Patterns in Bioinformatics

A Thesis Submitted for obtaining
the Scientific Title of PhD in Computer Science
from
Politehnica University Timișoara
in the Field of Computers and Information Technology
by

Eng. Laura BROASCĂ

PhD Committee Chair:
PhD Supervisor: prof.univ.em.dr.ing. Horia Ciocarlie
Scientific Reviewers:

Date of the PhD Thesis Defense:

The PhD thesis series of UPT are:

1.Automation
2.Chemistry
3.Energetics
4.Chemical Engineering
5.Civil Engineering
6.Electrical Engineering
7.Electronic Engineering and Telecommunications
8.Industrial Engineering
9.Mechanical Engineering
10.Computer Science and
Information Technology
11.Science and Material Engineering
12.Systems Engineering
13.Energy Engineering
14.Computers and Information Technology
15.Materials Engineering
16.Engineering and Management
17.Architecture
18.Civil Engineering and Installations
19.Electronics, Telecommunications
and Information Technologies

Politehnica University Timișoara, Romania, initiated the above series to disseminate the expertise, knowledge and results of the research carried out within the doctoral school of the university. According to the Decision of the Executive Office of the University Senate No. 14/14.07.2006, the series includes the doctoral theses defended in the university since October 1, 2006.

Romania, 300223 Timișoara, Bd. Vasile Pârvan no. 2B
Tel./fax +40-(0)256 404677
e-mail: editura@upt.ro

Foreword

This thesis has been elaborated during my activity in the Department of Computers and Information Technology of the Politehnica University Timişoara, Romania.

I address my special thanks to my PhD supervisor, Prof.Univ.em.dr.eng. Horia Ciocârlie for accepting me as a PhD candidate and for guiding and supporting me throughout my lengthy PhD research studies. At the same time, a special thank you is addressed to Sl.Dr.Eng. Versavia Maria Ancușa who has always encouraged me to keep moving forward even when things were difficult, and taught me how to always find a way to overcome tough situations. This collaboration slowly turned into friendship and that is more than I could have wished for.

I also wish to thank the group of medical specialists from "Victor Babeș" Infectious Diseases and Pneumoftiziology Clinical Hospital Timișoara, which have become part of our team while striving to make sense of the immense quantity of medical data that needed to be processed and understood: Dr. Ana Adriana Trușculescu, Dr. Diana Manolescu, Prof. Dr. Cristian Iulian Oancea.

Finally, I am extremely grateful of my family and friends which have been supporting me and encouraging me all along the way and without which I could not have gotten so far.

Timişoara, Month 2023 eng. Laura Broasca

BROASCA, Laura

Patterns in Bioinformatics

PhD theses of UPT, Series X, No. YY, Editura Politehnica, 2023, 127 pages, 66 figures, 12 tables.

ISSN:
ISBN:

Keywords
……………………………………………………………………………
……………………………………………………………………………

Abstract
……………………………………………………………………………
……………………………………………………………………………
……………………………………………………………………………
……………………………………………………………………………
……………………………………………………………………………
……………………………………………………………………………
……………………………………………………………………………
……………………………………………………………………………
……………………………………………………………………………
…………………………………………………………… ………………

Article I. Table of Contents

Article II. Notations, abbreviations, acronyms

The following listing describes the meaning of the notations, abbreviations and acronyms used throughout this thesis.

Abbreviation	Meaning
CAD	Computer-Aided Diagnosis
CN	Complex Networks
CPFE	Combined Pulmonary Fibrosis and Emphysema
CT	Computer Tomography
DILD	Diffuse Interstitial Lung Diseases
DLco	Diffusing capacity of the lungs for carbon monoxide
E	Emphysema
FA2	Force Atlas 2
FR	Fruchterman-Reingold
FVC	Forced vital capacity
GE	General Electrics
GGO	Ground Glass Opacity
GUI	Graphicall User Interface
HRCT	High Resolution Computed Tomography
HU	Hounsfield Unit
ILD	Interstitial Lung Diseases
IPF	Idiopathic Pulmonary Fibrosis
MRI	Magnetic Resonance Imaging
NSIP	Non-Specific Interstitial Pneumonia
OP	Organizing Pneumonitis
PFT	Pulmonary Function Test
PPI	Protein-protein interaction
S	Sarcoidosis
SPL	Secondary Pulmonary Lobules
UIP	Usual Interstitial Pneumonia

List of tables

List of figures

1. Introduction

Network science has been an increasingly important domain since the 1736 bridges of the Königsberg problem to today’s multiple applications. Its characteristics allow it to be used as a tool that in combination with domain-specific data sets can offer a whole new perspective on the way we analyze data.

Network science is based on depicting data as networks of interacting elements and underlining patterns or models. This view into data is an invaluable asset in the sense that it is a combination of displaying quantitative data in a dynamic visual manner.

Complex networks have been used as analysis support in various areas such as computer science, social sciences, medicine, astronomy, civil engineering, psychology, etc. The advantage of using such an approach is that it can be adapted and combined with multiple other data analysis technologies: big data, machine learning, prediction algorithms, or imaging technologies.

The medical domain has greatly benefitted from this new type of science. Metabolic networks, genetic pathways, disease networks, pneumology, neurological networks, assisted or enhanced diagnostics, medical imaging - they have all used network science as a means of gaining more diverse information into the studied data[1]–[4].

Another important aspect they have in common is that the amount of information characterizing each one has increased tremendously in the past decade. Human knowledge is now richer than ever has been and it is all due to human curiosity together with the evolved technical means and infrastructure involved in the process of research.

