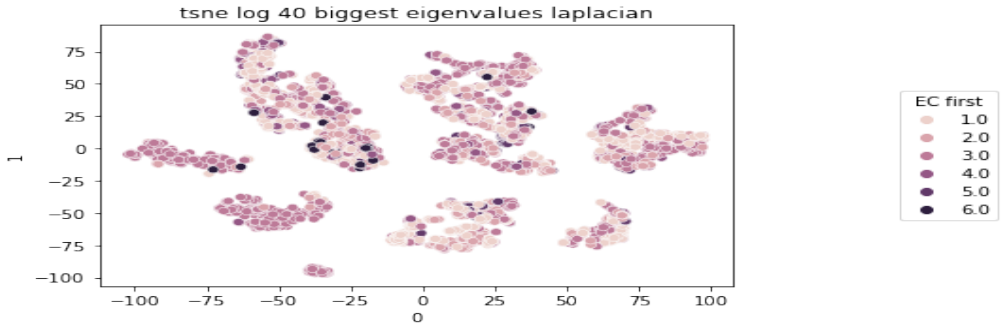
5.4.1 UMAP laplacian

In this section, Figures (5.15a,5.15b,5.15c,5.15d) show the UMAP analysis for 40 biggest eigenvalues of the laplacian L. Figures, (5.16a,5.16b,5.16c,5.16d) show the UMAP analysis for 40 smallest eigenvalues of the laplacian L. Figures (5.17a,5.17b,5.17c,5.17d) show the UMAP analysis for 40 smallest eigenvalues of the laplacian L. Figures (5.18a 5.18b,5.18c) show the UMAP analysis for 40 biggest, smallest and smallest rescaled eigenvalues of the laplacian L with respect the number of nodes. Figures (5.19a 5.19b,5.19c) show the UMAP analysis for 40 biggest, smallest and smallest rescaled eigenvalues of the laplacian L with respect the link density.

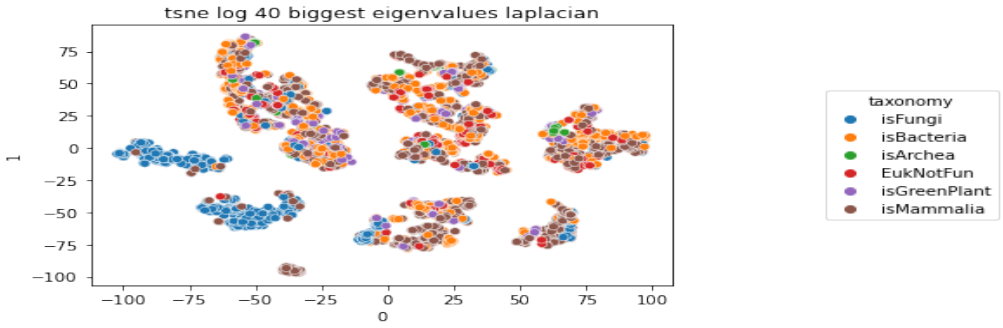
I then decided to choose a representant structure for each uniprot code and plot it in Figures (5.20a 5.20b,5.20c) for 40 biggest, smallest and smallest rescaled eigenvalues of the laplacian L with respect the number of nodes, and Figures (5.21a 5.21b,5.21c) for 40 biggest, smallest and smallest rescaled eigenvalues of the laplacian L with respect the link density.

We can see from all these figures many different clusters; however these clusters seem not to represent EC first, taxonomy, temperature and KM. However in my goal was to obtain number of nodes-indipendent plots, we see that that is not the case for all the three cases. This maybe the main cause of similarity of different structures sharing the

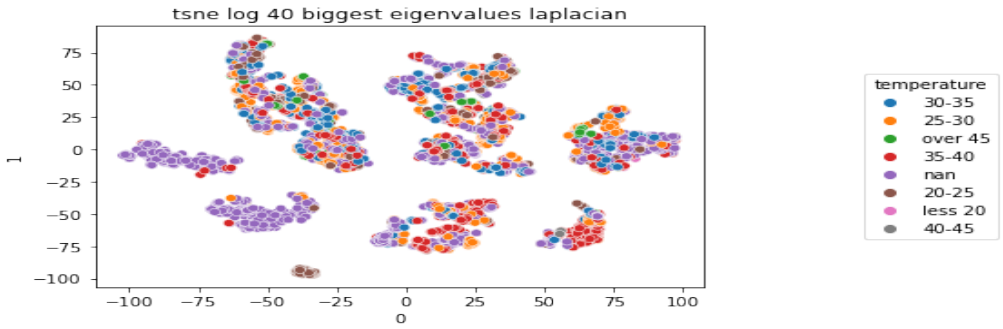
(a)



(b)



(c)



(d)

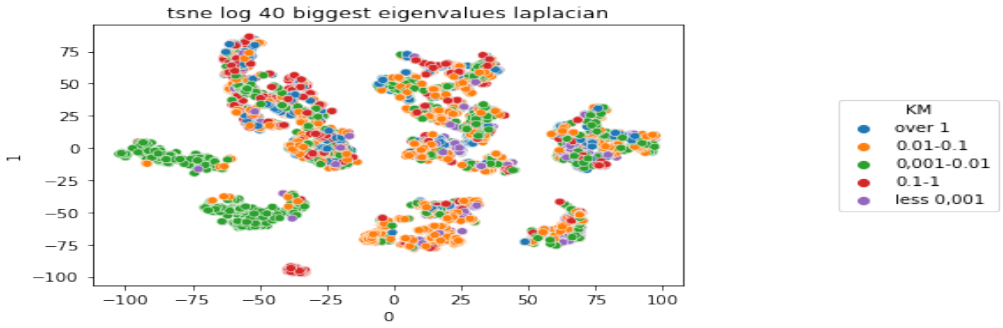
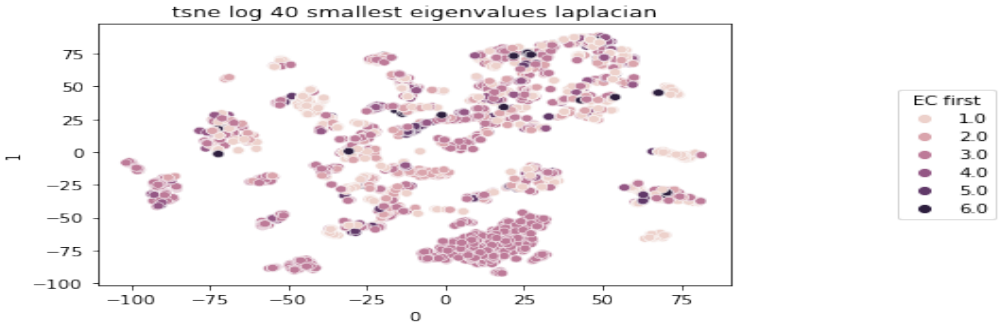
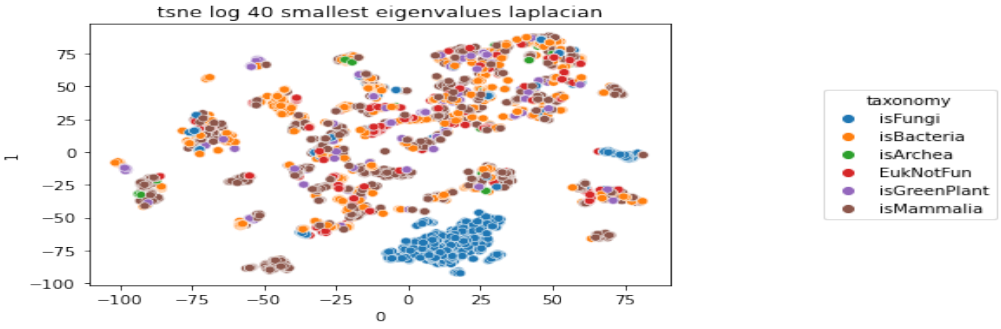


Figure 5.8: In this four figures are represented the tsne of the biggest 40 eigenvalues of the laplacian with respect to the label EC first (a), taxonomy (b), temperature (c), and KM (d) of the enzymes. They are representing many different groups that seem not to strictly qualify the enzymes via any of these labels. Around (-50,-50) can be seen the cluster of enzymes belonging the same family, 'P11838'. Part of 'P11838' is in around (-75,0)

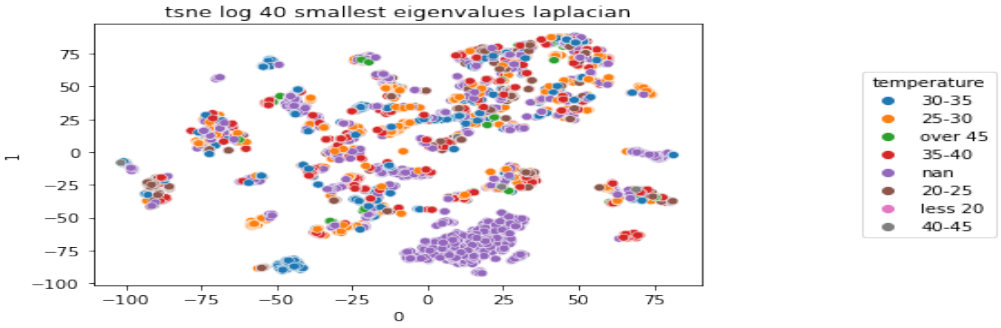
(a)



(b)



(c)



(d)

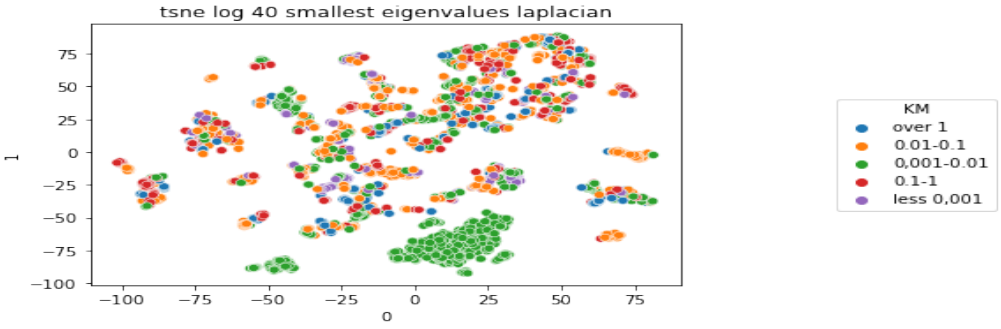
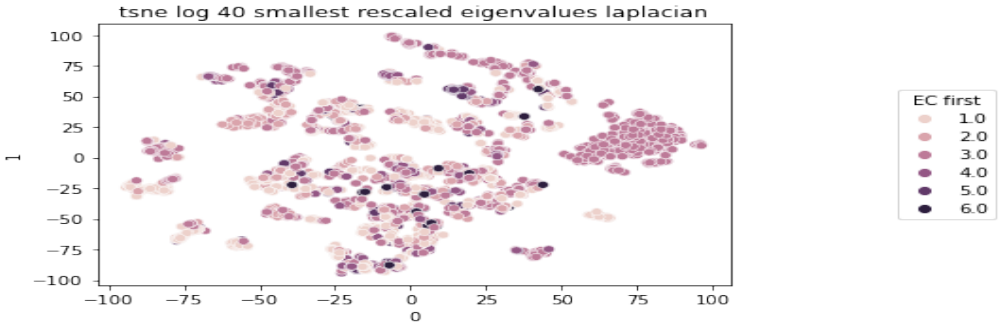
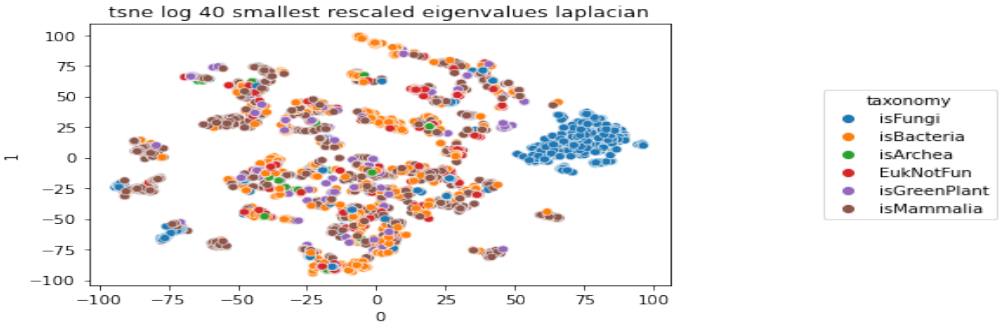


Figure 5.9: In these four figures, are represented the tsne of the smallest 40 eigenvalues of the laplacian with respect to the label EC first (a), taxonomy (b), temperature (c) and KM (d) of the enzymes.They are representing many different groups that seem not to strictly qualify the enzymes via any of these labels. Around (0,-75) can be seen the cluster of enzymes belonging to the same family, 'P11838'.

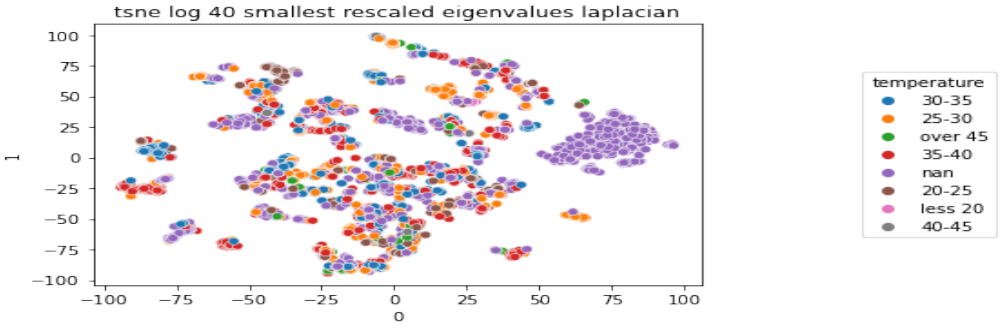
(a)



(b)



(c)



(d)

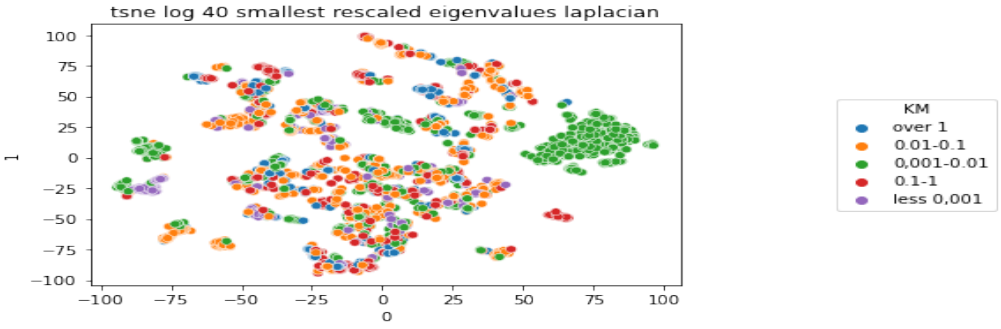
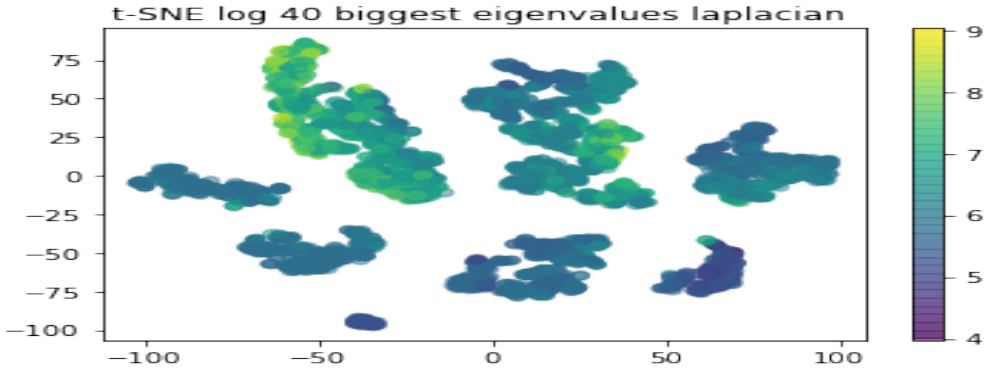
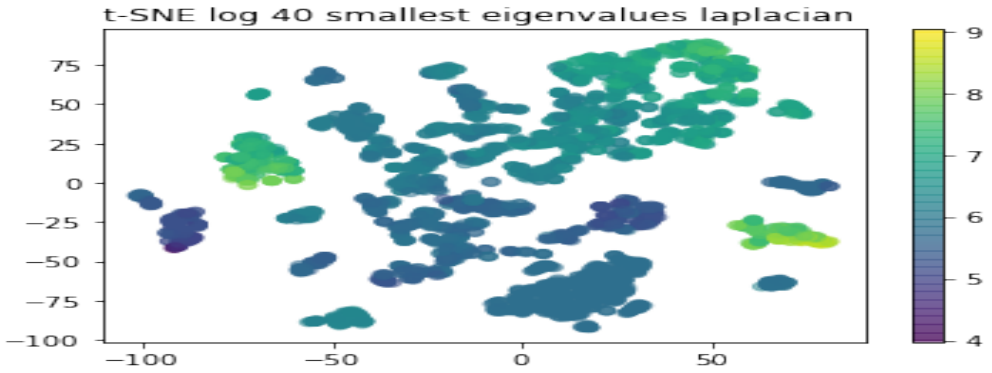


Figure 5.10: In these four figures, are represented the tsne of the smallest 40 eigenvalues of the laplacian with respect to the label EC first (a), taxonomy (b), temperature (c) and KM (d) of the enzymes.They are representing many different groups that seem not to strictly qualify the enzymes via any of these labels. Around (0,-75) can be seen the cluster of enzymes belonging to the same family, 'P11838'.

(a)



(b)



(c)

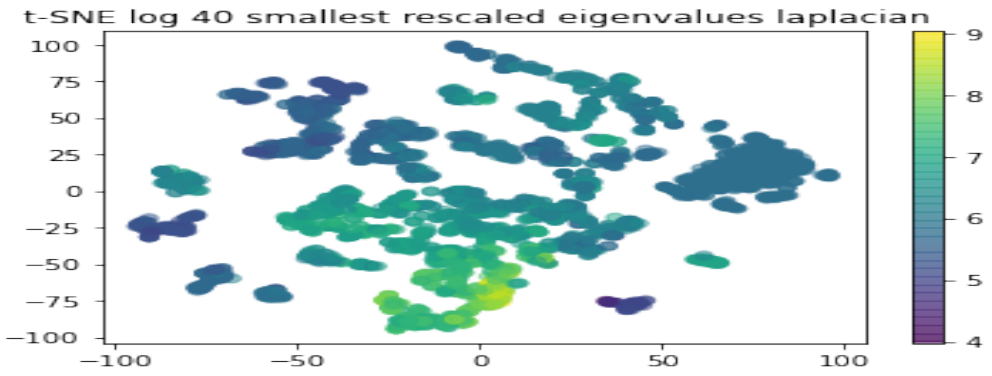
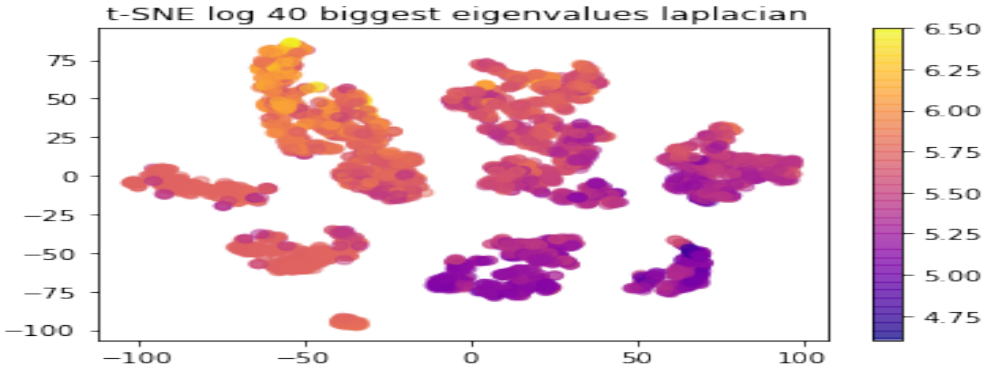
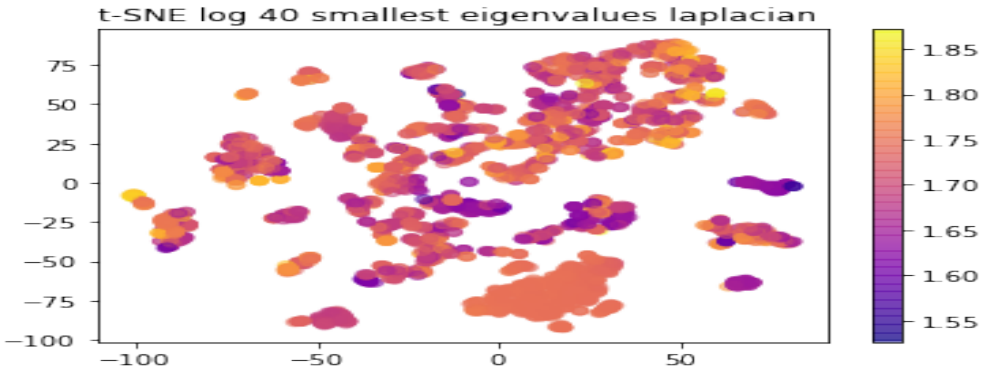


Figure 5.11: In these figures are represented the t-SNE analysis of the 40 biggest (a) and smallest (b) eigenvalues and smallest eigenvalues rescaled (c) labeled by their number of nodes

(a)



(b)



(c)