Complex networks, due to their graphical nature and mathematical support can be a game changer in all the problems related to patterns in Medical Science. Patterns should be visualized and analyzed, exactly the forte of complex networks. These are two complementary approaches (visualization and analysis) that stand at the base of this domain and are the main focus for development.

While visualization tools are meant to bring knowledge in a creative manner, they are however faced with multiple challenges when dealing with large datasets such as the ones in the biology domain[5], [6].

Complex networks analysts are always striving to find new methods of representing data in manners that could underline certain intrinsic characteristics of those networks, unobservable by the bare human eye[7]. Given that complex networks are usually based on large data sets, it is quite intuitive that, beyond a certain order of magnitude or dimension, the human brain cannot fully coagulate and extract useful information from the studied data [8].

This is where data visualization techniques come in handy. One of the most relevant ways of depicting data is through representation layouts which arrange the data according to different criteria [4], [9]. A large number of researchers choose complex networks as a common means of visually arranging data and indeed this approach seems to manage to bring all of the data together while emphasizing connections between nodes, and cluster formation [10], [11].

A fair number of network visualization tools have been developed in response to this emerging need to deepen one's understanding of such networks. These offer different approaches to representing data starting with basic statistics (node degree, betweenness centrality, closeness centrality, modularity) and continuing with force-directed visualization algorithms (Force Atlas, Fruchterman-Reingold, OpenOrd, etc) [12].

However, one issue with visualizing data as networks has to do with the fact that there is no obvious ranking or differentiation between vertices other than their visual dimension which can be proportional to the relevance of that node within the network according to a chosen criterion. Displaying networks in a 2D manner can only give us so many dimensions of analyzing them. 3D networks on the other hand have the advantage of spatially scattering vertices and clusters and thus get a better understanding of the distance between any two nodes of the graph. Nevertheless, one major drawback of currently available tools implementing this approach is that, in the context of large data networks, the user loses grasp on the whole network and clusters seem to be floating around (although most 3D visualization tools offer the possibility of manually manipulating them)[13]. This highlights the opportunity to improve spatial distribution across a multi-dimension canvas and it is one of the gaps this thesis aims to fill.

In terms of data pattern analysis in bioinformatics (the second direction covered by this thesis) there are currently multiple approaches available. Starting with machine learning algorithms, or even off-the-shelf software applications (e.g. CALIPER), these tools are designed to consolidate the knowledge of medical specialists and support medical diagnosis with a more objective „second opinion". Diffuse interstitial lung diseases (DILD) are among the pathologies which could make the most use of such software solutions. Medical specialists heavily rely on HRCTs to be able to correctly diagnose such illnesses, yet this procedure still depends on the doctors' „clinical sense", which is, however experienced, inherently a subjective process. This disadvantage underlines a need to integrate more objective means of assessing a patient's health state into daily practice. To be able to improve the decision-making process, approaches such as Computer Aided Diagnosis (CAD) have emerged as valuable candidates aiming to enhance analytical data and reduce diagnostic errors[14].

Despite their increasing popularity and the specialists' general interest in including these types of tools in their daily diagnostic process, CAD tools are still in their infancy and are not yet reliable enough to be fully integrated with medical practice[15]. Some of them might reqire quite a large suite of additional medical tests (which doctors might not have for every patient – e.g. Caliper) while other approaches (AI, machine learning) still fail to capture the dynamics of a pathology evolution.

These pitfalls leave room for a new approach, the one proposed here, which is focused on enhancing DILD diagnosis through a complex qualitative and quantitative measurement, as well as helping to close the gap regarding illness progression and early diagnosis of such ailments.

1.2. Network visualization tools facilities and pitfalls

While there are many tools being used at the moment by researchers, not all of them are perfectly suited for all types of domains and data types. This is why multiple factors need to be taken into consideration when choosing a certain tool for to display data as complex networks.

On the one hand, the data set dimension is a multifaceted subject. Smaller data sets (up to several hundreds of nodes and edges) are probably the most versatile type of data due to the fact that a small number of elements poses no problem in representing from a canvas size point of view. As a consequence, when dealing with smaller data sets, one can easily experiment with visualization layouts and tools and not worry about whether the data will fit the screen, or if labels will clutter the image so much so that they will have to omit to display them altogether. The dimension factor is in fact an important aspect when dealing with large sets of data because this also restricts the type of tools and layout algorithm one chooses. This type of data pushes applications to the limit from multiple points of view: how well they can structure data and group similar elements together, how well they take advantage of all the canvas space to relay a clear image of the network.

When dealing with large data sets (hundreds, thousands, or even millions of nodes and edges) the approach to representing such data volumes is not as straightforward as in the previous case. This poses quite a few challenges due to the fact that the drawing canvas becomes more and more crowded. As a consequence, the pressure falls on the shoulders of visualization algorithms, pushing them toward their limits in terms of performance and efficiency. This type of entry data calls for customized applications, maybe even dedicated to a certain data type. Specialization, while useful in some cases, loses its genericity in favor of unidirectional performance. But this is only natural from a point forward.

In terms of data set specifics or domain characteristics, there are multiple fields that need and can profit from a visual representation of their structural components and dynamics. However, they do not all share the same characteristics. The number of nodes and connections, graph density and the number of clusters generally tends to differ from one domain to another.

In the biology and omics area, researchers tend to deal with very dense networks, large numbers of nodes and edges, or incomplete data: DNA networks, genetic pathways, brain networks, or protein-protein interactions (PPIs)[16], [17].