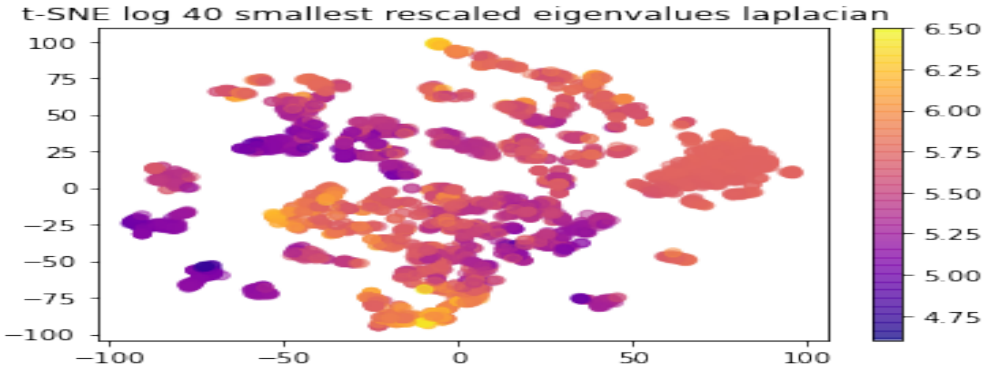
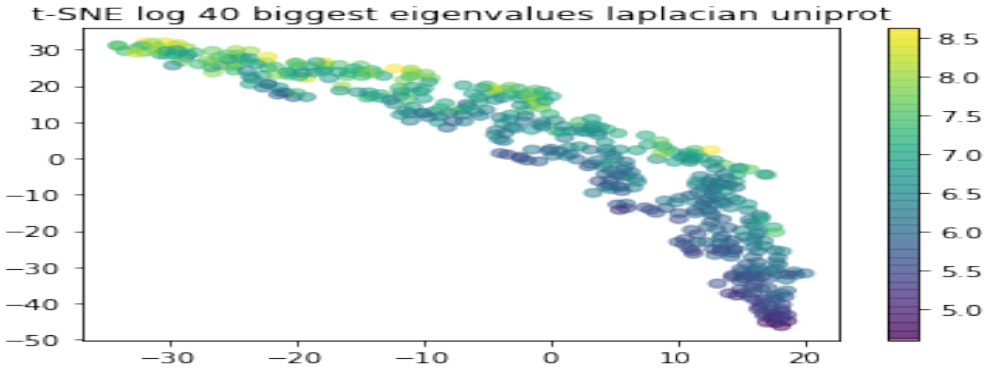
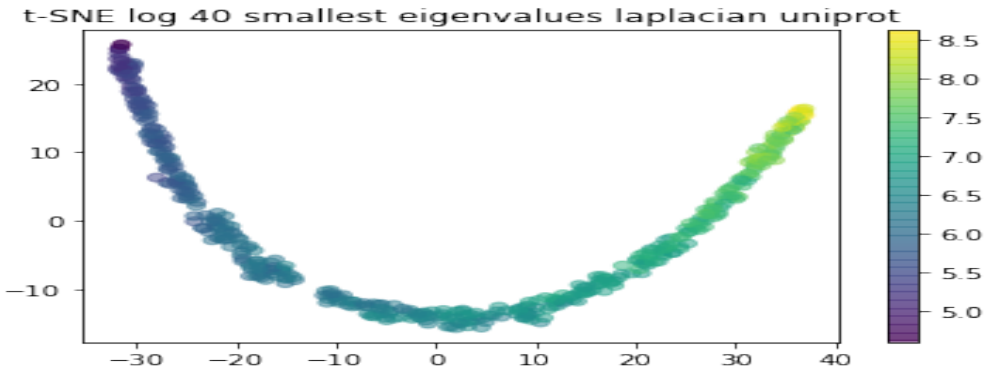


Figure 5.12: In these figures are represented the t-SNE analysis of the 40 biggest (a) and smallest (b) eigenvalues and smallest eigenvalues rescaled (c) labeled by their link density

(a)



(b)



(c)

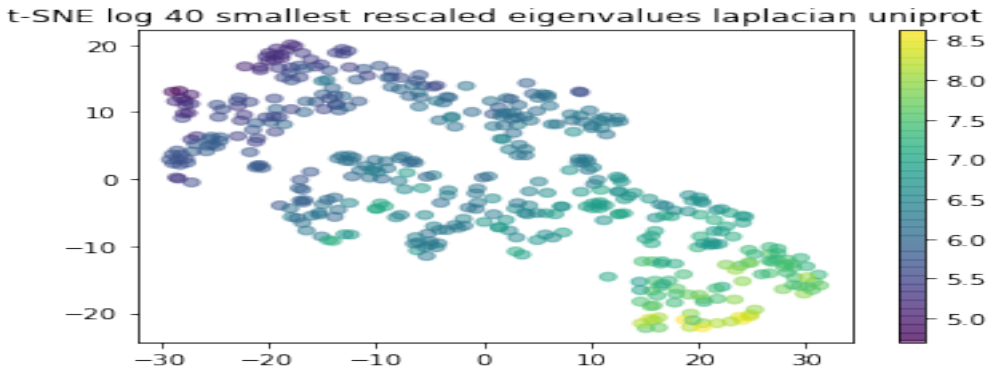
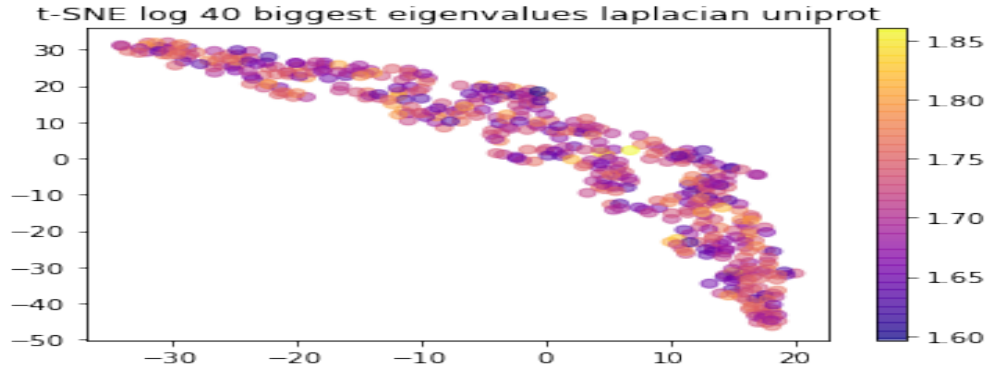
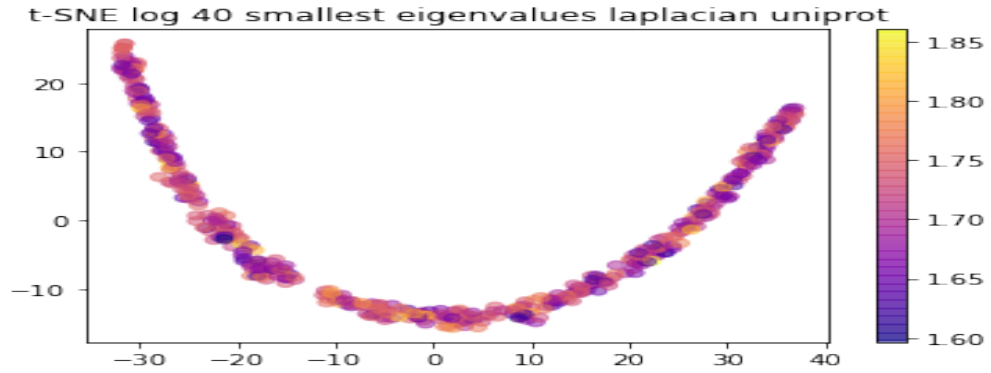


Figure 5.13: In these figures are represented the t-SNE analysis of the 40 biggest (a) and smallest (b) eigenvalues and smallest eigenvalues rescaled (c) labeled by their number of nodes. In these figures I have taken 1 representant for each uniprot code

(a)



(b)



(c)

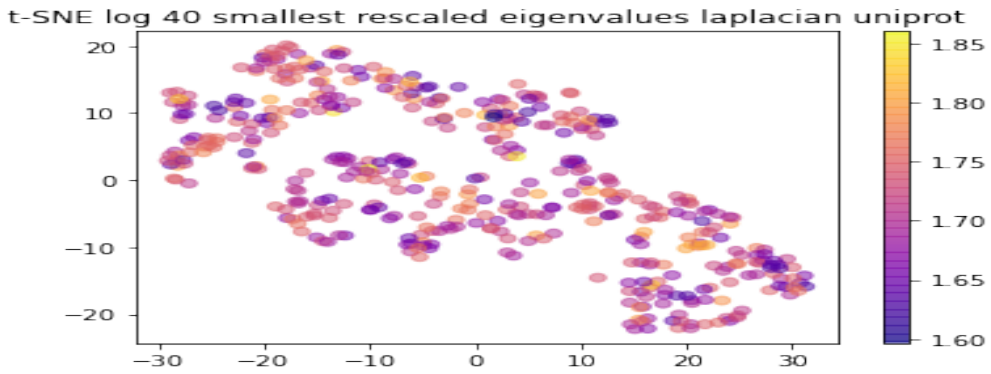
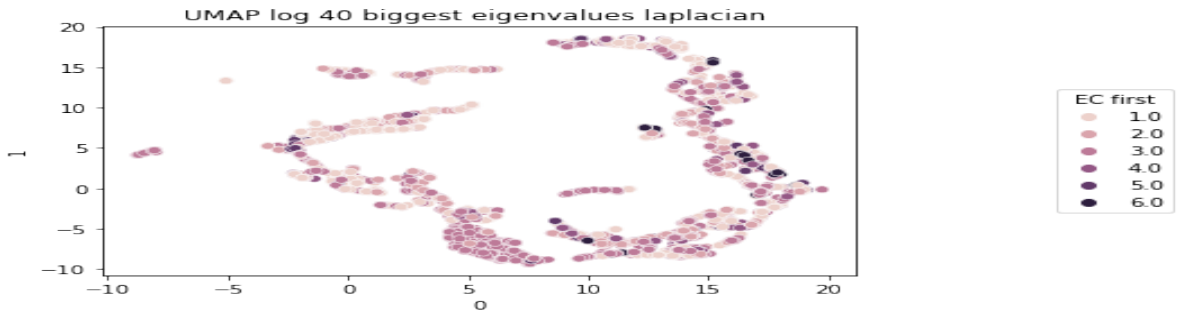


Figure 5.14: In these figures are represented the t-SNE analysis of the 40 biggest (a) and smallest (b) eigenvalues and smallest eigenvalues rescaled (c) labeled by their link density. In these figures I have taken 1 representant for each uniprot code

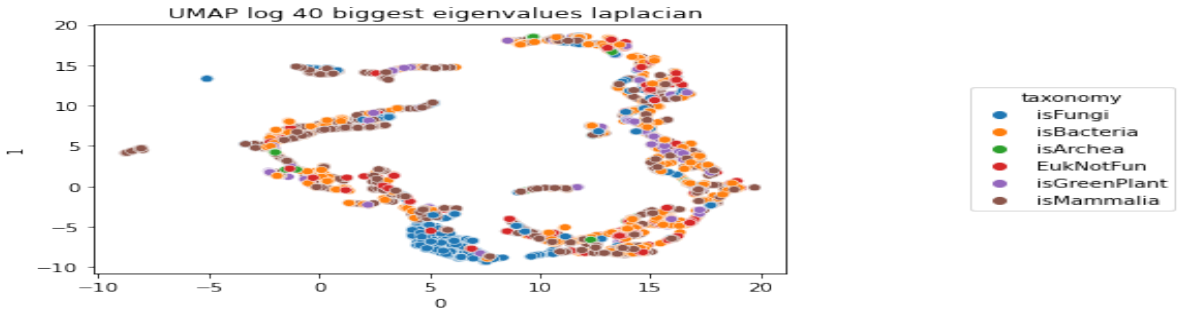
same uniprot (as suggested already on t-SNE section). Again the link density doesn't seem to be the principal factor for clusters. In the end I have chosen to show the analysis for a more restricted set of enzymes to see wether clusters could have formed for different sequences. With the parameter of nearest neighbors set to 10 I have found that just the smallest rescaled shows clusters. Increasing the number of nodes the clusters tend to vanish. In any case, smallest rescaled eigenvalues seem those more recommended for finding structures independently of the number of nodes both in UMAP and t-SNE.

This would need further investigation.

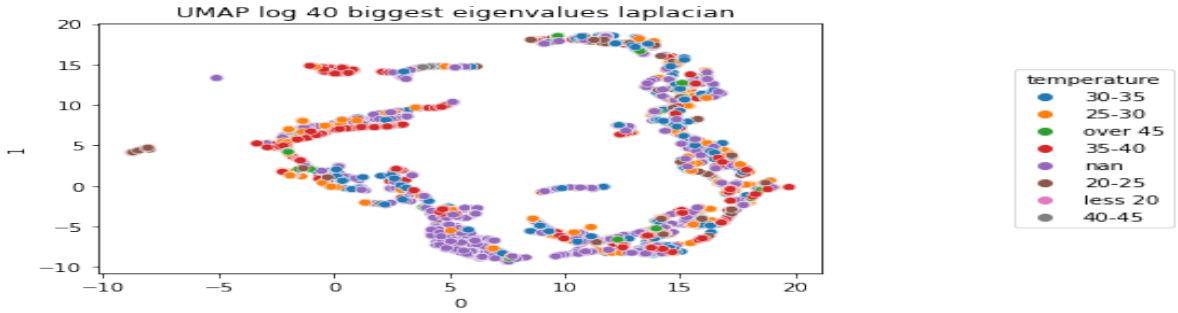
(a)



(b)



(c)



(d)

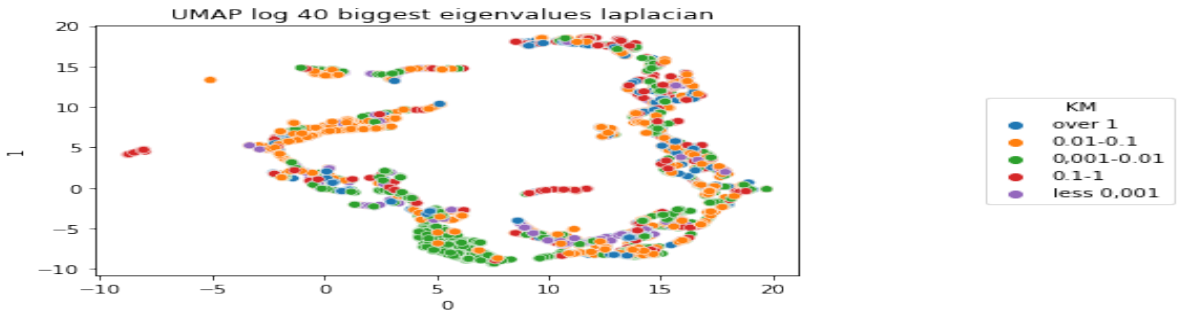
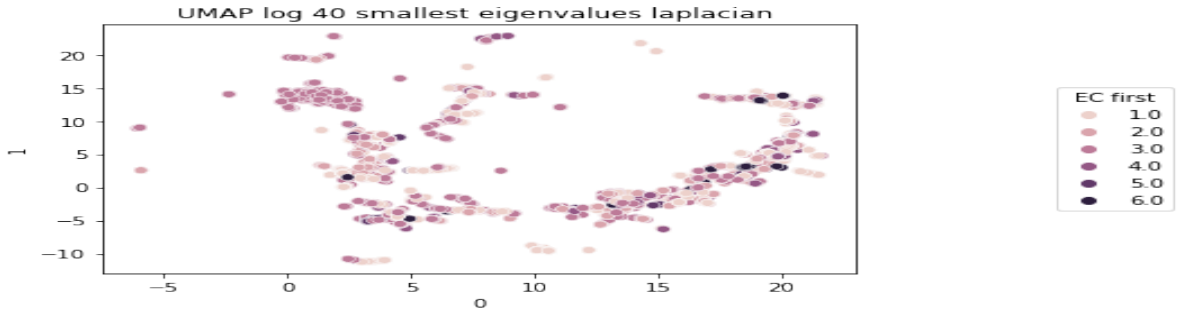
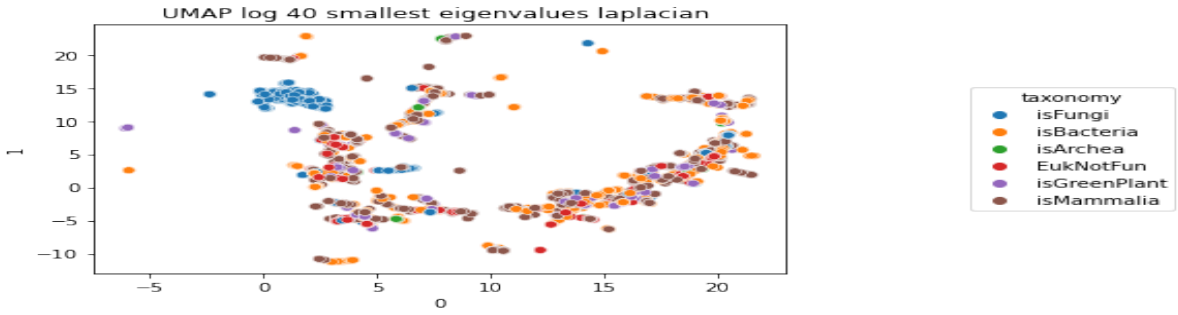


Figure 5.15: In these four figures are repersented the UMAP analysis of the 40 biggest eigenvalues of the laplacian (L) and labeled by EC first (a), taxonomy (b), temperature (c) and KM (d). The parameter of nearest neighbors is set to 5. They are representing different groups (less than in the case of t-SNE) that seem not to strictly qualify the enzymes via any of the labels. Around (5,-10) can be seen the cluster of enzymes belonging to the same family, 'P11838'.

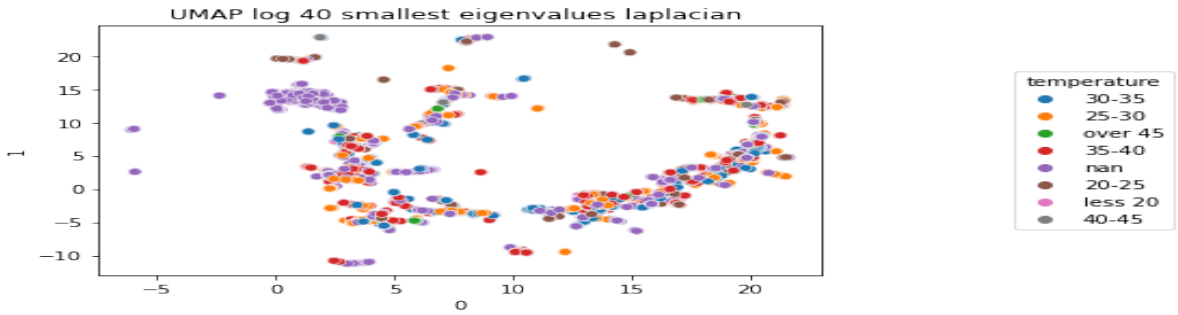
(a)



(b)



(c)



(d)

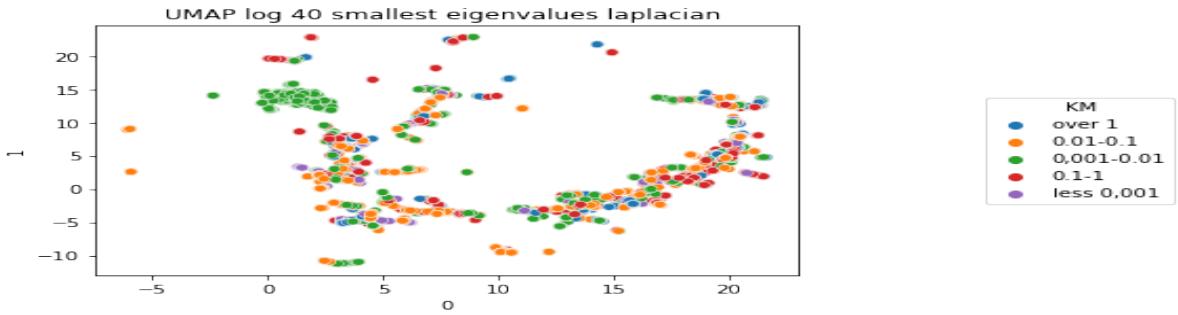
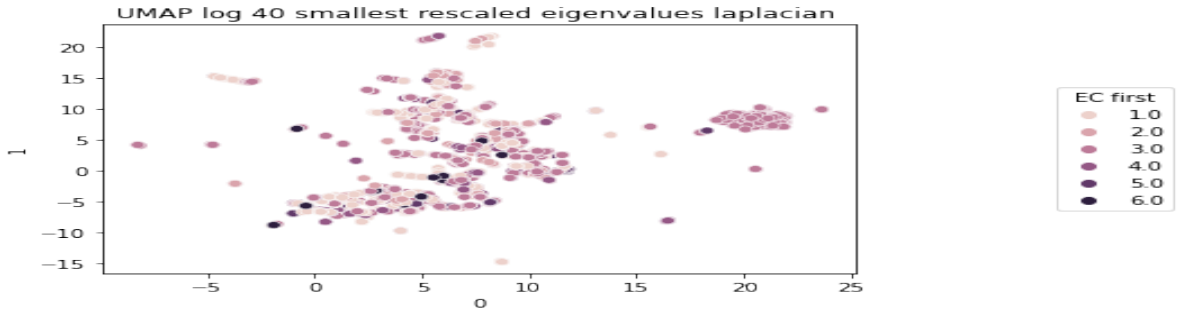
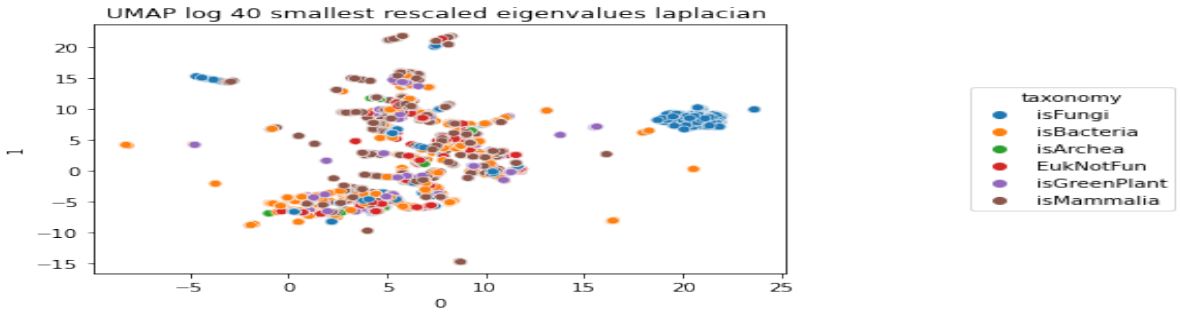


Figure 5.16: In these four figures are represented the UMAP analysis of the 40 smallest eigenvalues of the laplacian (L) labeled by EC first (a),taxonomy (b), temperature (c) and KM (d). The parameter of nearest neighbors is set to 5. They are representing different groups (less than in the case of t-SNE) that seem not to strictly qualify the enzymes via any of the labels. Around (0,10) can be seen the cluster of enzymes belonging to the same family, 'P11838'

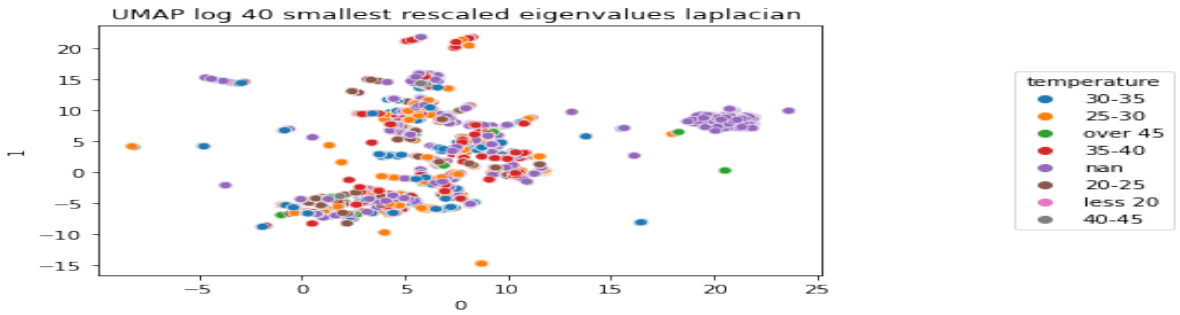
(a)



(b)



(c)



(d)

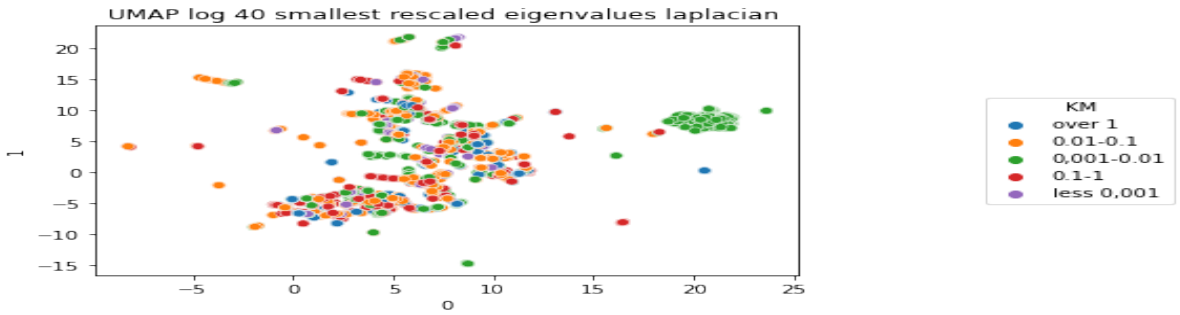
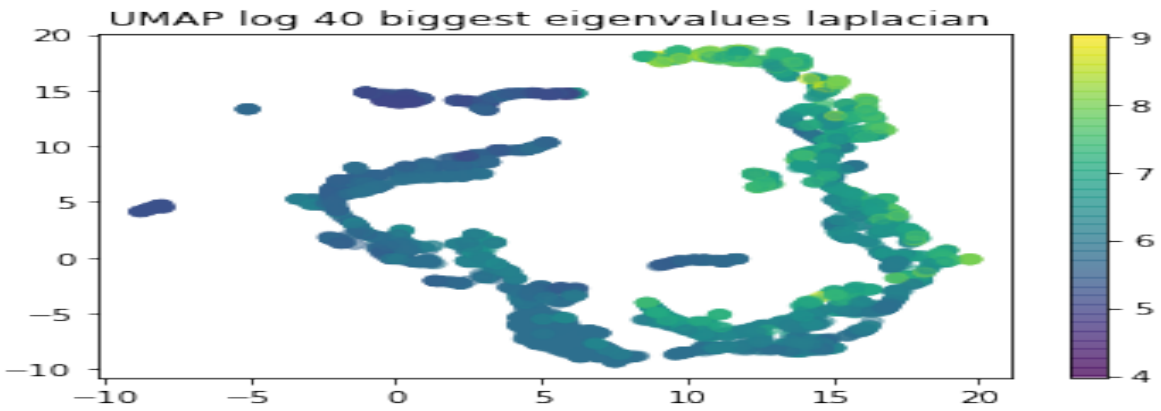
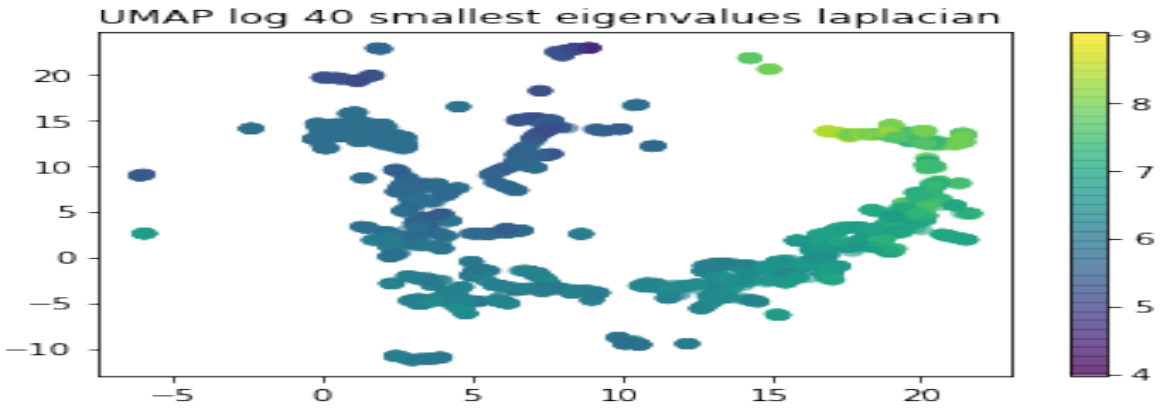


Figure 5.17: In these four figures are repersented the UMAP analysis of the 40 smallest eigenvalues rescaled of the laplacian (L) at 8 $\mathring{A}$ and labeled by EC first (a), taxonomy (b), temperature (c) and KM (d). The parameter of nearest neighbors is set to 5. They are representing different groups (less than in the case of t-SNE) that seem not to strictly qualify the enzymes via any of the labels. Around (20,10) can be seen the cluster of enzymes belonging to the same family, 'P11838'

(a)



(b)



(c)

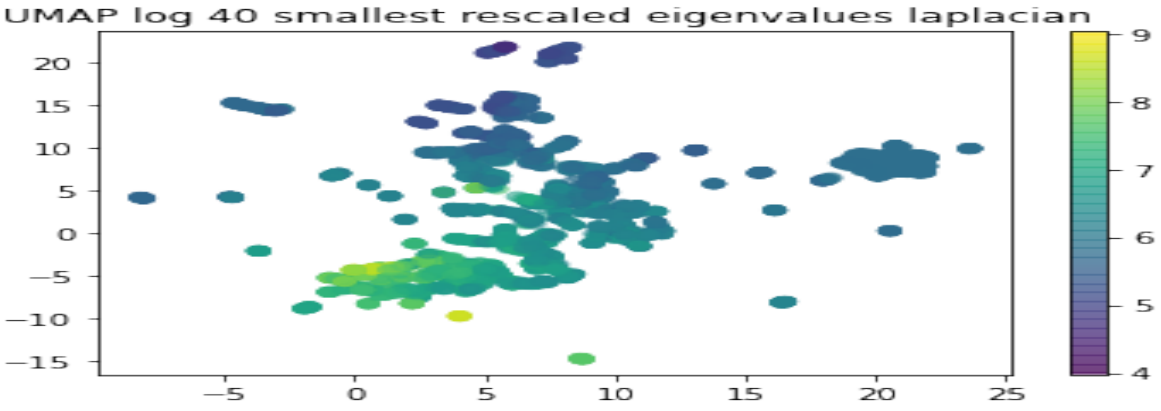
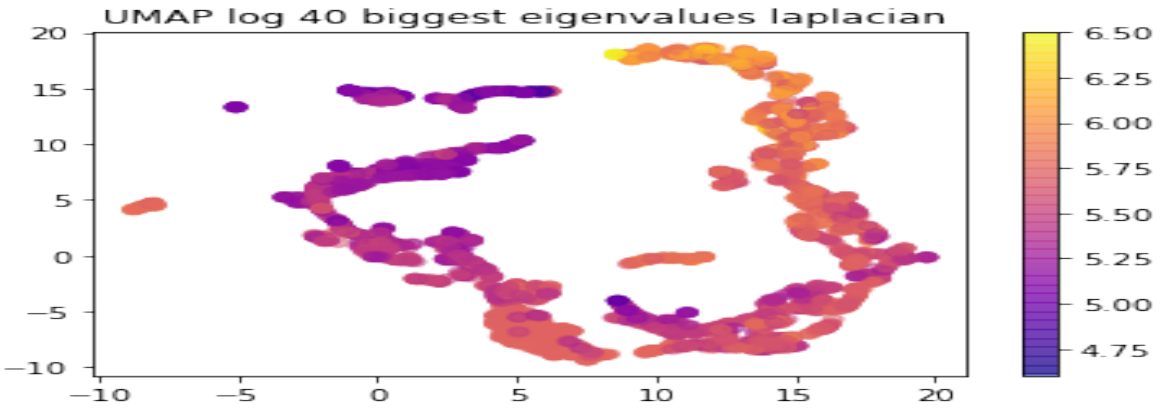
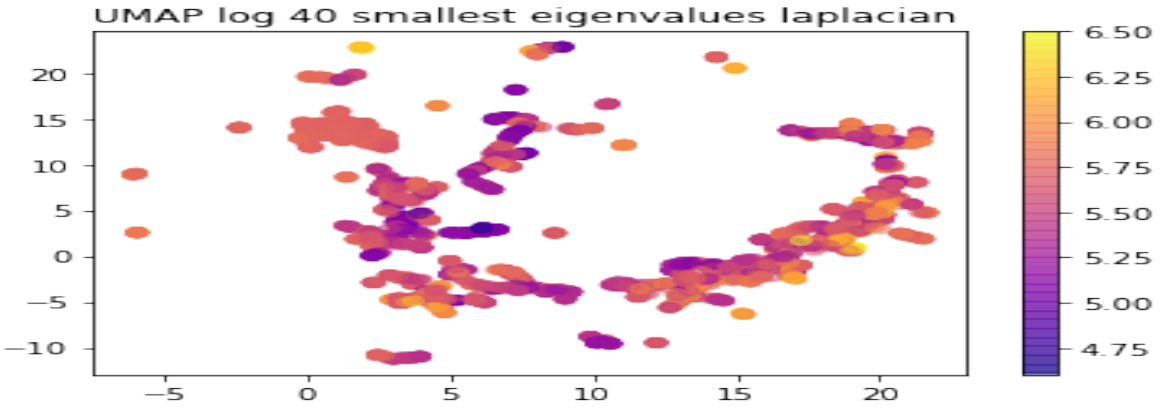


Figure 5.18: In these figures are represented the UMAP analysis of the 40 biggest (a), smallest (b) eigenvalues and smallest rescaled eigenvalues of the laplacian with threshold 8 $\mathring{A}$ and labeled by number of nodes represented in the log scale in the colormap. The parameter of nearest neighbors for (a) and (b) is set to 5 while for (c) is set to 10 as at 5 it was little too noisy.

(a)



(b)



(c)

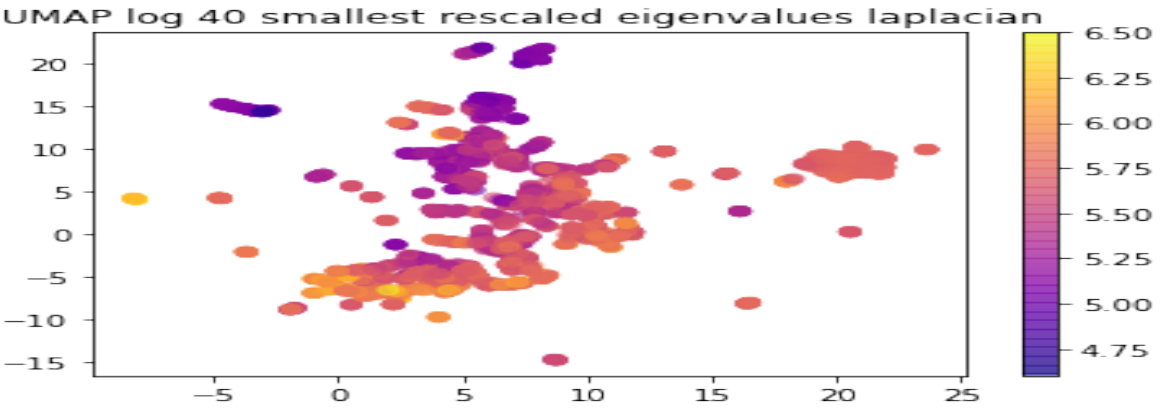
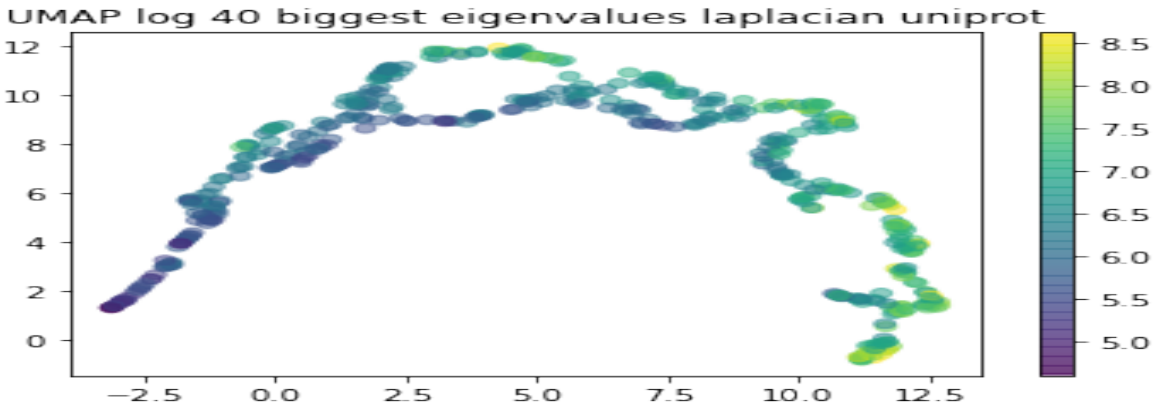
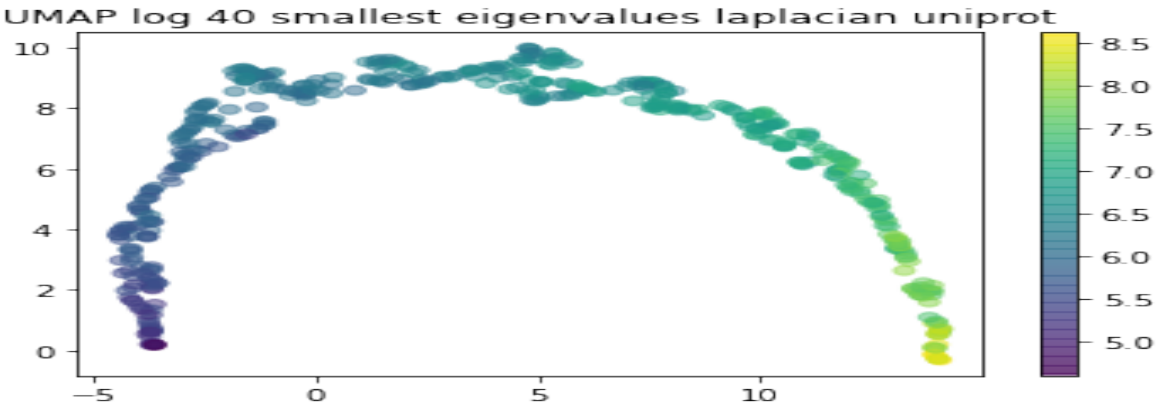


Figure 5.19: In these figures are represented the UMAP analysis of the 40 biggest (a), smallest (b) eigenvalues and smallest rescaled eigenvalues of the laplacian with threshold 8 $\mathring{A}$ and labeled by link density represented in the colormap. The parameter of nearest neighbors for (a) and (b) is set to 5 while for (c) is set to 10 as at 5 it was little too noisy.

(a)



(b)



(c)

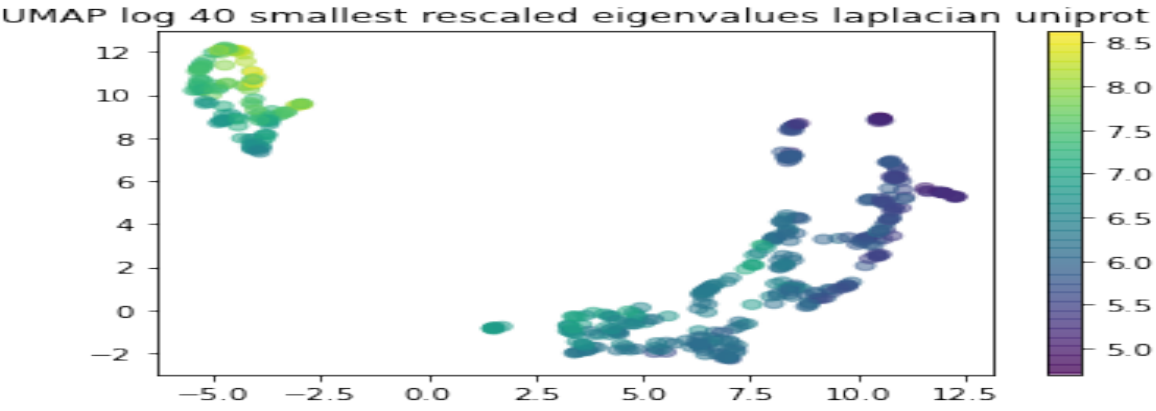
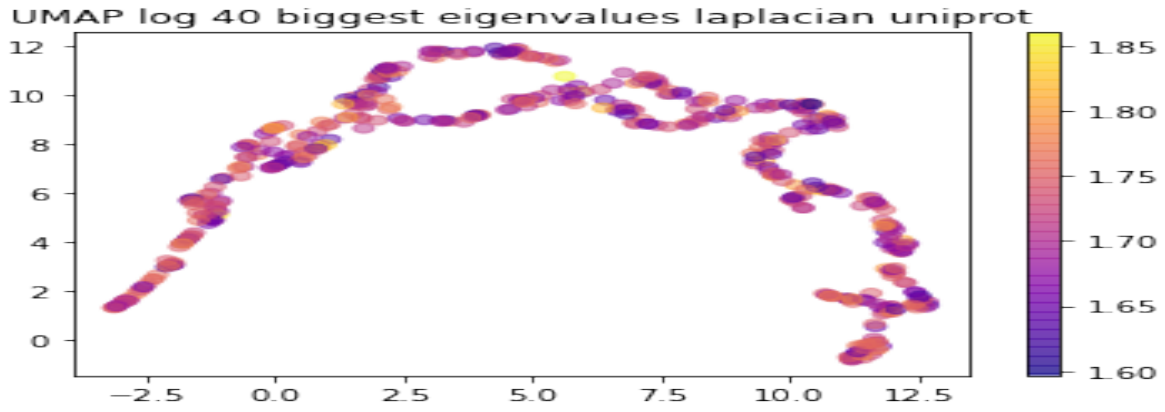
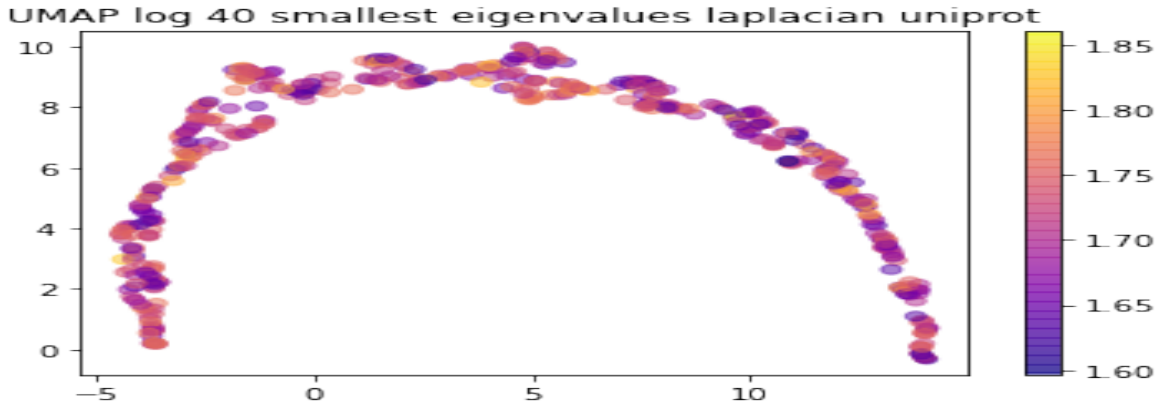


Figure 5.20: In these figures are represented the UMAP analysis of the 40 biggest (a), smallest (b) eigenvalues and smallest rescaled eigenvalues of the laplacian with threshold 8 $\mathring{A}$ of one structure form each uniprot and labeled by number of nodes represented in the colormap. The parameter of nearest neighbors for (a) and (b) and (c) are set to 10. There is a clear relation among number of nodes and the similarity among enzymes.

(a)



(b)



(c)

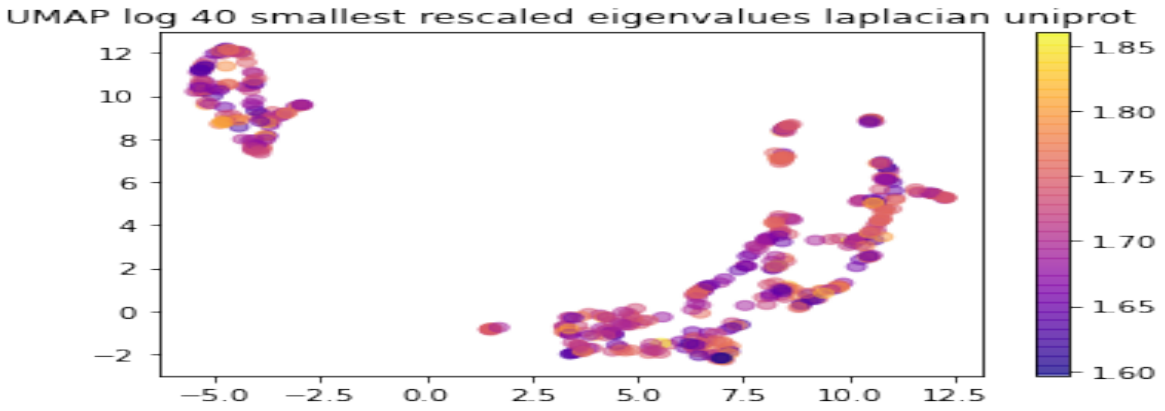


Figure 5.21: In these figures are represented the UMAP analysis of the 40 biggest (a), smallest (b) eigenvalues and smallest rescaled eigenvalues of the laplacian with threshold 8 $\mathring{A}$ and labeled by link density for each uniprot. The parameter of nearest neighbors for (a) and (b) and (c) are set to 10.

Chapter 6

Conclusions

The goal of this study is to show if geometrical features captured by Laplacian operator, associated to enzyme's contact maps, is able to represent similar characteristics of enzymes (EC, taxonomy, temperature, KM, link density and number of carbon alpha). In this study, I have considered the spectral properties of laplacian L and normalized laplacian $\mathcal{L}$.

As a first step of my analysis, I have explored the dataset, and the descriptive labels I had.

From a statistical analysis in which I compared distributions of KM and temperatures for all the different couples of families in taxonomy and EC, I see that: Archea live in different (higher) temperature environments with respect to all the others. KM is uniformly distributed but for Bacteria. The distributions of temperatures of class EC '2' and '4' are different from the others, and KM are distributed uniformly with respect all the ECs.

Secondly, I considered the network features to characterize the contact maps of the dataset. I have seen that I work with 'small world' graphs, looked at degree distributions and average clustering coefficients. I notice that exist 554 structures that belong to the same uniprot P11838, that are very similar among them. This is a general fact; different PDBs, associated to the same uniprot, have similar network's characteristics. The P11838 case, is taken as a benchmark to measure the similitude among structures in the spaces reconstructed by PCA, t-SNE and UMAP.