Social network analysis is also an area strongly relying on complex networks for a visual representation since it is the closest conceptual structure to the real social network. Naturally, network vertices are a correspondent of real entities, while edges represent the relationship between them. These types of networks as well have a large number of elements and density[18]. Author citation networks are yet another type of application of complex networks, as well as astronomical networks, or even road networks.

The purpose that such visualization tools are needed for is another important factor to consider, and this greatly depends on the user's needs. One of the most common applications of network layouts is that it constitutes a vizual support when dealing with abstract data. It could also represent another point of view along with charts and diagrams to complete the picture. However, sometimes this sort of visualization technique constitutes the main research tool in itself.

Fields such as genetics - which still have incomplete data – seek to fill in the missing pieces by looking at "the big picture", hence the integration of complex networks into the analysis process. Discovering the different tendencies or intrinsic characteristics of such networks together with similarities expressed by individual nodes could help uncover either missing elements or give new meaning to the existing network entities. However, the ability to accurately represent the graphical equivalent of such complex ecosystems becomes a challenge for current visualization tools, given the overwhelming amount of available data. This drawback forces network layout tools to make certain concessions on a number of aspects: whether it is in terms of graphical appeal, performance-wise, or network complexity.

Although there are numerous applications of network visualization tools and layouts [19], [20], the current thesis focuses on their role within the biology domain, one specific area which is both problematic and intriguing. This approach strives to reduce the aforementioned limitations by offering a 3D network layout for medium to large networks which offers more depth and a more meaningful visual representation of such ensembles.

1.3. Pattern analysis in the medical domain - challenges

The medical diagnosis process is more and more based on complex procedures generating high-resolution images of different parts of the body. MRI, CT/PET scans, X-rays, or 2D/3D ultrasound have not only evolved a lot in the past decades but also tend to incorporate different detection algorithms which help as decision support in the diagnosis of different diseases[21], [22].

On the one hand, visual support is always a more solid and reliable source of information that reflects reality as is, rather than by trying to clinically analyze the patient's health state, which is a more subjective and error-prone process[23].

In addition, the advantage of such high-resolution images is that they can be used as input for other pattern detection algorithms. These AI or machine learning algorithms have started becoming more and more important in the medical field[24]. Although they are not evolved enough yet and cannot substitute a doctor's diagnosis, for understandable ethical reasons which constitute an ongoing debate, they are definitely a second opinion worth taking into consideration. All this only provided that the detection algorithms have had enough training data so as to reduce the misdiagnosis chance as much as possible[25].

The applications of machine learning and pattern identification algorithms in the medical domain are numerous: computer-aided diagnosis, image interpretation, image fusion, image registration, image segmentation, image retrieval, and analysis. Abnormalities detection within high-resolution imaging such as lung disease detection, bone disease or abnormalities, heart disease, muscle structure analysis, tumor tissue evaluation, and death risk probability - these are all development directions encompassed by machine learning algorithms[26]–[28].

In terms of DILDs, which are a category of over 200 lung diseases, the diagnosis process is a very complex one, entailing multiple tests and procedures, cumulated with the medical knowledge and experience of the medical specialists. Besides being a subjective procedure in some aspects, this approach is also in need of technical tools which could enhance and speed up the diagnosis treatment process, especially since time is sometimes a critical aspect in managing lung pathologies. While there are several CAD tools available at the moment which have slowly started making their way into daily practice[29]–[31], these software applications are still not mature and comprehensive enough. Some tools like Caliper require a set of extra lung parameters and tests (e.g. pulmonary function tests, spirometry) to be able to offer a pertinent output, yet other programmatical approaches lean towards a more in-depth approach at evaluating lung sections. Nevertheless, one major drawback of such solutions is that they merely provide a static analysis of HRCTs, and fail to record the time variation of a pathology evolution. In addition, none of them innovate in terms of DILD early diagnosis and classification, which are crucial medical management information, needed to prolong the life quality and duration of the afflicted patients.

1.4. Research objectives

Given the problematics exposed here, this thesis aims to develop complementary solutions to the visualization and analysis of patterns from the medical and biological domain.

First the development of a new layout algorithm that fulfills the demanding requirements imposed by the medical data specifics. This new hybrid 3D approach strives to highlight and categorize complex networks data to reveal intrinsic characteristics which would otherwise not be available.

Secondly, on the account of data patterns, this thesis proposes a new complex networks approach to HRCT processing, which would allow medical specialists to perform an in-depth analysis of medical data with much higher accuracy than the human eye. This technical solution also proposes a novel way of understanding the dynamics of DILD pathologies (deterioration rate or affected lung volume) from an angle that has not yet been exploited at its true potential: complex networks analytics.

2. Theoretical Background

2.1. Complex networks

The natural way of visualizing biological data is that of a network [32]. Because similarly to a network, DNA, for example, consists of genes that interact with each other and interconnect. It is easier to view clusters, groups of elements with the same or similar function, or groups of elements that are so tightly coupled that they are seen as part of a process (metabolic processes for example).

One other advantage of complex networks is that there are different types of network models to choose from in order to best fit the training data. Choosing the right type of network model not only helps understand patterns that rule the network but can also even help predict missing links that may not have been experimentally discovered until now [33].

2.2. Complex networks metrics

A complex network is essentially a graph, and as such, it usually consists of vertices and edges. The vertices (or nodes) are the elements that compose the system and the edges (or links) are the connections between the nodes.

2.2.1 Degree distribution

Node degree is probably the simplest and the most basic attribute of a node. This characteristic represents the number of links a certain node has to other vertices in the graph. The short notation for node degree is $deg(n_i)$. Depending on whether the graph is directed or not, a node n_i might have an in-degree ($deg-(n_i)$) and an out-degree ($deg+(n_i)$) (for incoming respectively outgoing links or directed graphs) or simply, a degree (for undirected graphs).