For random networks, for both L and $\mathcal{L}$, it is known that exists a relation between the number of nodes and link density to eigenvalues. This fact is present in my dataset. I have found changing polynomial dependence between 40 biggest eigenvalues and link density, and 40 smallest eigenvalues and number of nodes. I have tried a stratified rescaling for each smallest eigenvalue and number of nodes, to avoid the dependence on the size of the enzyme.

I now had, as features' space: 40 biggest, smallest and smallest rescaled eigenvalues. The dependence on the number of nodes is never disappeared. This dependence is confirmed particularly well in all the PCA analysis, it is, indeed, pretty visible a gradient and I conclude that the main factor of discrimination is the number of nodes (completed by the link density on the perpendicular direction).

In both t-SNE and UMAP, this dependence didn't deny the creation of other clusters. Infact those PDBs belonging to P11838 are always clustered together.

However, increasing (for both t-SNE and UMAP) the number of nearest neighbors, in the algorithms, the different clusters tend to disappear and form a unique cluster

where it is present the gradient in the number of nodes. In any case, the other labels (EC,taxonomy, temperature and KM) are not recognized. Infact none of those seems to be discriminated entirely.

Extracting the noise of redundancies of structures, choosing one structure for each uniprot code, it is visible always the dependence in the number of nodes, and just in the case of the smallest rescaled eigenvalues for UMAP, appear two very separated clusters that cannot be conducted to labels already available. A further, biological study, must be done to understand if these clusters are fictitious, dependent on the choice of the parameter of nearest neighbors, or show some true similarity among enzyme.

In general I haven't found a way to judge the validity of clusterings found by these algorithms, but the benchmark of P11838. So further more detailed studies need to be done when clusterings are found. In any case, smallest rescaled eigenvalues seem those eigenvalues that are able to spot more structure independently of the number of nodes. Besides UMAP produces more sharper distinction of clusters. To improve further the localization of clusters, it can be thought to 'rescale' with respect to the link density, or even, 'rescale' simoultaneously the biggest eigenvalues with respect to link density and the smallest eigenvalues with respect the number of nodes and build an 80 dimensional space for t-SNE and UMAP.

Appendices

Appendix

I now show the PCA,t-SNE and UMAP of normalized Laplacian at 8 $\mathring{A}$ the results in the same form of the chapter **Results**. For brevity, for laplacian and normalized laplacian at 12 $\mathring{A}$, I will avoid to show the PCA, and just show the dependencies on link density and number of nodes for both UMAP and t-SNE, as we have learnt that the captions are not well characterized.

.1 PCA normalized laplacian 8 $\mathring{A}$

.2 t-SNE normalized laplacian 8 $\mathring{A}$

.3 UMAP normalized laplacian 8 $\mathring{A}$

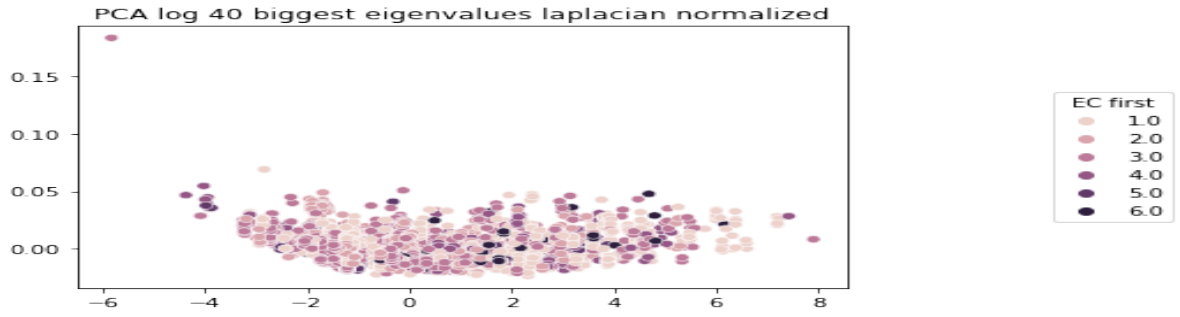
.4 t-SNE Laplacian 12 $\mathring{A}$

.5 UMAP Laplacian 12 $\mathring{A}$

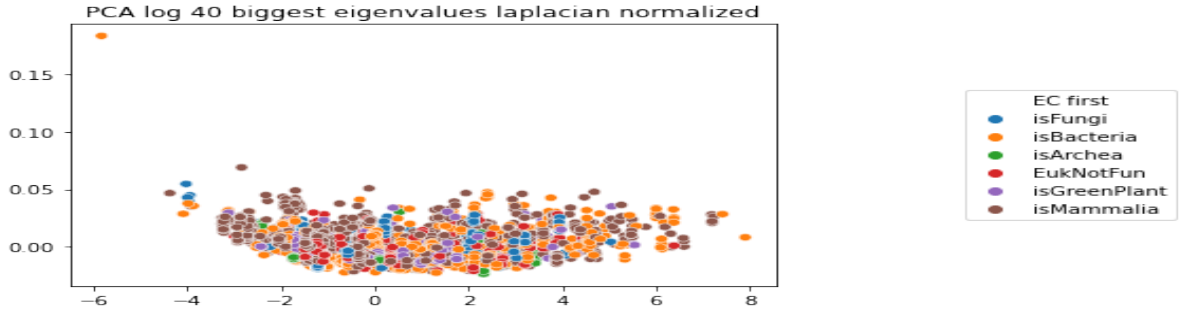
.6 t-SNE Normalized Laplacian 12 $\mathring{A}$

.7 UMAP Normalized Laplacian 12 $\mathring{A}$

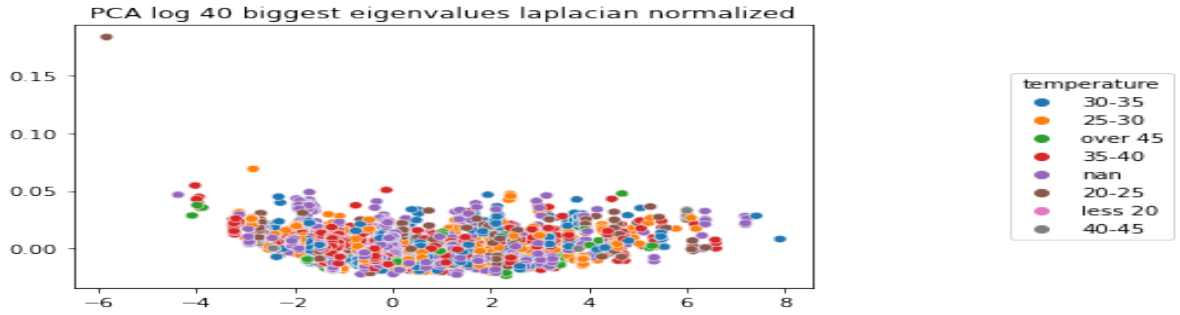
(a)



(b)



(c)



(d)

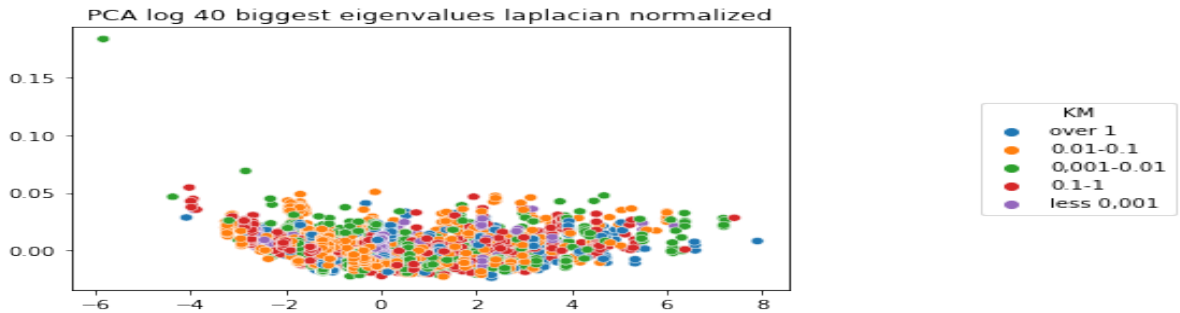
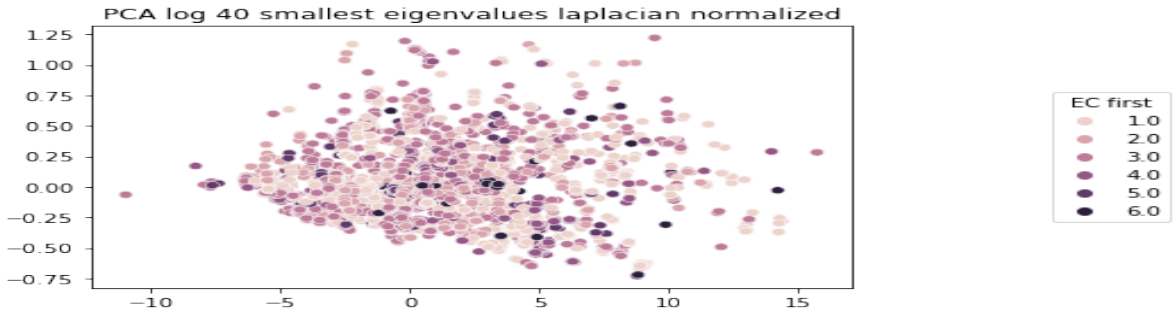
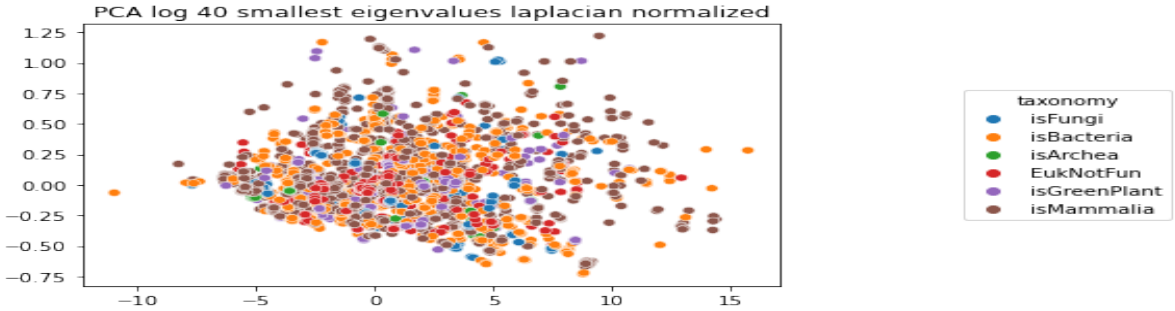


Figure 1: In these figures are represented the scatterplots of the PCA to the 40 biggest eigenvalues of the normalized laplacian ($\mathcal{L}$) of threshold 8 $\mathring{A}$ of enzymes colored with respect to their average EC first (a), taxonomy (b), temperature (c) and KM (d).Apparently there is no possibility of conducting discriminant classification of any of the different labels. All the classes are infact spread along the figure and mixed without any sharp distinction among them.The explained variance is The explained variance is: [0.99 0.01]

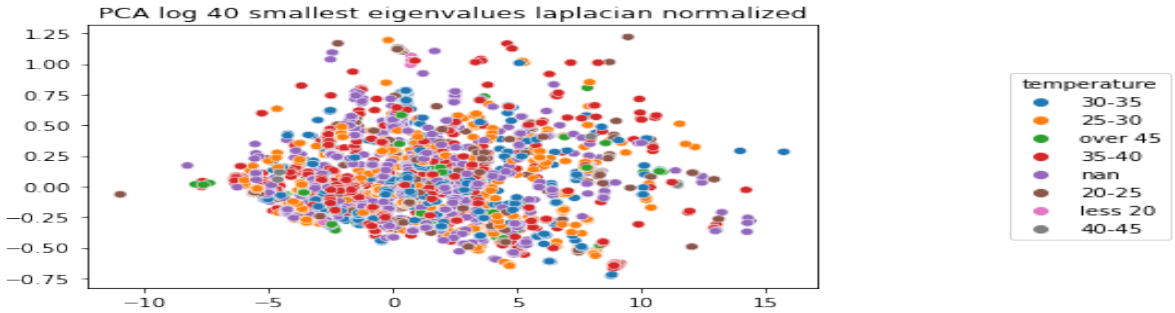
(a)



(b)



(c)



(d)

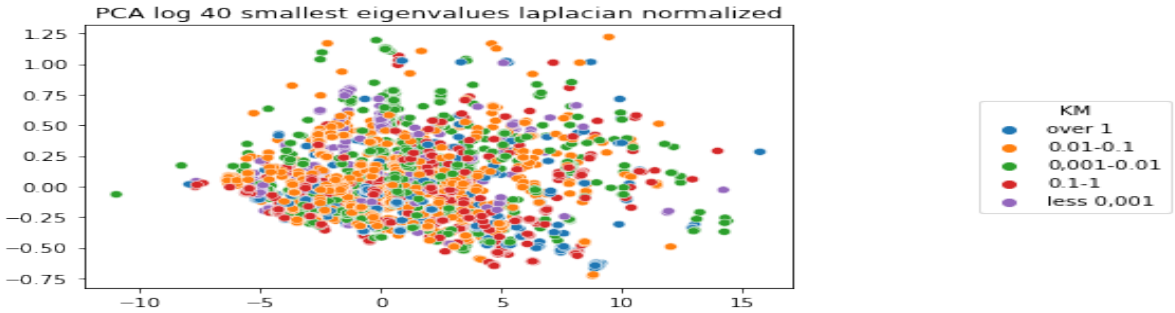
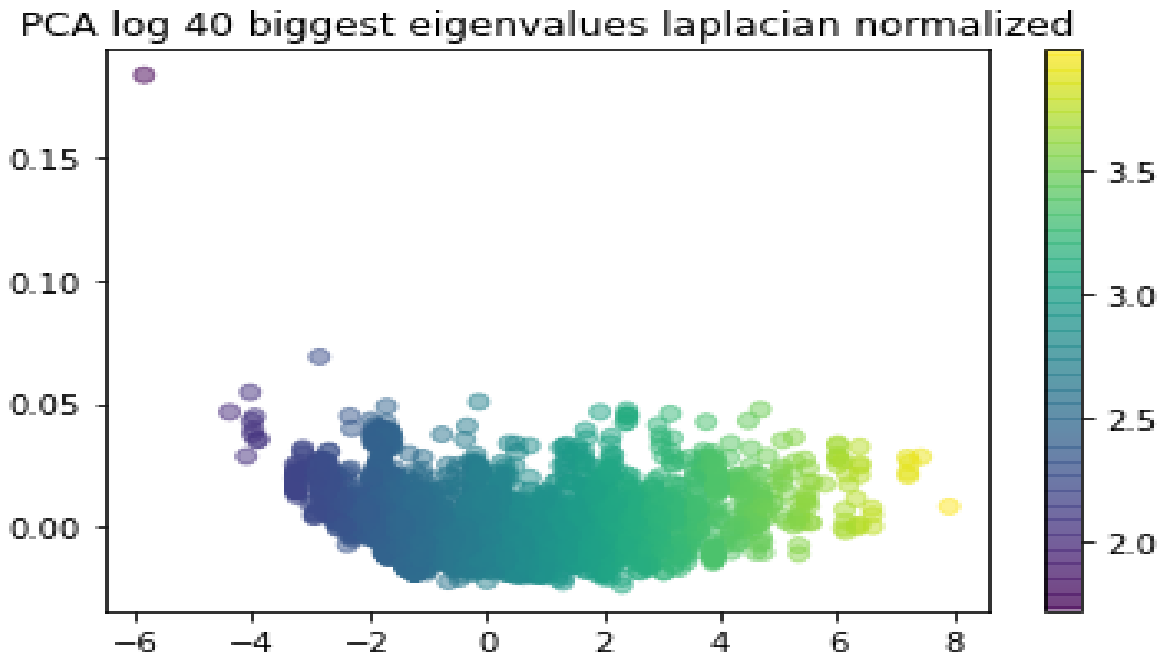


Figure 2: In these figures are represented the scatterplots of the PCA to the 40 smallest eigenvalues of the normalized laplacian ($\mathcal{L}$) of threshold of 8 $\mathring{A}$ of enzymes colored with respect to their EC first (a), taxonomy (b), temperature (c), and average KM (d).Apparently there is no possibility of conducting discriminant classification of any of the different labels. All the classes are infact spread along the figure and mixed without any sharp distinction among them.The explained variance is [0.99 0.01]

(a)



(b)

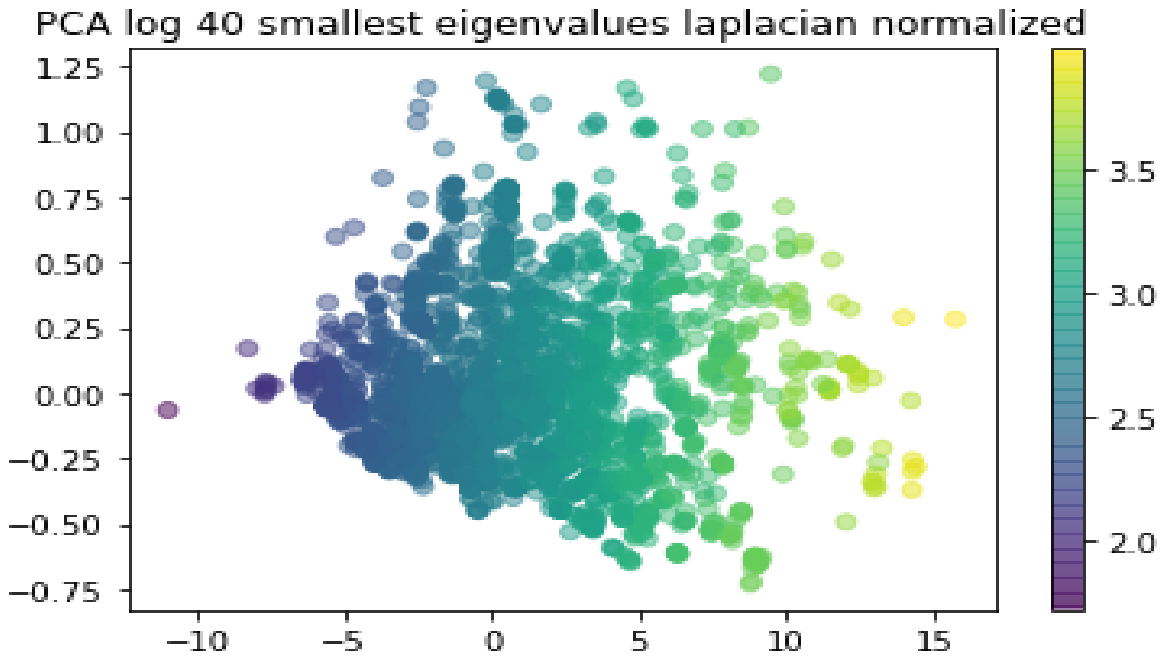
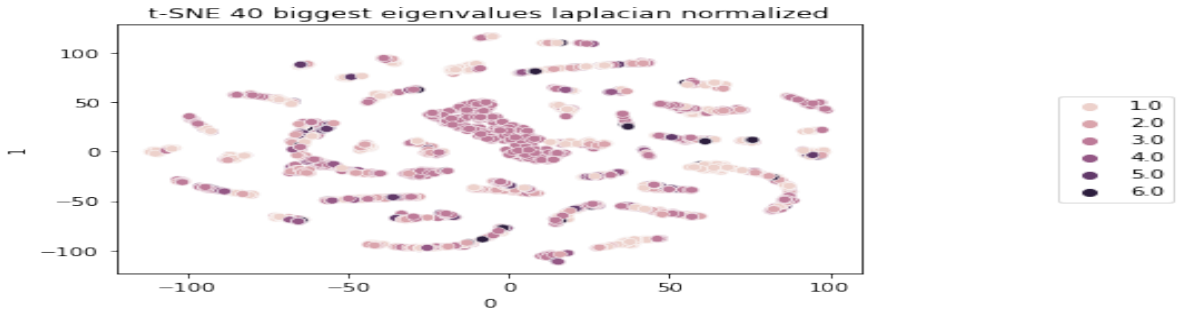
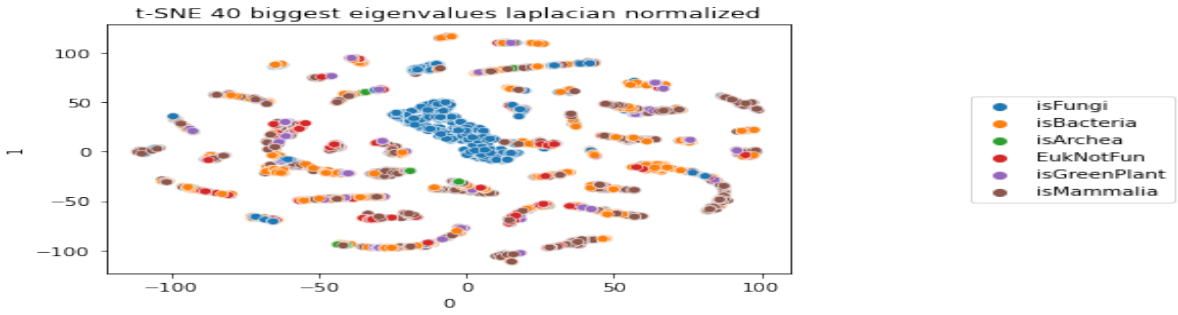


Figure 3: In these figures are represented the scatterplots of the PCA to the log of the 40 biggest eigenvalues (a) and the log of the 40 smallest eigenvalues (b) of the normalized laplacian ($\mathcal{L}$) obtained with threshold $8\mathring{A}$ of enzymes colored with respect to the number of nodes. I have decided to plot the logarithm as it shows better the dependence on the number of nodes rather then without. Infact log is invertible in the domain of the laplacian eigenvalues.In the colormap the scale is logarithmic so that it represents better the color gradient.

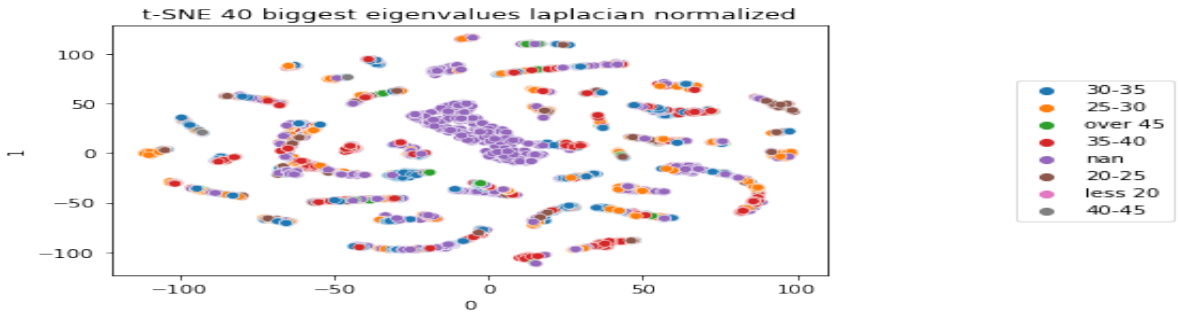
(a)



(b)



(c)



(d)

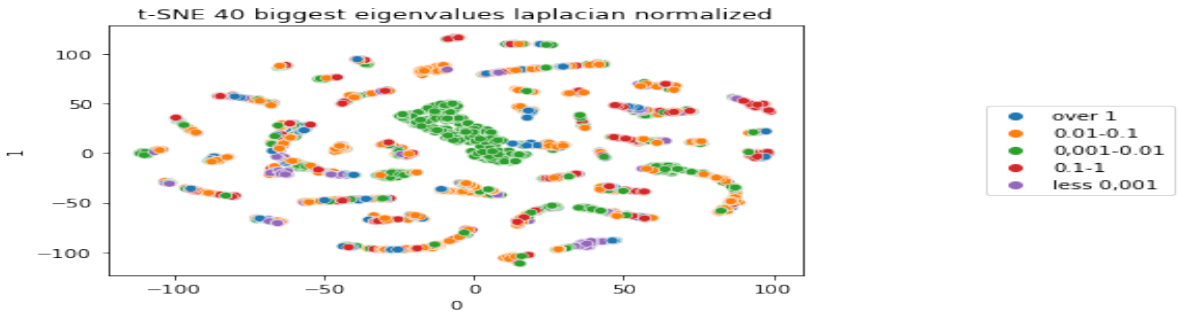
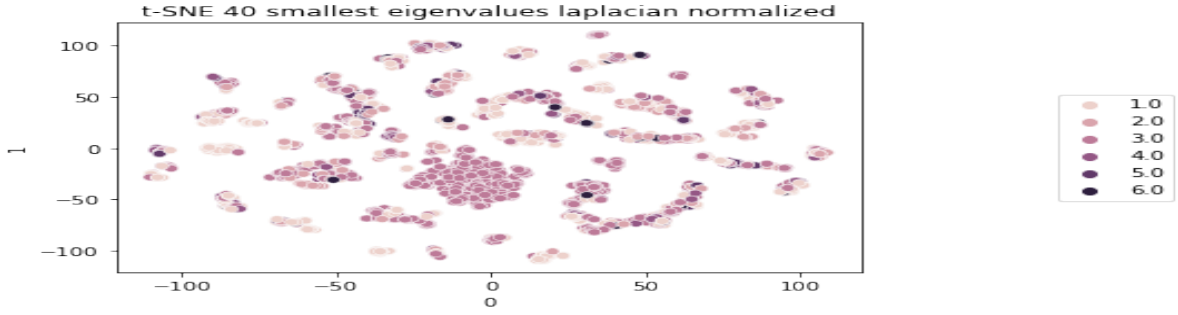
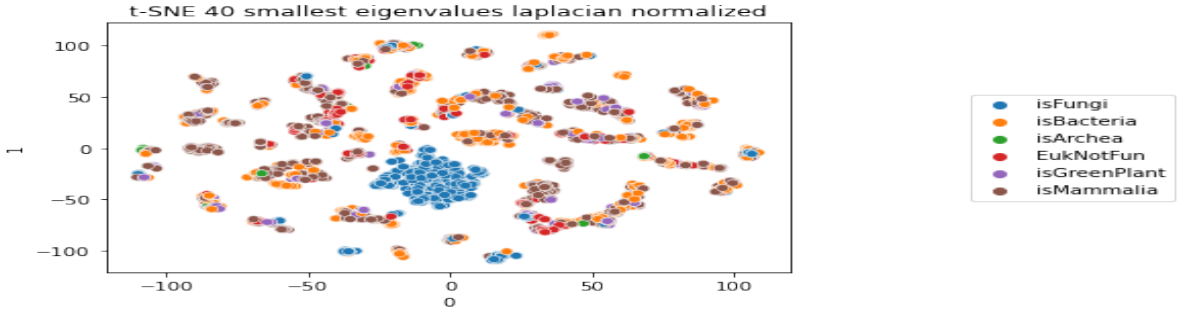


Figure 4: In these figures are represented the tsne of the biggest 40 eigenvalues of the normalized laplacian obtained with threshold 8 $\mathring{A}$ with respect the number of nodes and labeled with respect the EC first (a), taxonomy (b), temperature (c), KM (d) of the enzymes. They are representing many different groups that seem not to strictly qualify the enzymes with any of the labels, however around (0,0) can be seen a cluster of enzymes belonging to the same family, 'P11838'

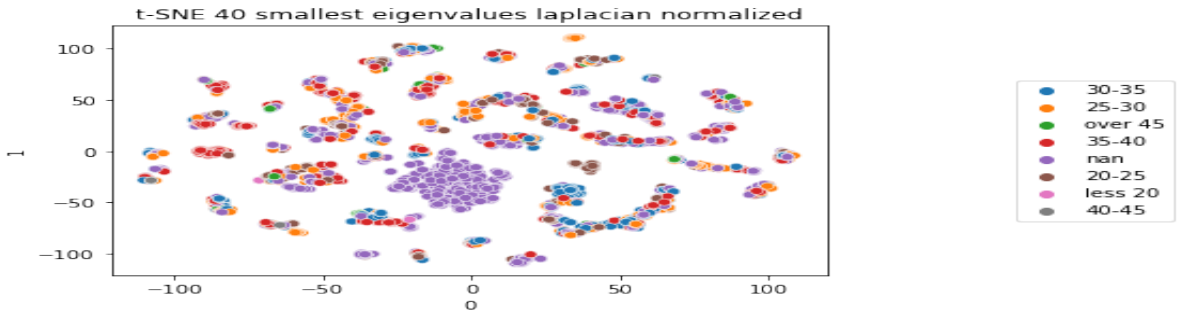
(a)



(b)



(c)



(d)

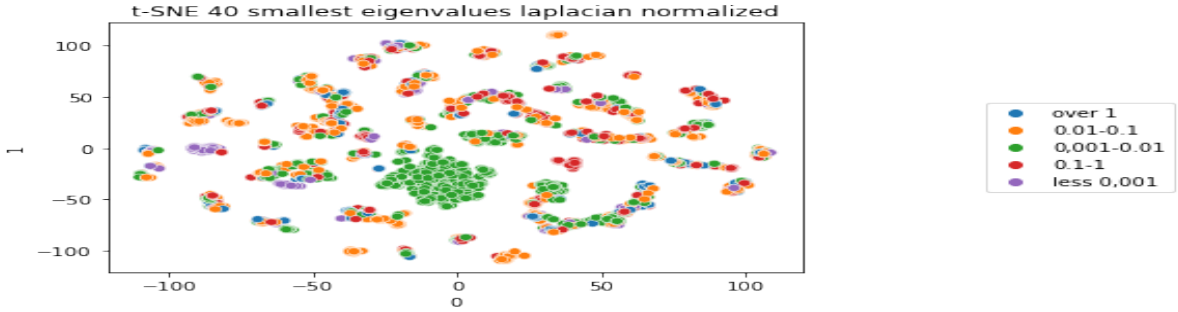
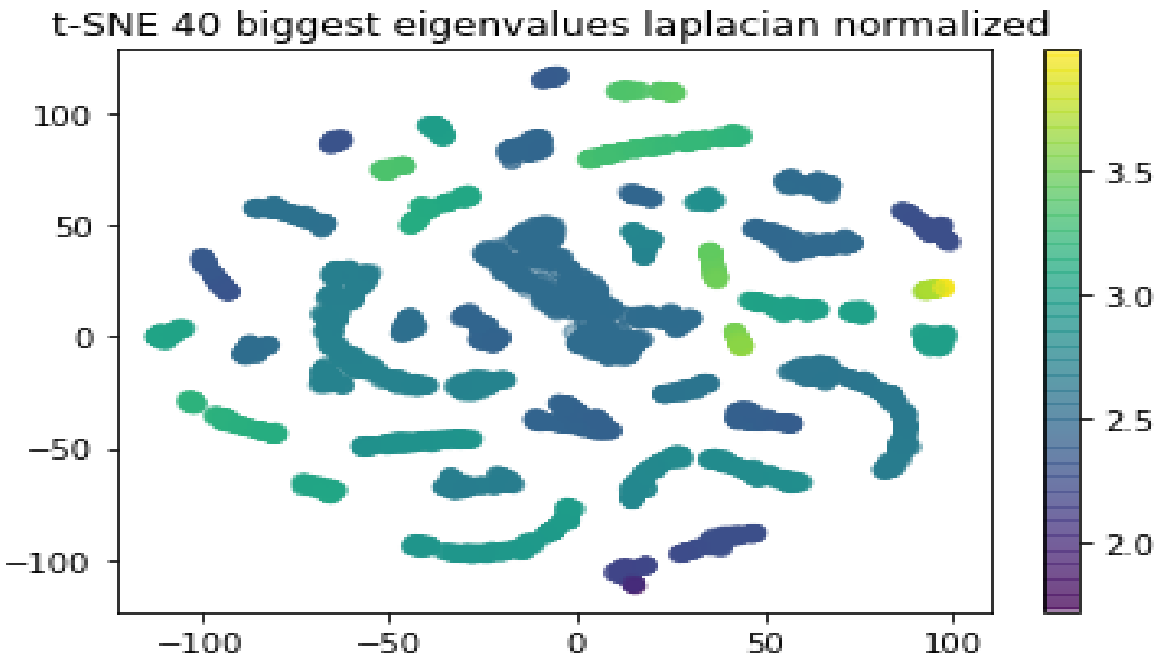


Figure 5: In these figures are represented the tsne of the smallest 40 eigenvalues of the normalized laplacian with threshold 8 $\mathring{A}$ with respect the EC first (a), taxonomy (b), temperature (c) and KM (d) . They are representing many different groups that seem not to strictly qualify the enzymes via any of these labels, however around (0,0) can be seen a cluster of enzymes belonging to the same family, 'P11838'

(a)



(b)

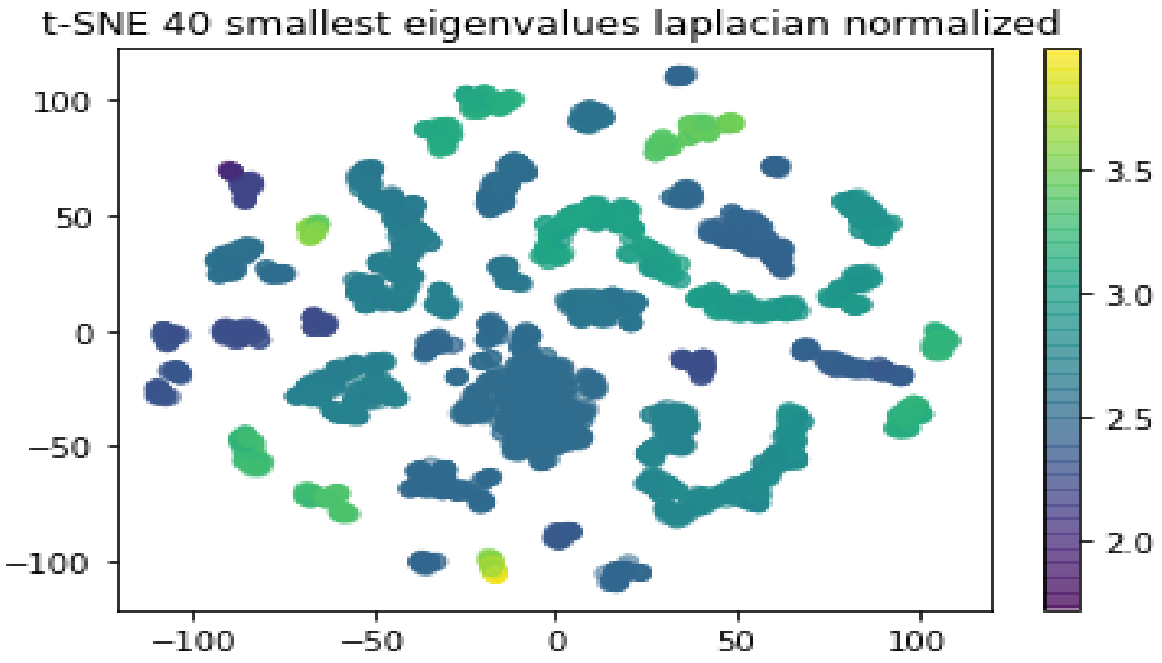


Figure 6: In these figures are represented the tsne of the 40 biggest eigenvalues (a) and 40 smallest eigenvalues (b) of the normalized laplacian $\mathcal{L}$ obtained with threshold 8 $\mathring{A}$ with respect the number of nodes

(a)

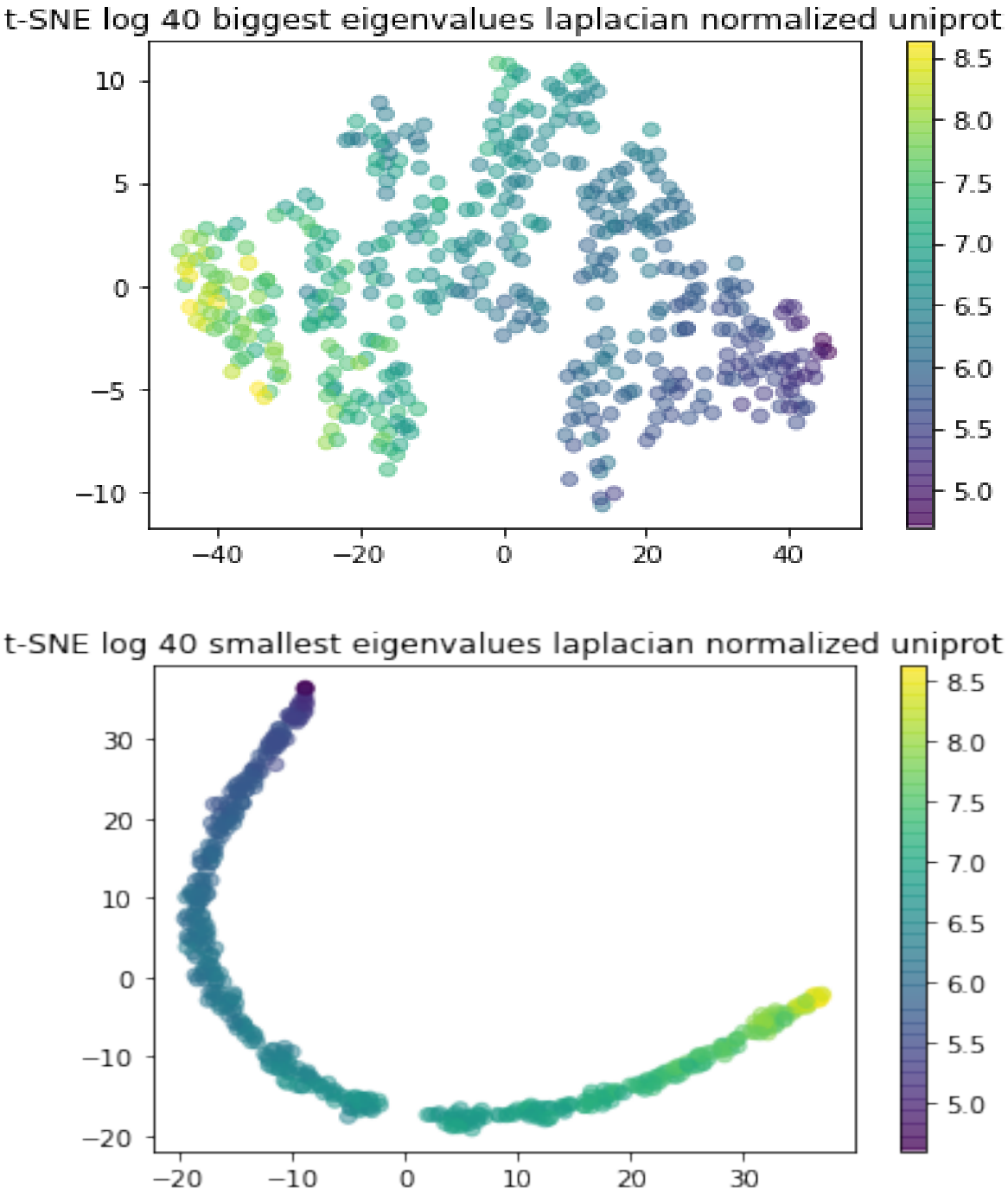


Figure 7: In these figures are represented the t-SNE of the 40 biggest (a) and 40 smallest (b) and smallest normalized (c) eigenvalues of the normalized laplacian with respect to the number of nodes. In each case has been chosen a representant per uniprot. Note that setting nearest neighbors to 10 is not sufficient to produce a unique structure (as in the laplacian) in the biggest eigenvalues, yet a dependence on the number of nodes persists.

(a)

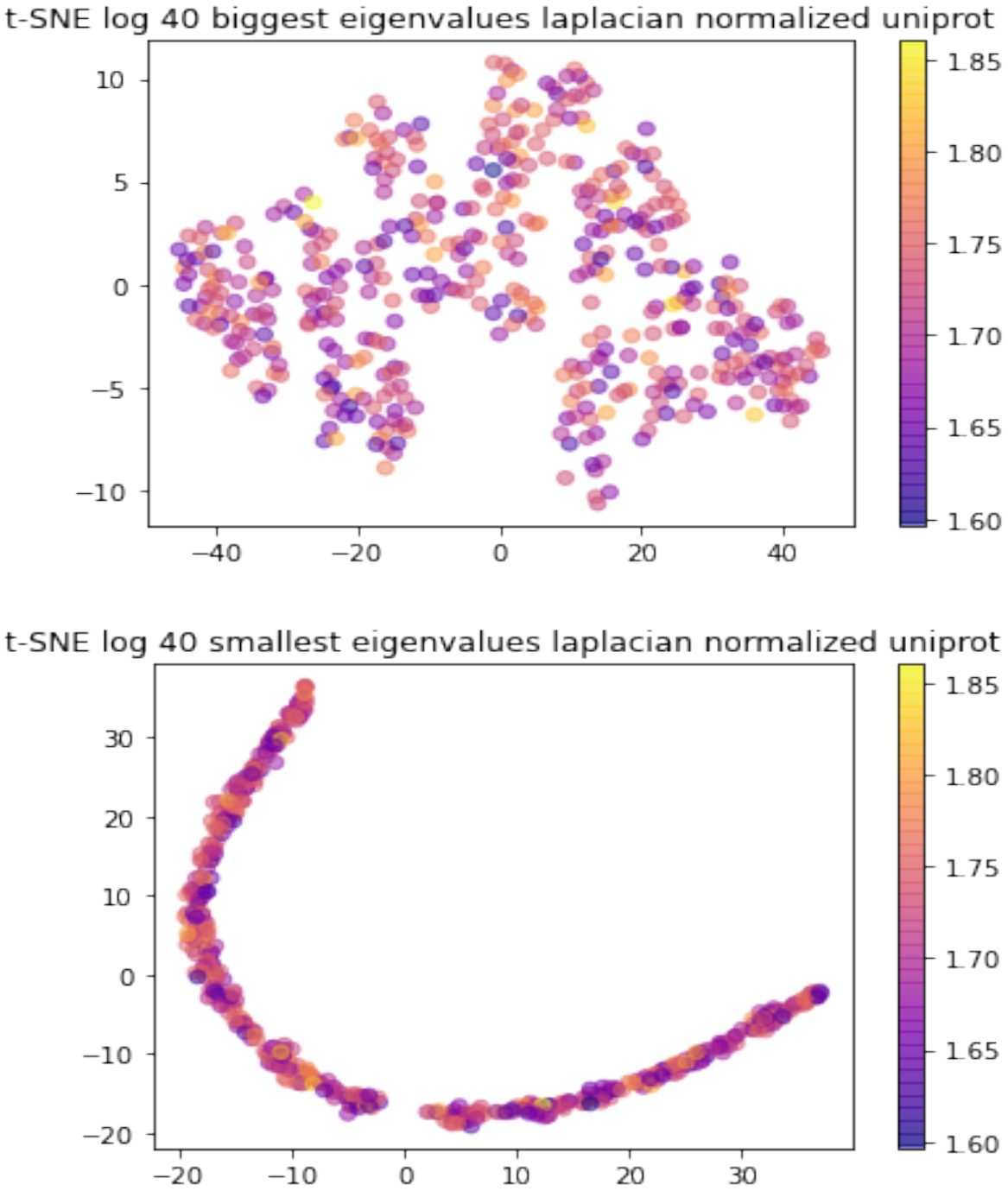
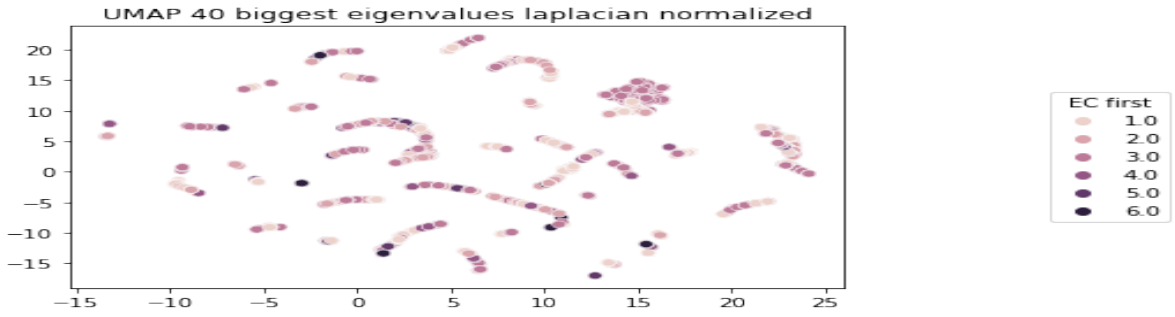
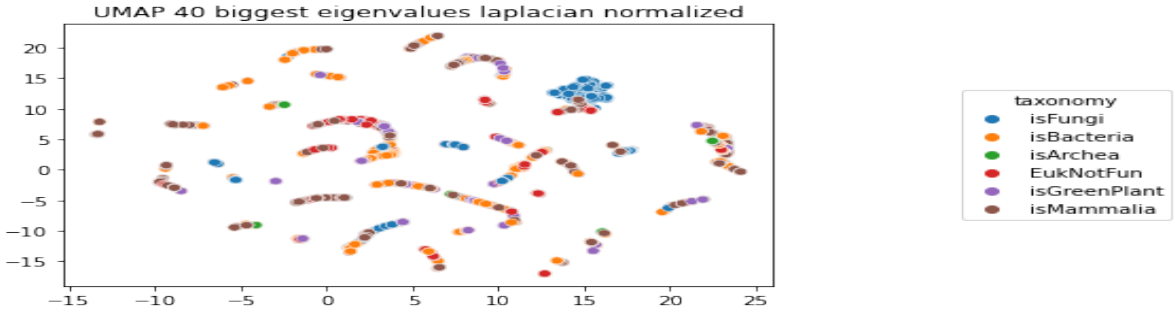