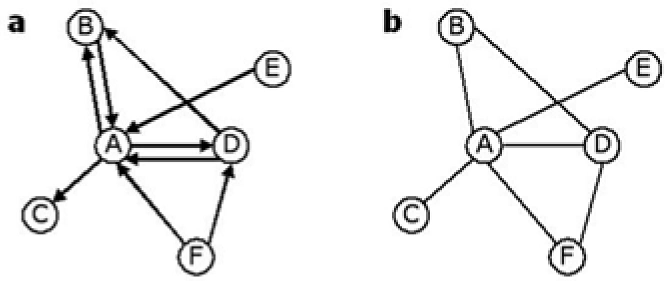


Figure 2.1 a) Directed graph b) Undirected graph [34]

Complex networks such as genetic networks can be represented as both, depending on their type and specificity. Protein-protein interaction networks (PPI), gene regulatory networks (GRNs), Signal transduction networks, or Metabolic networks are just a few examples that can be represented as directed graphs since connections in the graph are actually interactions that happen at certain points in time and the order and precedence in which these take place can be translated into directed edges [34].

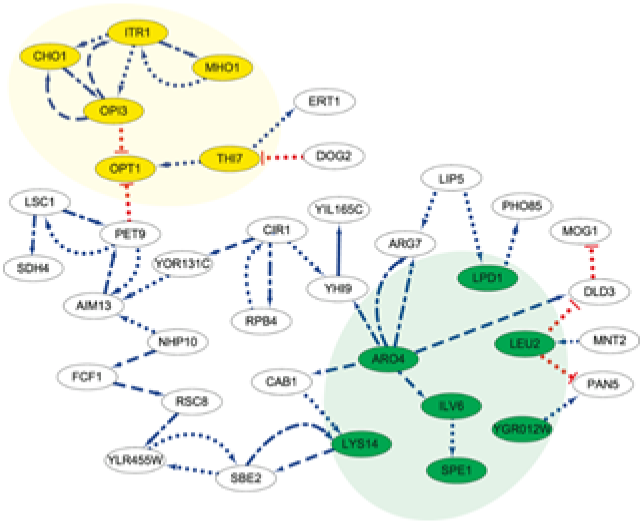


Figure 2.2 Sample metabolic network [35]

The network diameter represents the longest path that links two nodes of the graph.

The degree distribution of a graph $(P(k))$ can be described as the probability that a randomly selected node has precisely k edges. For a graph with n nodes, n_k of them having the degree k, then [36], [37]:

$$P(k) = n_k/n \tag{2.1}$$

2.2.2 Clustering coefficient

In graph theory, the clustering coefficient measures the degree to which graph vertices tend to cluster together [38]. There are two types of clustering coefficients: the global clustering coefficient and the local clustering coefficient.

The global clustering coefficient (C) measures the overall indication of clustering in a graph and is calculated as the ratio of closed triplets of nodes (or closed triangles) to the total number of triplets (open or closed) in the graph.

The local clustering coefficient of a node gives an idea about its embeddedness, or in other words, how close its neighbors are to being a complete graph. For directed networks, the coefficient can be calculated as: directed networks the formula becomes:

$$C_i = \frac{|e_{jk}: v_j, v_k \in N_i, e_{jk} \in E|}{k_i(k_i - 1)} \tag{2.2}$$

Where e_{jk} is the edge connecting vertice v_j to v_k, N_i being the neighborhood for a node v_i and can be defined as:

$$N_i = \{v_j: e_{ij} \in E \ V \ e_{ij} \in E\} \tag{2.3}$$

while k_i represents the number of neighbour nodes (($|N_i|$) of a vertex. In the case of undirected networks, the formula becomes:

$$C_i = \frac{2|e_{jk}: v_j, v_k \in N_i, e_{jk} \in E|}{k_i(k_i - 1)} \tag{2.4}$$

2.2.3 Modularity

Modularity (Q) is a measurement that indicates network structure. The purpose of this indicator is to quantify the division strength of a network into modules (or communities) [39]–[41]. It can take values in the range $[-1, 1]$. A high modularity coefficient suggests a network with dense connections between members of the same communities, yet few links between nodes pertaining to different clusters. A lot of examples having this characteristic can be found in biological networks or social networks.

As a practical example, assuming a vertex v belongs to community i (ς_i) and it does not take part in community j (ς_j) , then, when defining a membership variable s which characterizes v concerning the two communities, $s_v = 1$ in the case of ς_i and $s_v = -1$ for ς_j.

There are a number of different ways for computing network modularity [32], and one of them is the following:

$$Q = \sum_{\varsigma_i \in \varsigma_{set}} \left[\frac{E_{\varsigma i_{in}}}{E} - \left(\frac{2 \cdot E_{\varsigma i_{in}} + E_{\varsigma i_{out}}}{2 \cdot E} \right)^2 \right] \tag{2.5}$$

where ς_{set} is the set of all communities, ς_i is a cluster belonging to the set, $E_{\varsigma i_{in}}$ represents the number of edges between nodes in the community, $E_{\varsigma i_{out}}$ is the number of outgoing connections from the community and is the total number of edges [42].

2.2.4 Density

The density D is the measurement that reflects the ratio of the number of existing edges E in a network with N nodes to the total number of possible edges [43]. It can be computed as:

$$D = \frac{2(E-N-1)}{N(N-3)+2} = \frac{T-2N+2}{N(N-3)+2} \tag{2.6}$$

where ties T are unidirectional.