Figure 8: In these figures are represented the t-SNE of the 40 biggest (a) and 40 smallest (b) and smallest normalized (c) eigenvalues of the normalized laplacian with respect to link density. In each case has been chosen a representant per uniprot

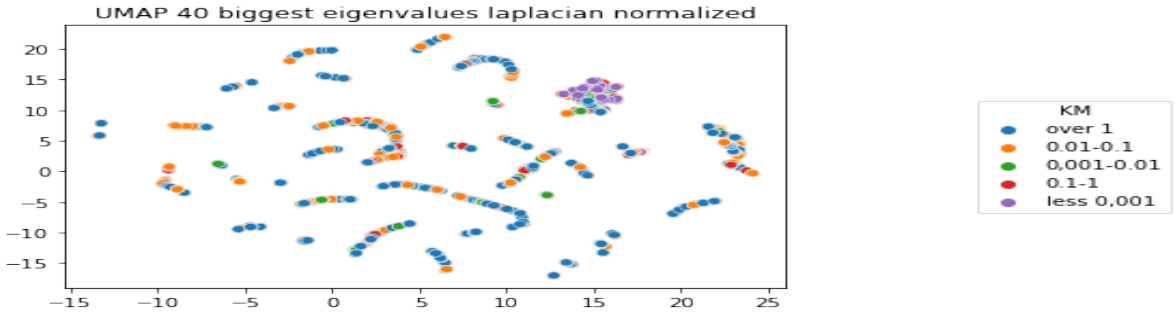
(a)



(b)



(c)



(d)

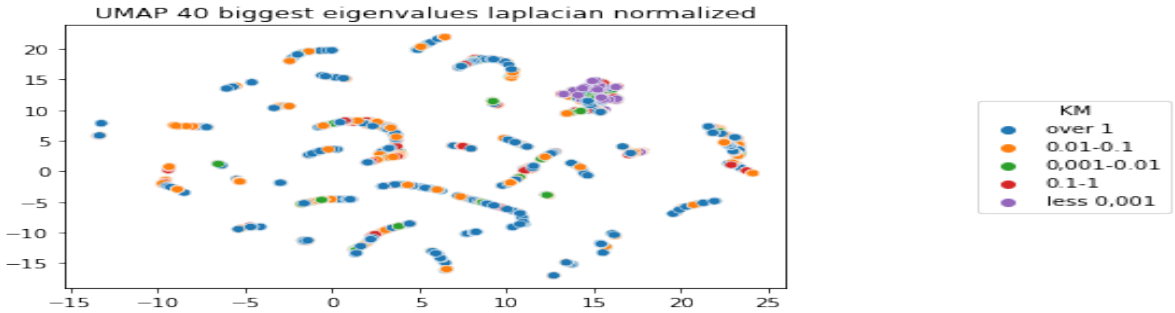
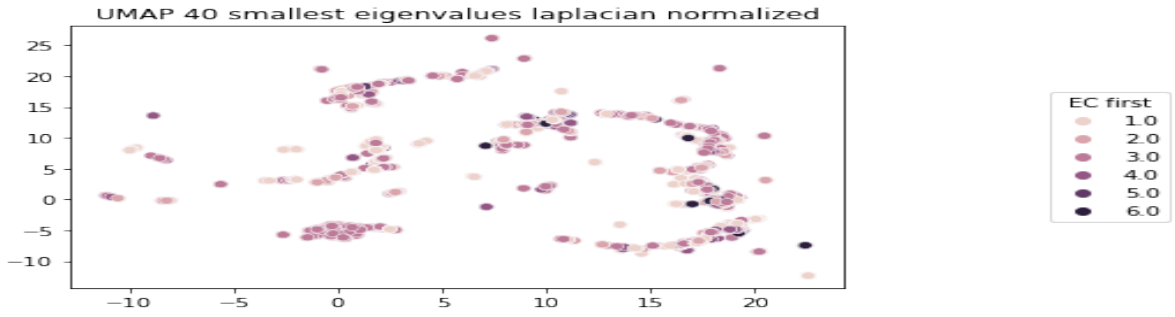
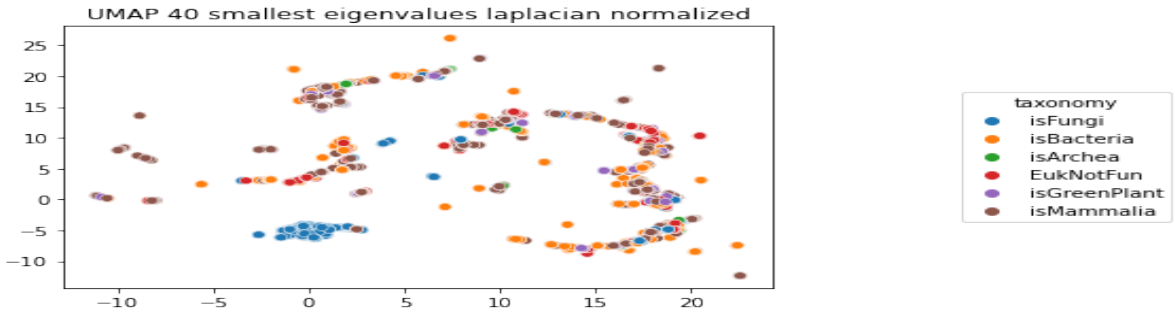


Figure 9: In these figures are represented the UMAP analysis of the 40 biggest eigenvalues of the normalized laplacian $\mathcal{L}$ with threshold 8 $\mathring{A}$ and labeled EC first (a), taxonomy (b), temperature (c) and KM (d). The parameter of nearest neighbors is set to 5. They are representing different groups (less than in the case of t-SNE) that seem not to strictly qualify the enzymes via any of the labels, however around (15,15) can be seen a cluster of enzymes belonging to the same family, 'P11838'

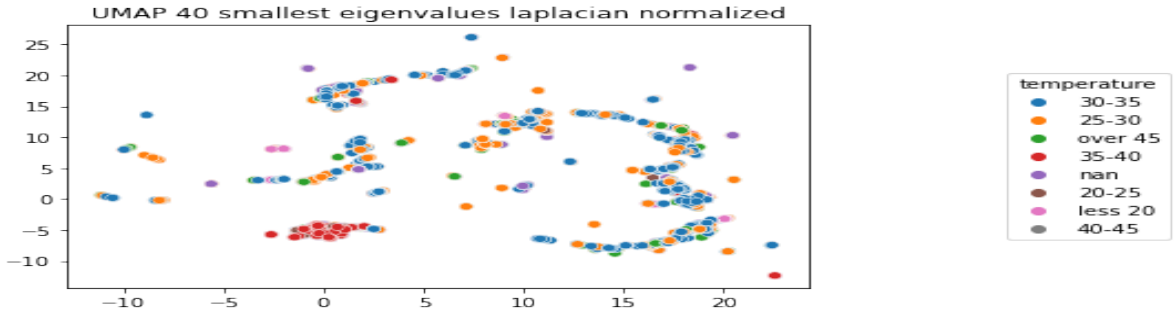
(a)



(b)



(c)



(d)

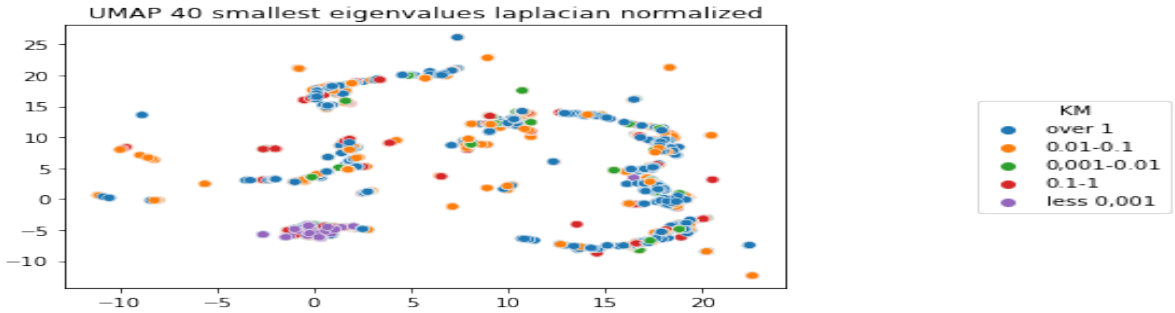
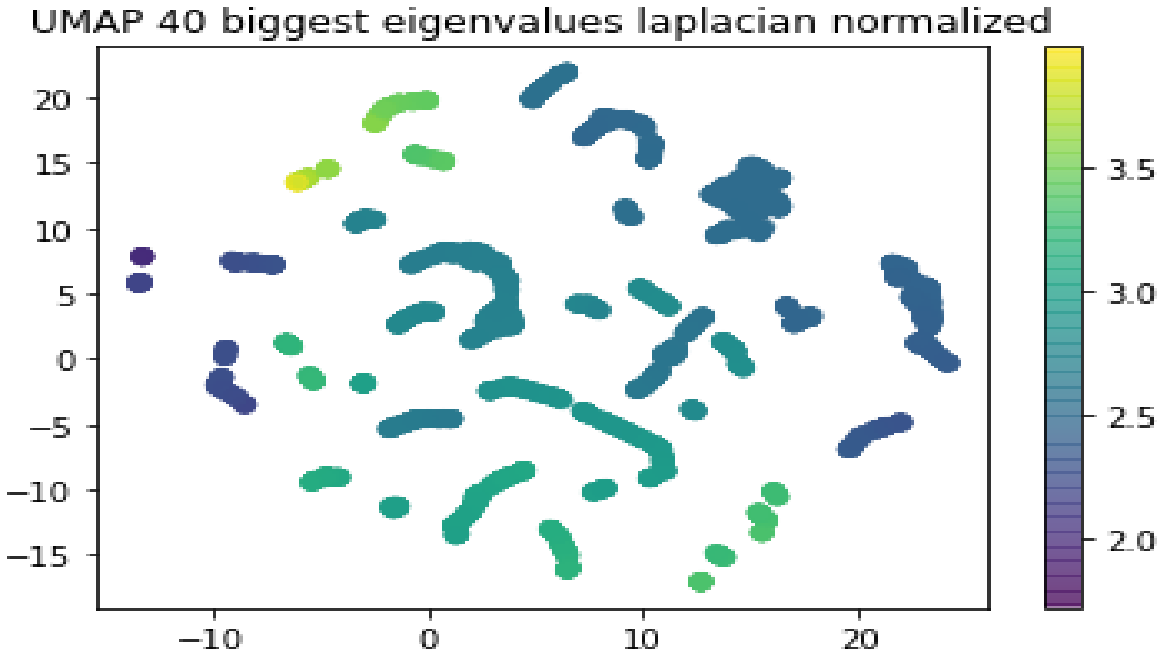


Figure 10: In these figures are represented the UMAP analysis of the 40 smallest eigenvalues of the normalized laplacian $\mathcal{L}$ and labeled by EC first (a), taxonomy (b), temperature (c) and KM (d). The parameter of nearest neighbors is set to 5. They are represented different groups (less than in the case of t-SNE) that seem not to strictly qualify the enzymes via any of these labels, however around (-5,0) can be seen a cluster of enzymes belonging to the same family, 'P11838'

(a)



(b)

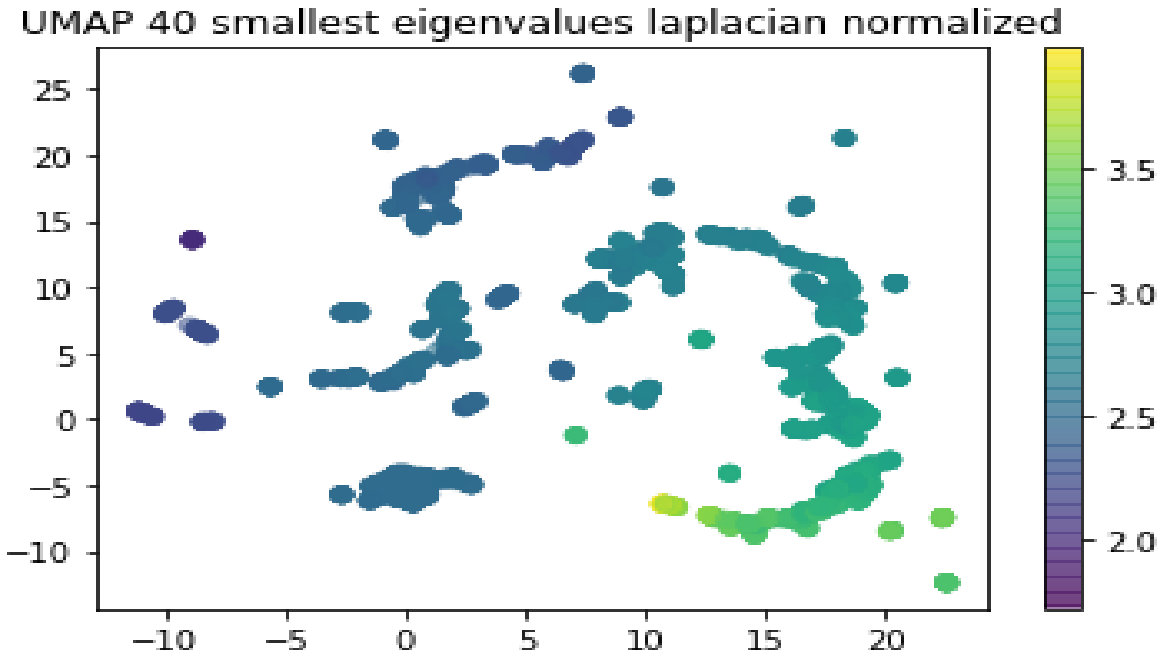
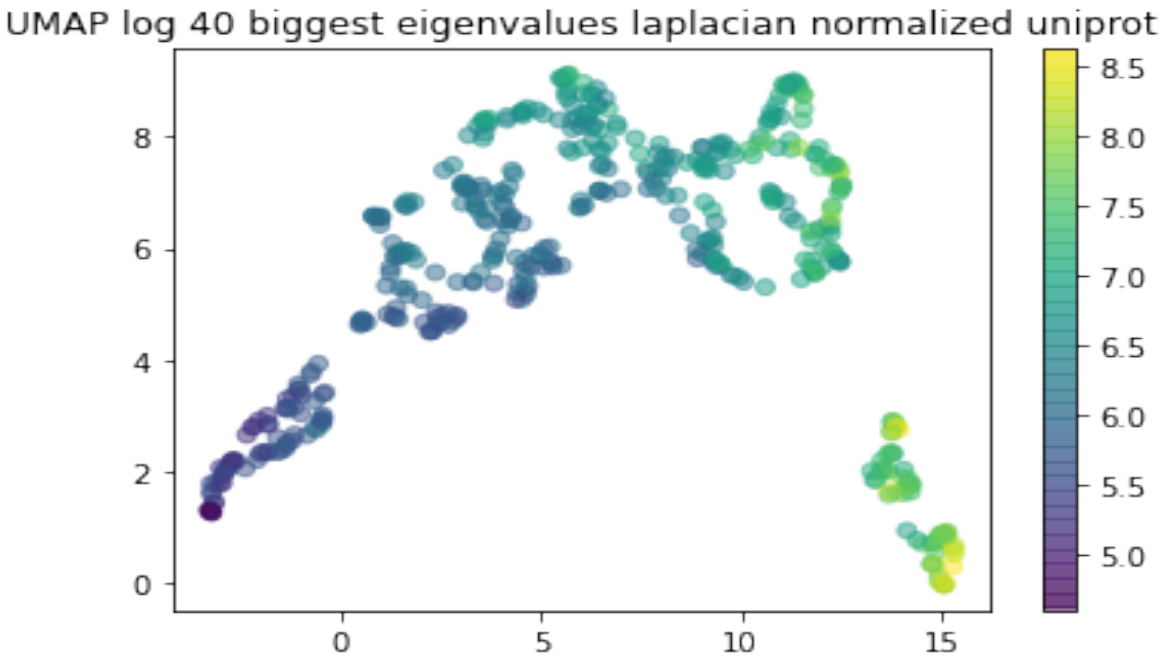


Figure 11: In these figures are represented the UMAP analysis of the 40 biggest (a) and smallest (b) eigenvalues of the normalized laplacian $\mathcal{L}$ obtained with threshold 8 $\mathring{A}$ and labeled by the number of nodes represented in the log scale in the color map. The parameter of nearest neighbors is set to 10

(a)



(b)

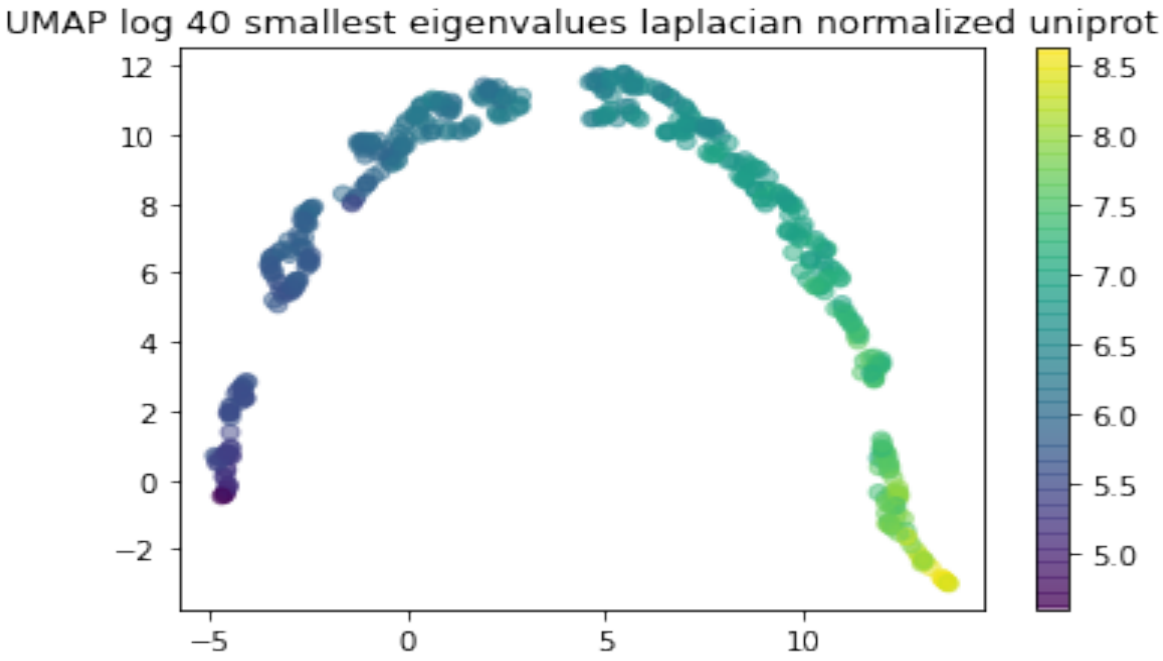
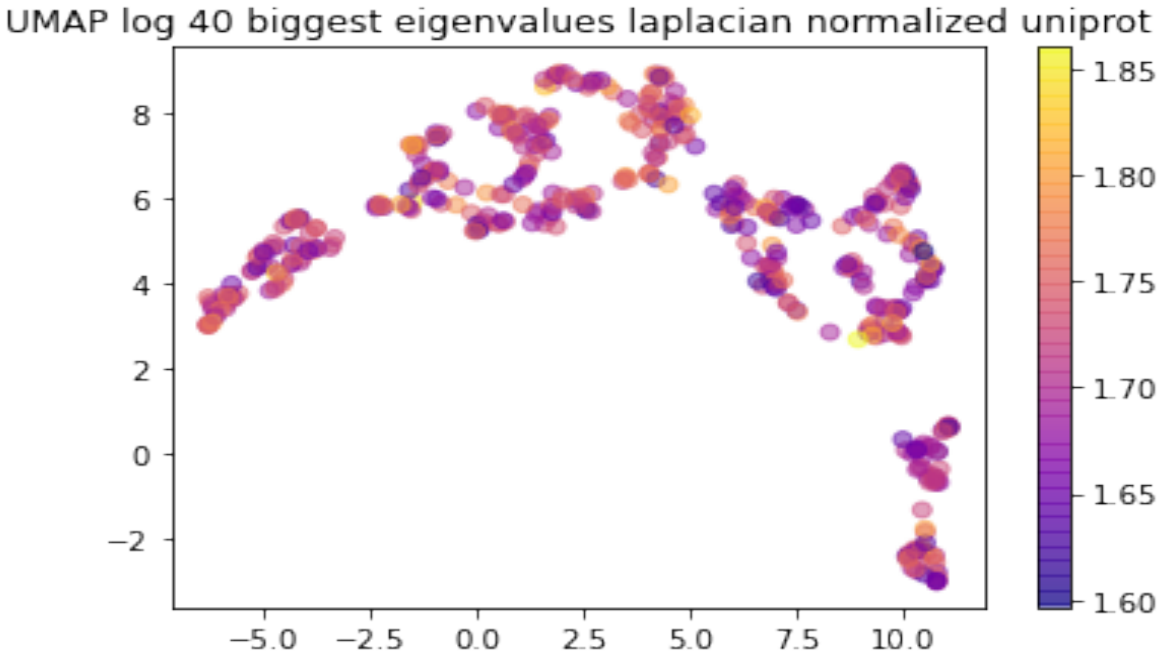


Figure 12: In these figures are represented the UMAP analysis of the 40 biggest (a) and smallest (b) eigenvalues of the normalized laplacian $\mathcal{L}$ obtained with threshold 8 $\mathring{A}$ and labeled by the number of nodes represented in the log scale in the color map. Here is selected a structure for each uniprot. The parameter of nearest neighbors is set to 10. It seems a general fact, that the normalized laplacian produces more clusters compared to the laplacian.