2.3. Centrality Measures

2.3.1 Degree Centrality

Degree centrality is based on the degree of each node and assigns them a weight based on the degree [44], [45]. Degree centrality is also a specific case of another measure called *k-path* centrality which counts all paths of length *k* or less which start from a node. In the case of $k = 1$, this measure is identical to *degree centrality*.

2.3.2 Closeness centrality

Closeness centrality (or the closeness of a node) is calculated as the sum of all distances (shortest paths) from a node to all other nodes. A larger value of this measure indicates less centrality and as a consequence, this is an inverse centrality measure.

$$C(x) = \frac{1}{\sum_y d(y,x)} \tag{2.7}$$

where $d(y,x)$ is the distance from vertice *x* to *y*.

2.3.3 Betweenness centrality

Betweenness centrality is a centrality measure based on the number of shortest paths that a vertice is a part of. The greater the number, the bigger the node centrality is [46].

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.8}$$

Or as defined by Freeman:

$$C_k^{BET} = \sum_i \sum_j \frac{g_{ijk}}{g_{ij}} \tag{2.9}$$

where g_{ij} represents the number of geodesic paths from *i* to *j* and g_{ijk} is the number of geodesic paths that go from *i* to *j* and pass through a node *k*. "The measure is, in effect, *k*'s share of all shortest-path traffic from *i* to *j*, summed across all choices of i and j" [45].

2.4. Network models

2.4.1 The Albert Barabasi mathematical model

The Albert Barabasi mathematical model is based on a degree distribution that resembles a power law degree distribution. Growth and preferential attachment are two of the distinct characteristics of this model. Meaning that as the network extends, a new link is being added and new links will always be prone to attaching to nodes that already have a high degree, because the probability of this being true is higher [4], [47]. As an example of studied networks which support Barabasi's theories, there's the Human Disease network [2], metabolic networks [32], [48], computer networks [49], cosmic networks [50], social networks [51], etc.

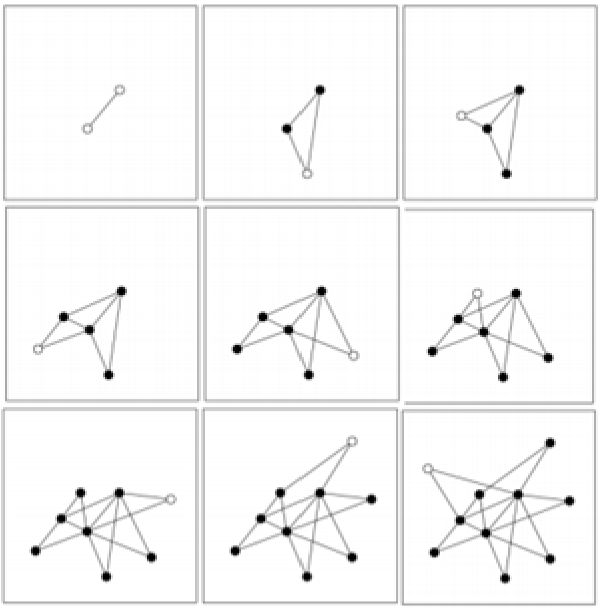


Figure 2.3 Evolution of the Albert Barabasi model [33]

The algorithm can be described in the following: it starts with an original graph (m_0) of connected vertices. As new nodes are being gradually added to the network, they are linked to $m \leq m_0$ already added nodes with a probability that depends on the degree of the existing nodes.

$$p_i = \frac{k_i}{\sum_j k_j} \tag{2.10}$$

where k_i represents the degree of each node *i* and the $\sum_j k_j$ – the sum of all existing nodes j.

2.4.2 The Erdős–Rényi model

The Erdős–Rényi model is designed for generating random graphs. The utility of this model comes into play when trying to demonstrate that certain graphs satisfy a certain property [52].

Given graph $G\{E, V\}, n = |V|, m = |E|$. The is a probability *p* that a pair of nodes in the graph are connected.

$$\langle m \rangle = p\frac{n(n-1)}{2} \tag{2.11}$$

$$\langle k \rangle = \frac{1}{n}\sum_i k_i = \frac{2\langle m \rangle}{n} = p(n-1) \approx pn \tag{2.12}$$

$$p = \frac{\langle m \rangle}{n(n-1)/2} \tag{2.13}$$

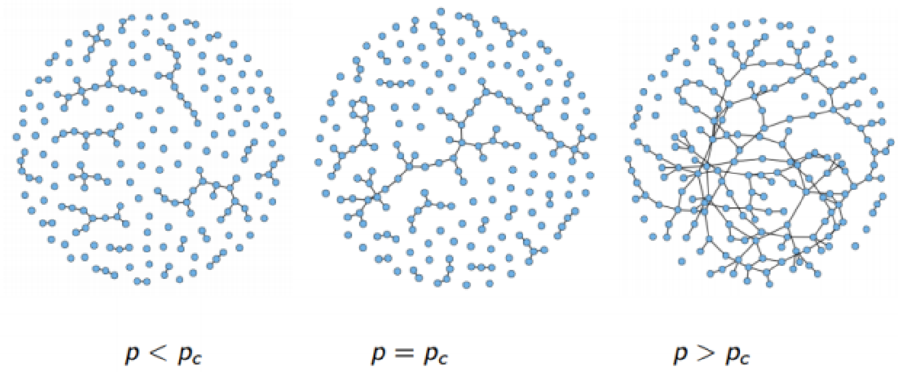


Figure 2.4 Evolution of Erdos-Renyi graph [53]

When $p = 0$ then the graph is empty, for $p = 1$ graph is complete. There is a p_c where the graph starts to change in structure ($p < p_c$ to $p > p_c$). For a large connected component forms [54], [55].

2.4.3 The Watts-Strogatz network model

The Watts-Strogatz network model was developed by Duncan J. Watts and Steven Strogatz. It is a random graph generation model which creates graphs with small-world properties.

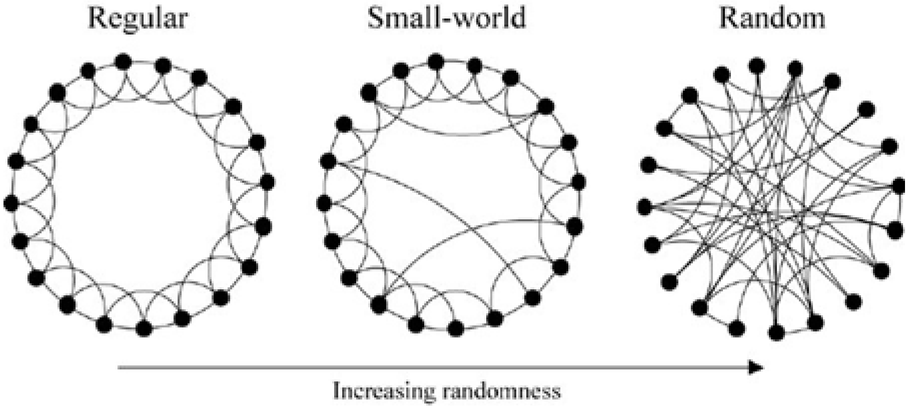


Figure 2.5 The construction of the Watts-Strogatz model [56]

The method by which this model is composed starts with a ring of N nodes. Each node gets linked to its k nearest neighbors. Then, one by one each node's edge to its nearest neighbor is redirected to another neighbor with a probability p. Once this step is done for all nodes, the same procedure applies to each node's connection to its second nearest neighbor, and afterward to the third, and so on until their furthest neighbors are considered [56].

The probability (*p*) influences how random the resulted graph will be, however it keeps the number of vertices and edges constant [57].

2.4.4 The random geometric graph

The Random geometric graph describes a spatial network, with nodes placed randomly and links between nodes added only if the distance between the two vertices is within a certain established range. This type of model suits very well social networks as the resulting communities seem to resemble very much real human social networks [58].

2.5. Clustering Algorithms

2.5.1 Louvain community detection

As its name suggests, the Louvain method is a community detection algorithm conceived by Etienne Lefebvre in 2007 and later on improved by Vincent Blondel, Jean-Loup Guillaume, and Renaud Lambiotte. It bears the name of the Louvain University, where all its creators worked during the development of this method [59].

This algorithm was first designed for undirected and unweighted graphs, however, it can and has been adapted to directed or weighted graphs. There are at the moment numerous network visualization tools that include it in their arsenal, such as Gephi, NetworkX, iGraph framework for Python and R, Neo4J, etc.

First of all, modularity (based on the concept *Q* introduced by Newman and Girvan [60]) is defined agreed as:

$$Q = \sum_r (e_{rr} - a_r^2) \tag{2.14}$$

where e_{rr} is the ratio of links that link two vertices within cluster r, a_r is the fragment of edges that have either one or both nodes inside the cluster r, and sum extends to all clusters r in a given graph. The accuracy of partitioning a network into communities is proportional to the value of Q.

Starting from the weak and strong community definitions established by Radicchi el al. [61], the authors propose a new one: if V_1, V_2,..., V_n are *m* communities of *G*, V_k, *k=1,2,...,m* should satisfy that

$$\bigcup_{k=1}^{k-m} V_k = G \tag{2.15}$$

and

$$\forall j \in V_k, \sum_{i \in V_k} A_{i,j} \geq max\left\{\sum_{i \in V_t} A_{i,j}\, , t = 1,2,\dots,m\right\} \tag{2.16}$$

In other words, a community should reflect the condition that the degree of each node within its community should be higher than its degree to the rest of the communities.

$F_{k,i}$ defines the attraction force of cluster *k* to vertex *i* and $F_{k,i}$ may be defined as:

$$F_{k,i} = \sum_{j\in V_k} A_{i,j} \tag{2.17}$$

The process consists of a few steps which include grouping each node and half of its neighbors into temporary communities, calculating the attraction force F_k for every *k* and *i*, moving each node to the communities with the largest attraction forces, respectively, removing duplicate communities if there are any. These steps are to be repeated either for a predefined number of iterations or until the partitions are fixed.

The algorithm's overall complexity is O(n2) which claims to be lower than other similar existing algorithms.