(a)



(b)

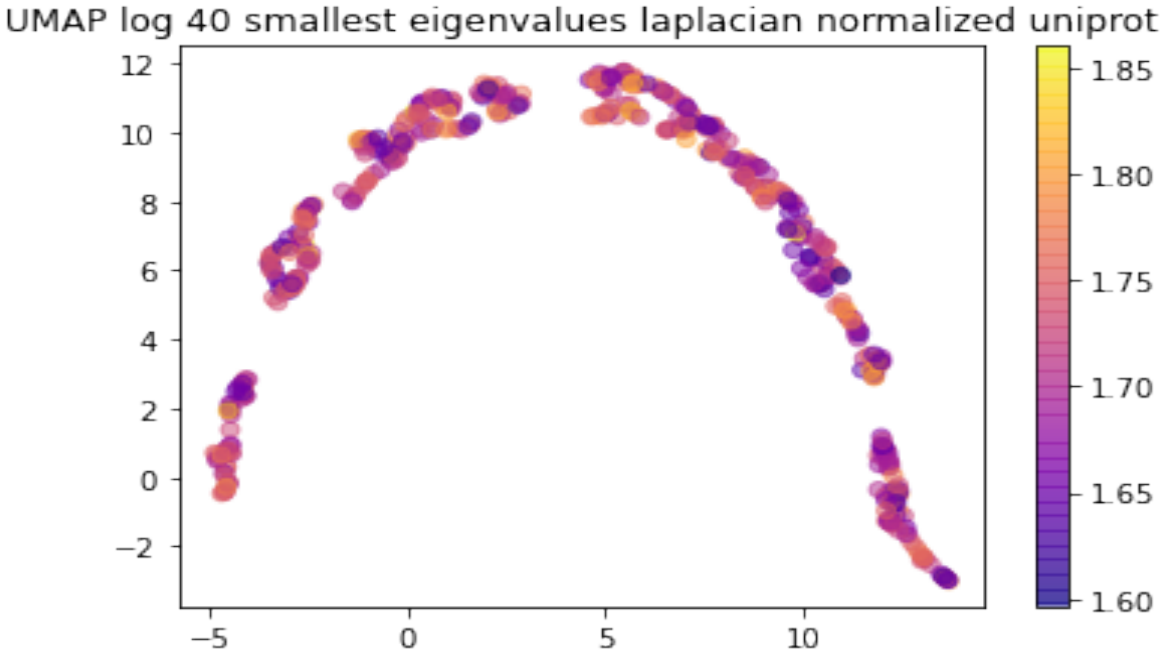
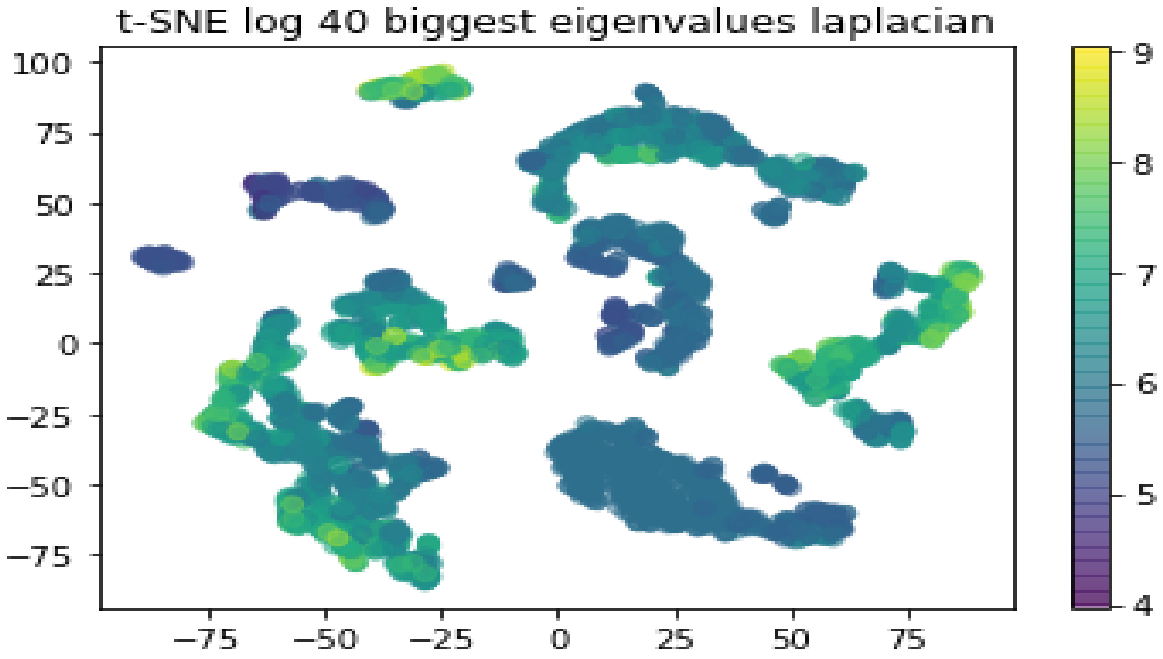


Figure 13: In these figures are represented the UMAP analysis of the 40 biggest (a) and smallest (b) eigenvalues of the normalized laplacian $\mathcal{L}$ obtained with threshold 8 $\mathring{A}$ and labeled by the link density . Here is selected a structure for each uniprot. The parameter of nearest neighbors is set to 10

(a)



(b)

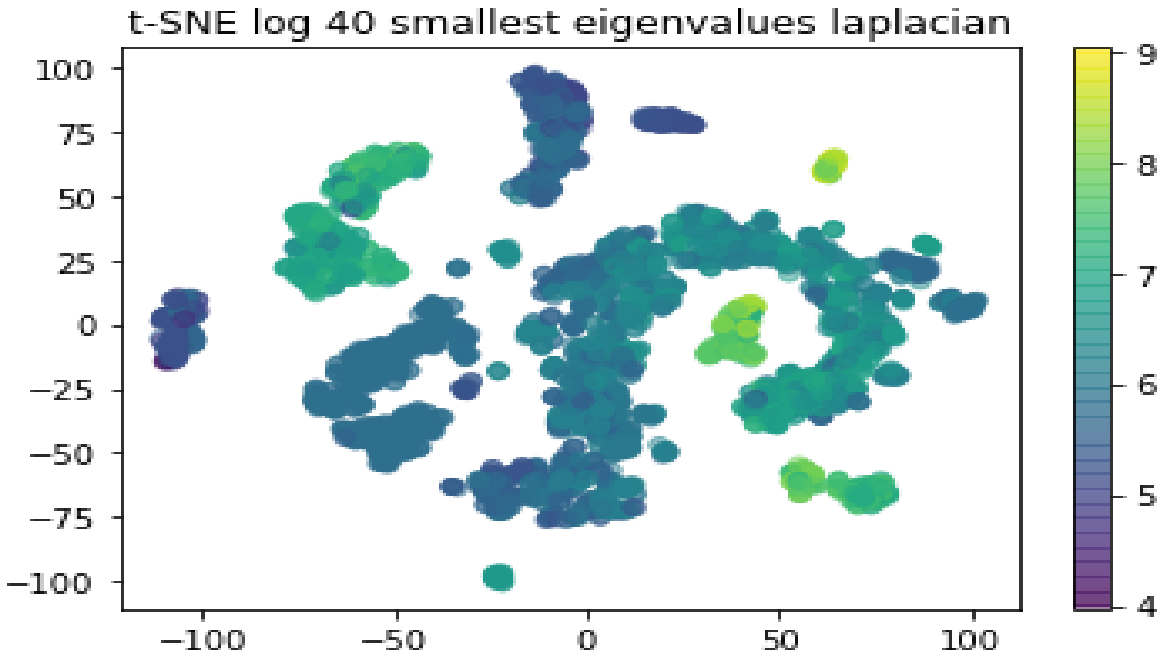
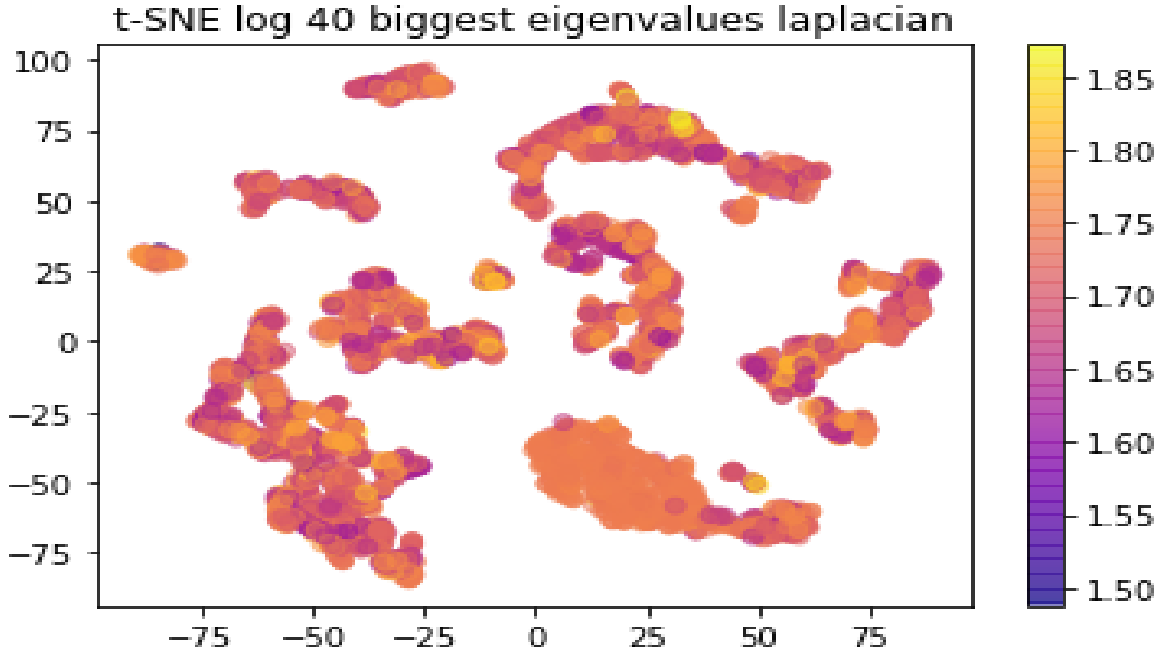


Figure 14: In these figures are represented the t-SNE analysis of the 40 biggest (a) and smallest (b) eigenvalues of the laplacian L obtained with threshold 12 $\AA$ and labeled by the number of nodes represented in the log scale in the color map. The parameter of nearest neighbors is set to 10

(a)



(b)

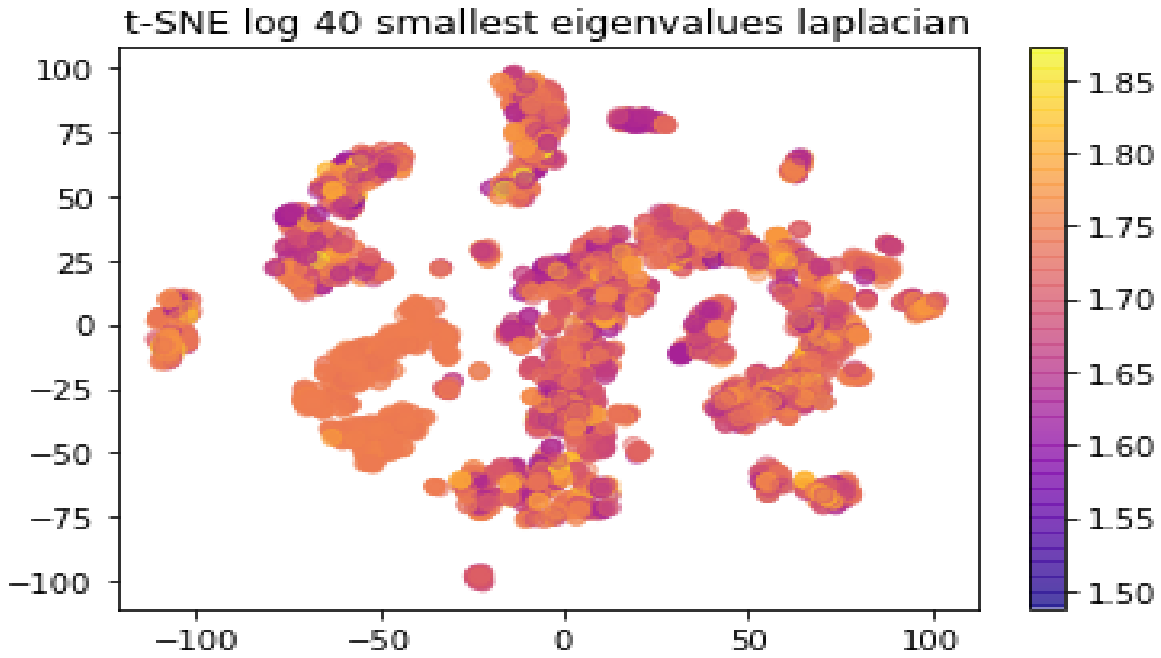
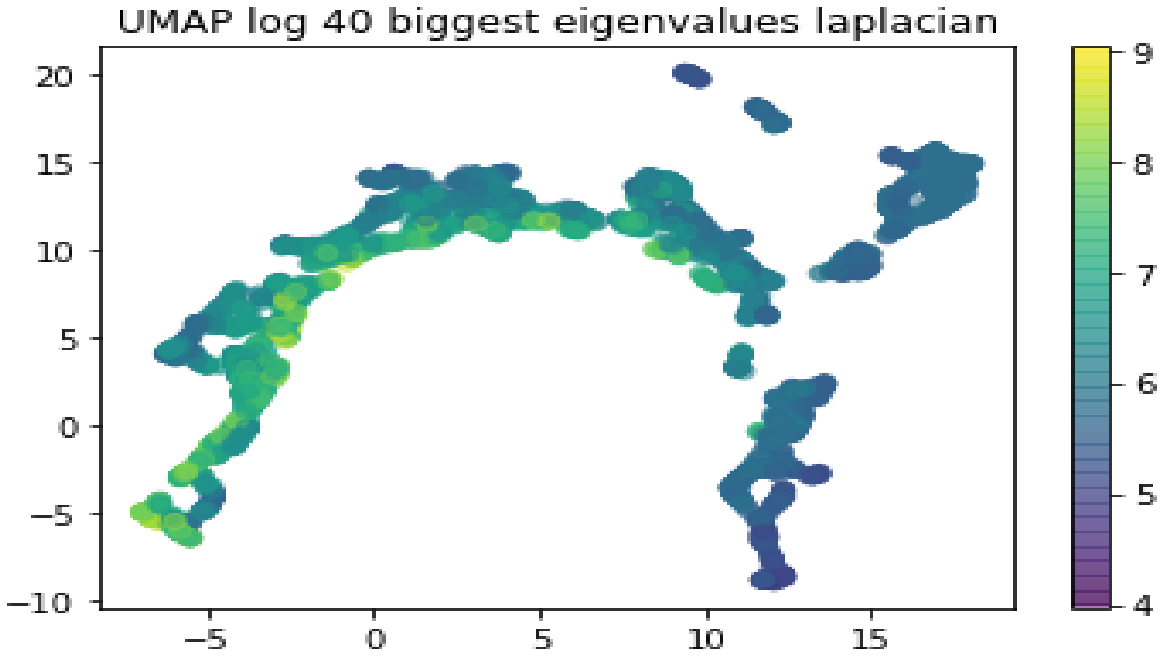


Figure 15: In these figures are represented the t-SNE analysis of the 40 biggest (a) and smallest (b) eigenvalues of the laplacian L obtained with threshold 12 $\AA$ and labeled by the link density represented in the log scale in the color map. The parameter of nearest neighbors is set to 10

(a)



(b)

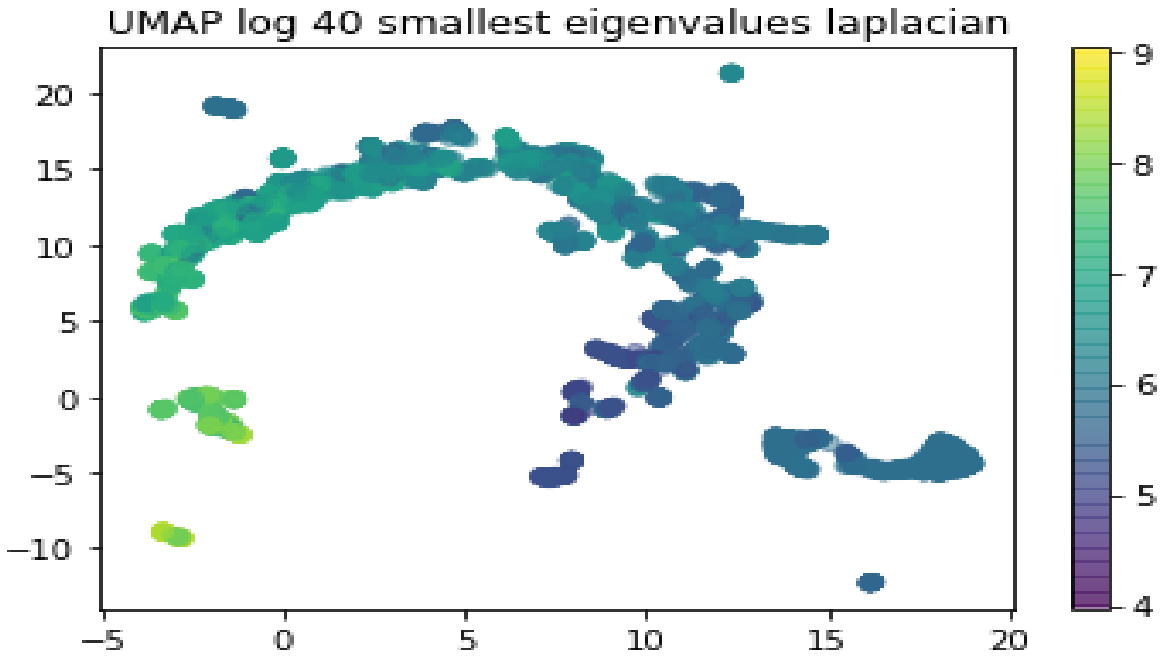
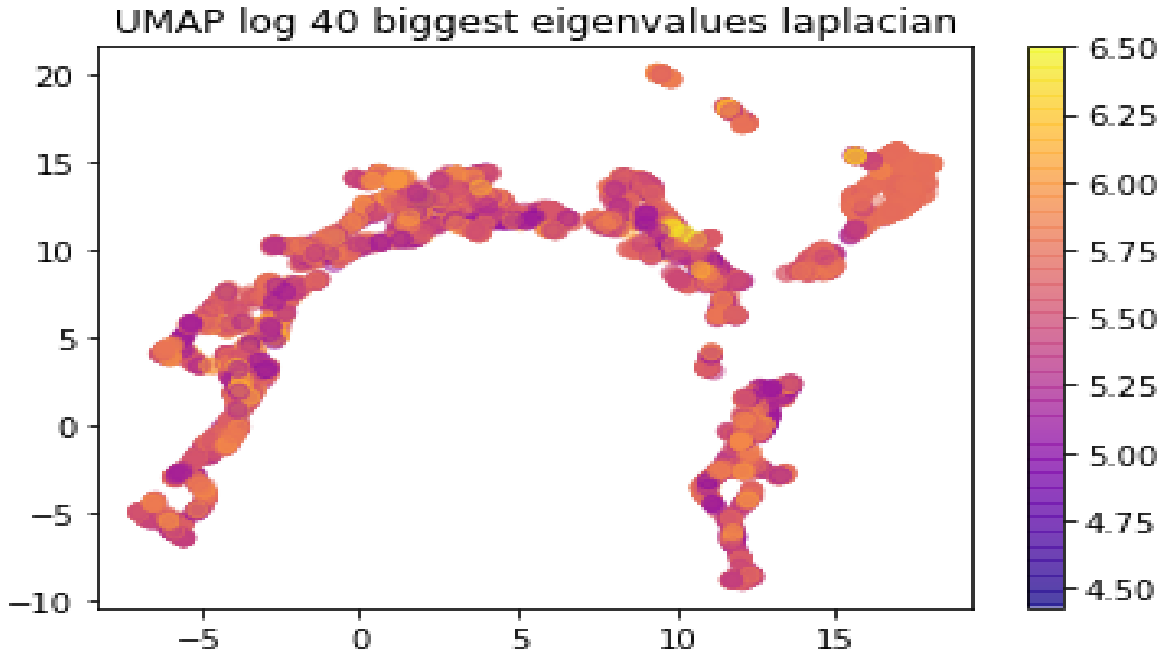


Figure 16: In these figures are represented the UMAP analysis of the 40 biggest (a) and smallest (b) eigenvalues of the laplacian L obtained with threshold 12 $\AA$ and labeled by the number of nodes represented in the log scale in the color map. The parameter of nearest neighbors is set to 10

(a)



(b)

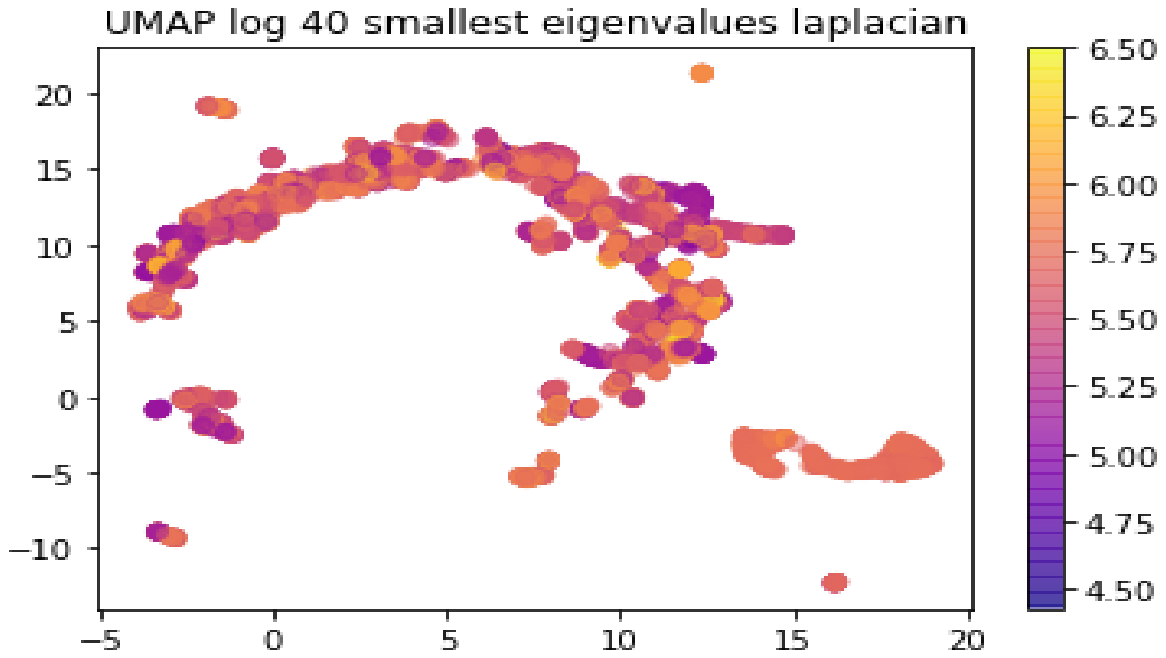
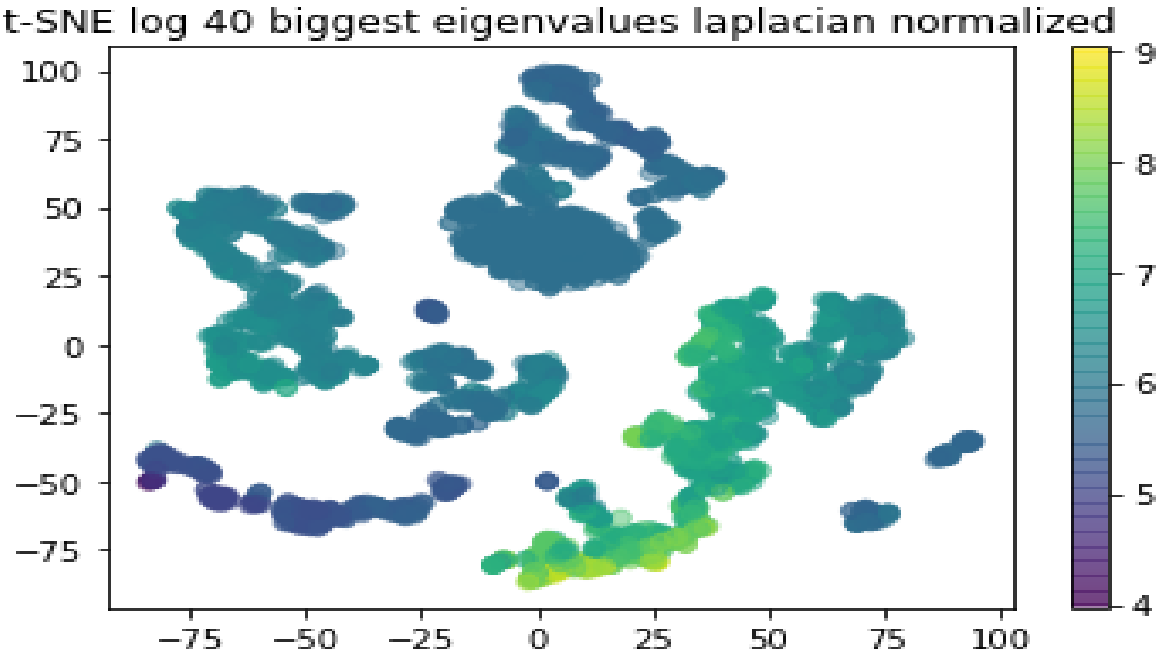