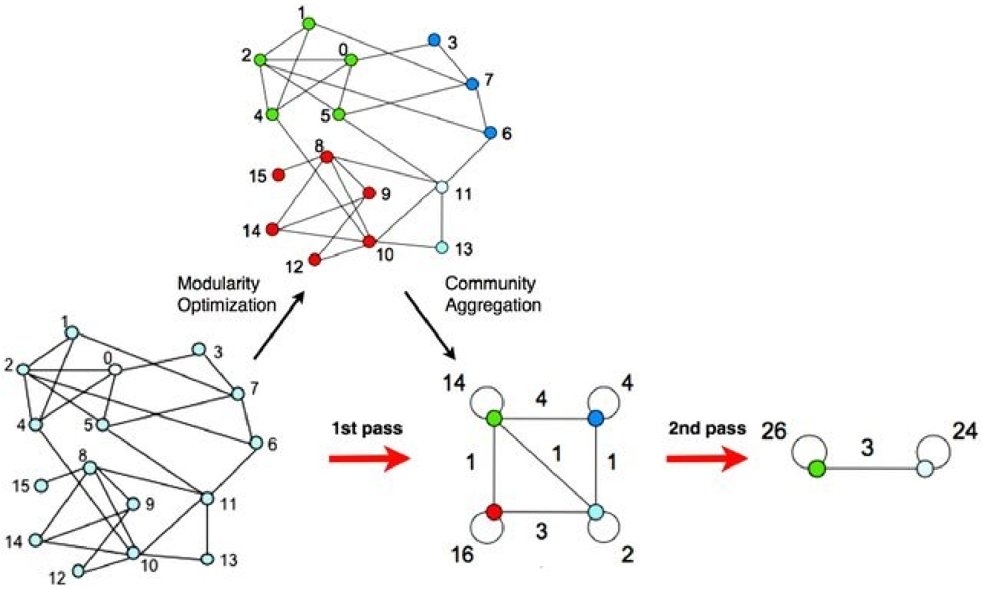


Figure 2.6 Louvain algorithm phases: modularity optimization and community aggregation [59]

The algorithm's ability to correctly identify connected components has been challenged in the past years by other field researchers claiming that the Louvain algorithm has a flaw in the fact that it can yield internally disconnected communities - meaning that some nodes pertaining to the same community are only interconnected by paths going through external nodes[62].

This is only a manifestation of an underlying issue, claim the Leiden algorithm creators, and the the subject of the qualitative partitions has also been discussed and analyzed by others in the context of large networks [63], [64].

2.5.2 Leiden algorithm

The Leiden algorithm is a community detection algorithm that comes to fill the gaps left by the Louvain algorithm and claims to be both faster and more accurate in detecting clusters. Moreover, it scales well on very large networks (millions of nodes - but this also depends on the available memory) [62]. It can be used with the help of the iGraph package (Python version).

Unlike the Louvain algorithm, Leiden starts with a single partition consisting of all network nodes. They are then redistributed among communities and then further refined in order to be aggregated and then repeat the whole process with the newly formed aggregate network, for a number of iterations until the result reaches a fixed phase which means no further improvements can be made. (Fig. 2.7)

In terms of efficiency, Leiden aims to shorten the processing time through the use of a local moving phase which eliminates unnecessary processing of stable nodes and focuses only on the ones which don't seem to have found their final state yet.

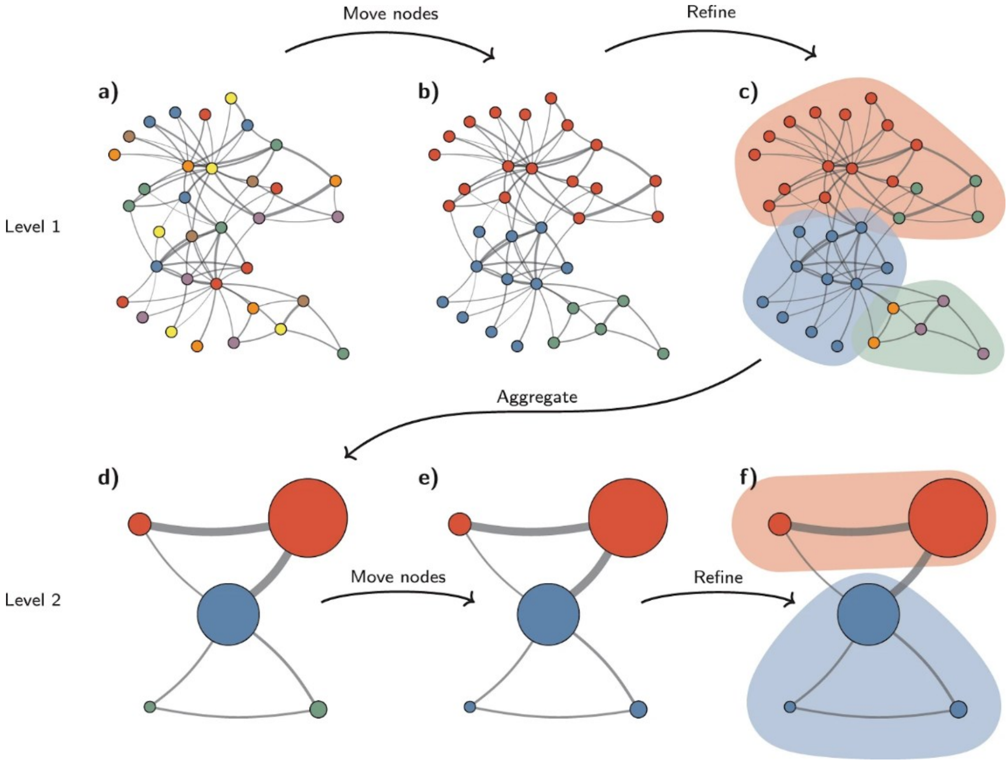


Figure 2.7 Leiden community detection algorithm [62]

2.5.3 K-means clustering

K-means clustering is one of the most popular clustering algorithms, also used in conjunction with machine learning techniques. The algorithm consists of several steps starting with selecting some centroids (k) for future clusters. These elements are also called reference buses. The rest of the nodes are then assigned to their corresponding clusters according to their closeness (euclidean distance) to one of the k centroids. When all vertices are assigned to a community, centroid positions are recalculated adjusting them to the newly created clusters with the following formula:

$$\mu_j = 1/n_j \sum_{i=1}^{n_j} x_{ij} \tag{2.18}$$

The error measure is then recalculated with:

$$\phi = min \sum_{j=1}^{k} \sum_{i=1}^{n_j} |x_{ij} - \mu_j|^2 \tag{2.19}$$

In equation (2.19), *k* stands for the total community number, while the number of buses of the *j*-th community is noted with $n_{j.}$ X_{ij} is the bus in the *j*-th community, while the centroid of the *j*-th community is symbolized by *μj*. The euclidean distance between x_{ij} and *μj* is noted with $|x_{ij}-\mu_j|$.