Figure 17: In these figures are represented the UMAP analysis of the 40 biggest (a) and smallest (b) eigenvalues of the laplacian L obtained with threshold 12 $\AA$ and labeled by the link density represented in the log scale in the color map. The parameter of nearest neighbors is set to 10

(a)



(b)

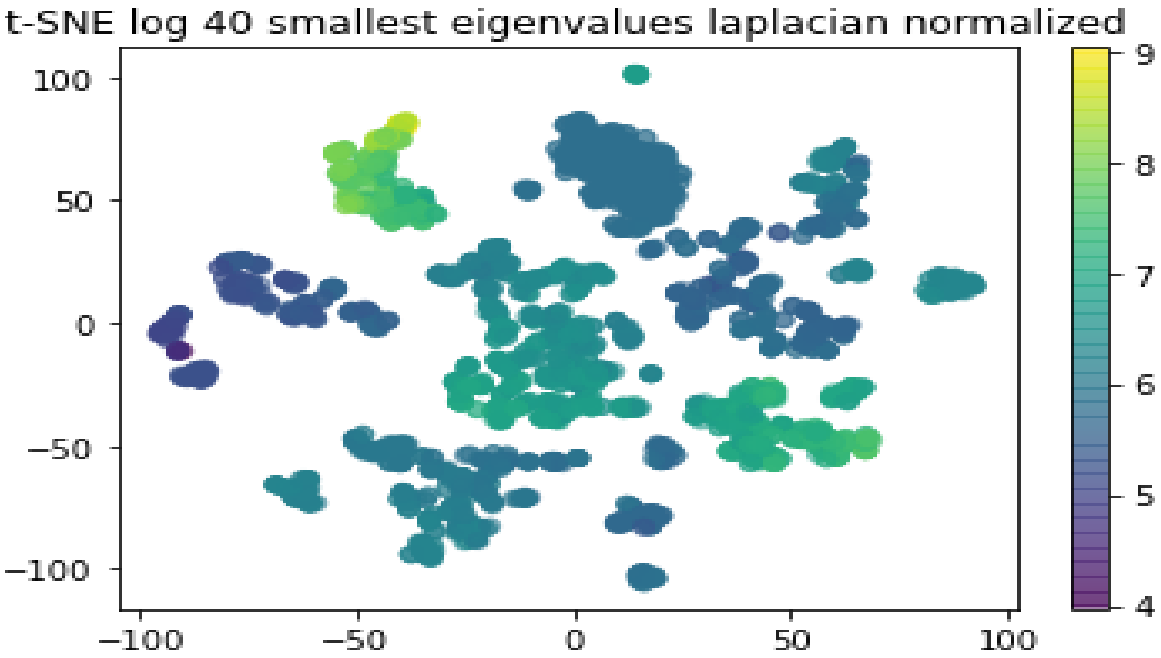
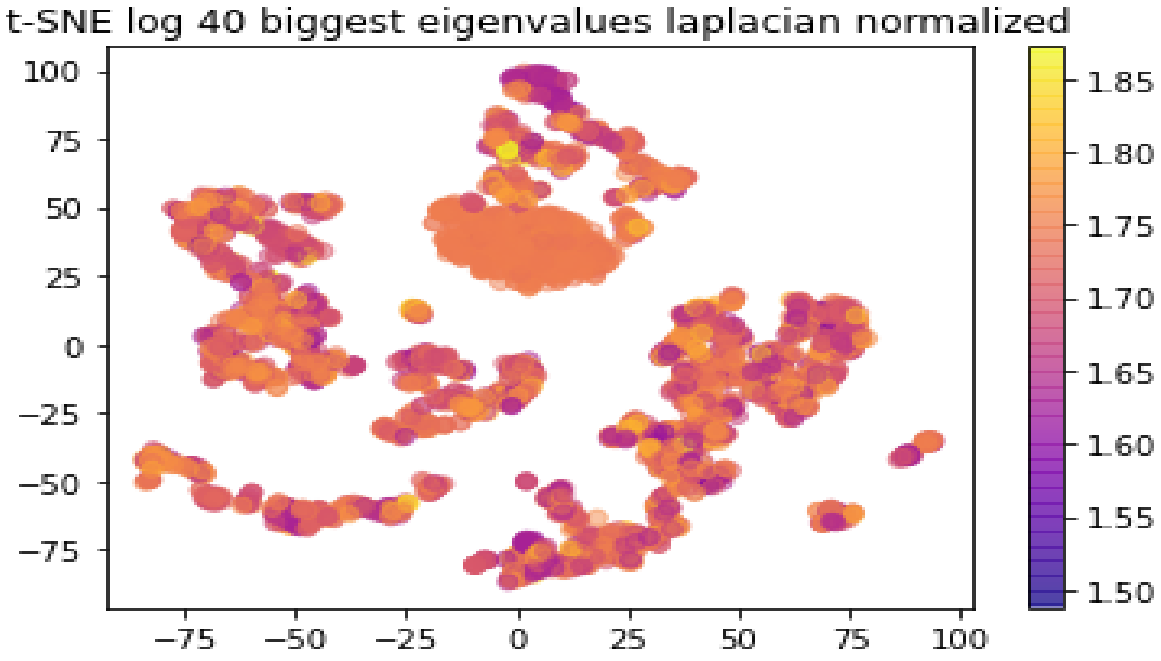


Figure 18: In these figures are represented the t-SNE analysis of the 40 biggest (a) and smallest (b) eigenvalues of the normalized laplacian $\mathcal{L}$ obtained with threshold 12 $\mathring{A}$ and labeled by the number of nodes represented in the log scale in the color map. The parameter of nearest neighbors is set to 10

(a)



(b)

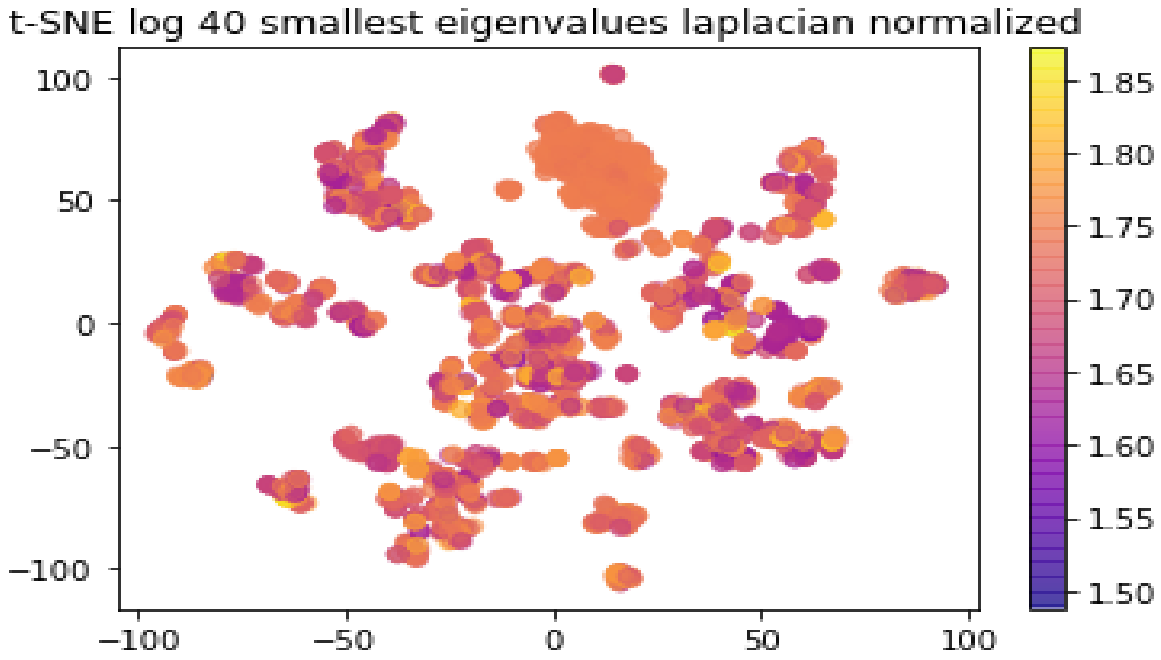
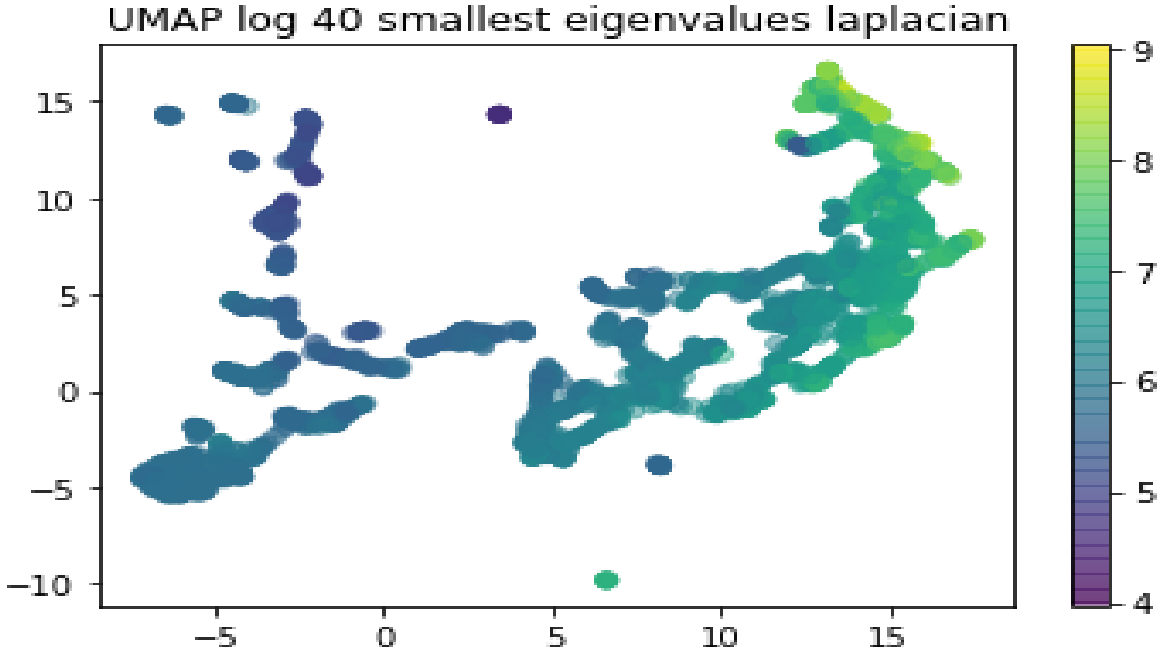


Figure 19: In these figures are represented the t-SNE analysis of the 40 biggest (a) and smallest (b) eigenvalues of the normalized laplacian $\mathcal{L}$ obtained with threshold 12 $\mathring{A}$ and labeled by the link density represented in the log scale in the color map. The parameter of nearest neighbors is set to 10

(a)



(b)

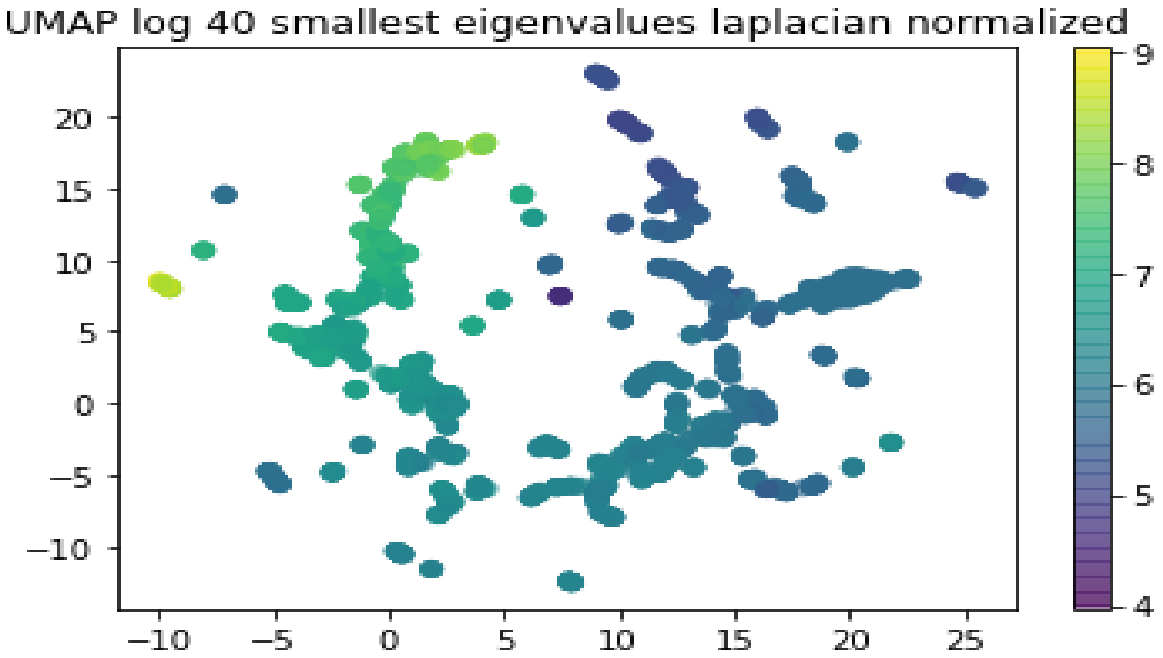
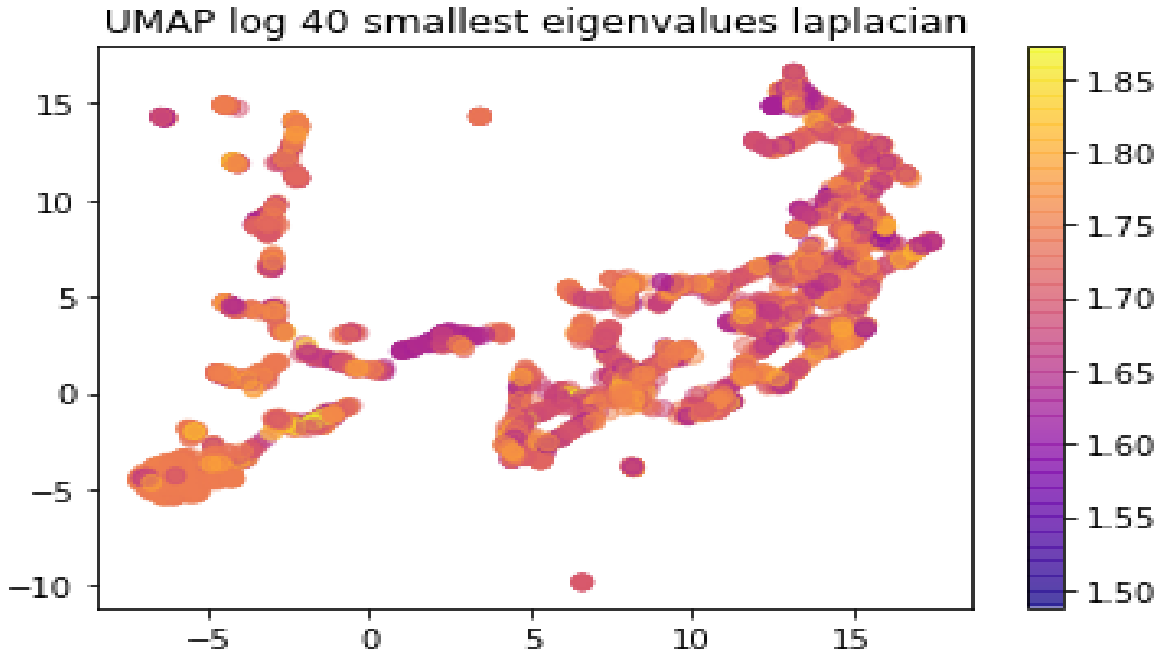


Figure 20: In these figures are represented the UMAP analysis of the 40 biggest (a) and smallest (b) eigenvalues of the normalized laplacian $\mathcal{L}$ obtained with threshold 12 $\mathring{A}$ and labeled by the number of nodes represented in the log scale in the color map. The parameter of nearest neighbors is set to 10

(a)



(b)

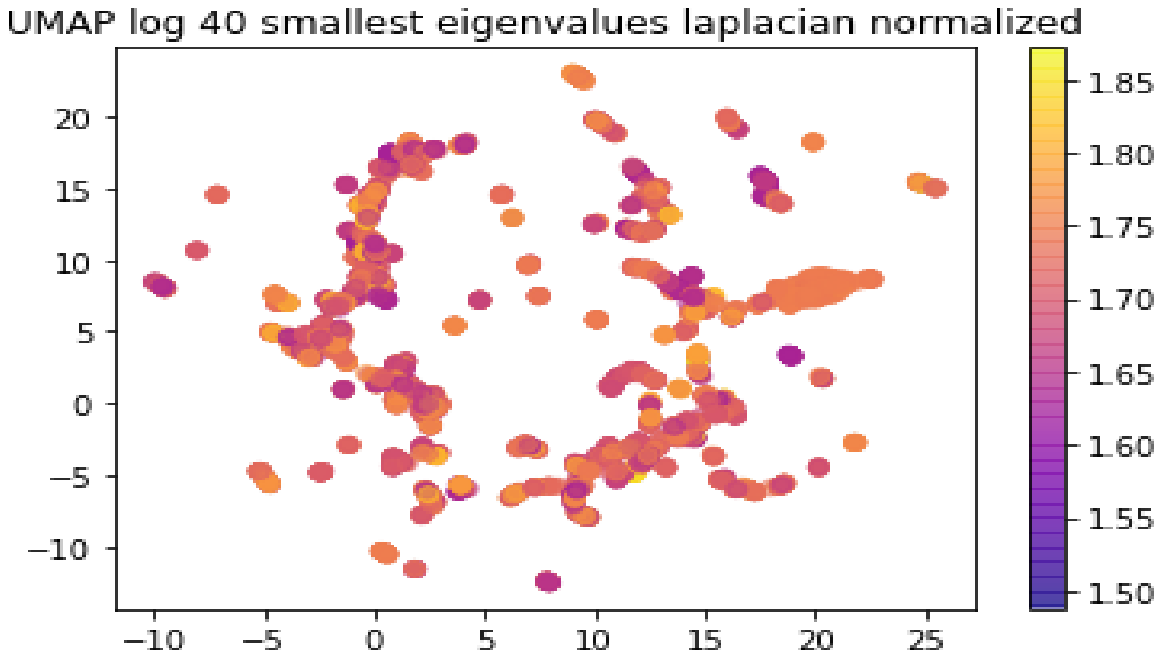


Figure 21: In these figures are represented the UMAP analysis of the 40 biggest (a) and smallest (b) eigenvalues of the normalized laplacian $\mathcal{L}$ obtained with threshold 12 $\mathring{A}$ and labeled by the link density represented in the log scale in the color map. The parameter of nearest neighbors is set to 10

Bibliography

[1] https://en.wikipedia.org/wiki/EnzymeCommissionnumber

[2] https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/methods-for-determining-structure

[3] Havel, T.F., Kuntz, I.D. Crippen, G.M. The theory and practice of distance geometry. Bltn Mathcal Biology 45, 665–720 (1983). https://doi.org/10.1007/BF02460044

[4] Belkin, M., Niyogi, P. Semi-Supervised Learning on Riemannian Manifolds. Machine Learning 56, 209–239 (2004). https://doi.org/10.1023/B:MACH.0000033120.25363.1e

[5] van der Maaten, L.J.P.; Hinton, G.E. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9:2579-2605, 2008.

[6] The art of using t-SNE for single-cell transcriptomics: Dmitry Kobak Philipp Berens; https://doi.org/10.1038/s41467-019-13056-x

[7] http://web.mit.edu/cocosci/Papers/sci_reprint.pdf

[8] UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction: Leland McInnes,John Healy,James Melville;September 21, 2020

[9] arXiv:1802.03426v3 [stat.ML] 18 Sep 2020

[10] https://papers.nips.cc/paper/2001/file/f106b7f99d2cb30c3db1c3cc0fde9ccb-Paper.pdf

[11] https://lvdmaaten.github.io/publications/papers/JMLR_2014.pdf

[12] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9086514, The Classification of Enzymes by Deep Learning: ZHIYU TAO , BENZHI DONG, ZHIXIA TENG , AND YUMING ZHAO

[13] https://www2.imm.dtu.dk/projects/manifold/Papers/Laplacian.pdf

[14] Chung 1997: graph spectral theory

[15] Wikipedia page of embedding problem

[16] Self-similarity of complex networks: Chaoming Song , Shlomo Havlin , and Hernan A. Makse

[17] Milgram, S. Psychol. Today 2, 60 (1967).

[18] Statistical mechanics of complex networks: Reka Albert and Albert-Laszlo Barabasi

[19] The Laplacian eigenvalues of graphs: a survey: Xiao-Dong Zhang, arXiv:1111.2897

[20] B. Mohar, The Laplacian spectrum of graphs, in Y. Alavi et al. (Eds.), Graph Theory, Combinatorics, and Applications, Vol.2, pp.871-898, Wiley, New York, 1991.

[21] 3D genome reconstruction from chromosomal contacts: Annick Lesne, Julien Riposo, Paul Roger, Axel Cournac Julien Mozziconacci

[22] The structure and function of complex networks: M. E. J. Newman

[23] Network-based strategies for protein characterization Alessandra Merlotti, Giulia Menichetti, Piero Fariselli, Emidio Capriotti, Daniel Remondini

[24] Biyikoglu, T., et al. (2007). Laplacian eigenvectors of graphs: Perron-Frobenius and Faber-Krahn type theorems. Berlin Heidelberg: Springer-Verlag

[25] Francesco Albarelli, Caratterizzazione di contact maps proteiche tramite analisi spettrale del laplaciano e misure di centralità