The previous steps are repeated until all reference buses are stable and remain in a fixed position, meaning that the algorithm has finished. This algorithm is highly dependent on the initial selection of the k clusters [65].

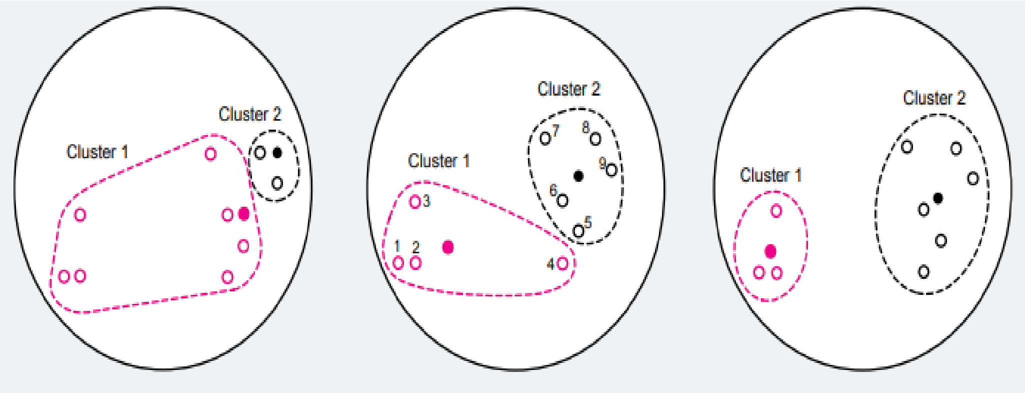


Figure 2.8 K-means Clustering example with diagrams [65]

Figure 2.3 depicts how two iterations of K-means clustering would develop given that the data consists of nine 2D data points. These points are then grouped into two distinct communities. Vertices in community 1 have a bright pink color, while the ones in the second community are all black. Data nodes are represented by empty circles, while reference points are all filled circles. Communities are highlighted through dotted lines. This example shows that, even if the initial reference points are wrongly chosen, the algorithm swiftly converges toward the correct clustering[65].

2.6. State-of-the-Art

2.6.1 Graph layout algorithms

Force Atlas & Force Atlas 2

FA and FA2 are continuous layout algorithms based on making highly connected nodes attract and unrelated nodes (or indirectly related) or clusters repulse. The live visual spatialization of the network nodes is one of the bonuses of this algorithm. Asides from getting a visually pleasing graph, the user can also incrementally watch the nodes pulling farther apart, or getting closer together. Due to the specificity of this algorithm, it runs indefinitely until the user obtains the desired result and stops it. The nodes' positions are not related to any specific variable, and they can only be interpreted in comparison to the others. Being part of the force-directed algorithms, it is naturally based on Hooke's law of physics.

The formula for this force-directed algorithm is based on the repulsion formula of the electrically charged particles $F_r = k/d^2$ and the attraction formula of springs $F_a = -k \cdot d^2$ [66].

As their creator Mathieu Jacomy notes, Force Atlas - launched in 2007 - can handle a graph of up to 10,000 vertices, and Force Atlas 2 (as an improved version of the former, launched in 2011) can work with 1,000,000 nodes.

Fruchterman-Reingold

The Fruchterman-Reingold is also a force-directed algorithm but it differs from Force Atlas/ForceAtlas2. The attraction forces formula is $F_a = d^2/k$ and the repulsion formula is $F_r = -k^2/d$, with adjusting the scaling of the network) [67].

This approach aims to distribute vertices evenly in the available space, make edges lengths uniform, minimize edge crossings, and fit to the frame. Also, as it was projected by its owner, it was thought as a system of rings and springs, where the rings represent vertices and respectively the springs represent the edges and forces between them. This idea was inspired by the work of Eades, which also launched the idea of calculating repulsing forces between all nodes but attracting forces only between connected ones [67].

Although as opposed to the Force Atlas layout, it produces a more evenly distribution of nodes across the graph, and behaves well for a maximum of 1000 nodes.

OpenOrd (VxOrd)

OpenOrd (2010) is also a force-directed 2D algorithm that aims to single out clusters from one another. It is based on VxOrd, an implementation of Fruchterman Reingold, and is designed as a parallel algorithm that increases its efficiency and stops automatically. It can handle large datasets (100 to 1,000,000 nodes) and it aims to improve three aspects of Fruchterman Reingold: it is faster, it is more visually appealing and it favorises uncovering the global structure of a network to the detriment of a precise local structure [55]. The formula it attempts to solve is:

$$min_{x_1,\dots,x_n} \sum_i \left(\sum_j \left(w_{ij} d(x_i, x_j)\right)^2\right) + D_{x_i}\right) \quad (2.20)$$

Where D_{x_i} is the density of the points $D_{x_1 \dots x_n}$ near x_i, $\sum_j \left(w_{ij} d(x_i, x_j)\right)^2\right)$ is the attractive term, drawing points closer together, while D_{x_i} is the repulsing term pulling them apart.

The interesting part of the algorithm consists of iterations in which the position of each vertex is calculated in two possible ways. In addition, each stage of the algorithm is controlled via a simulated annealing type schedule, consisting of five phases: liquid, expansion, cool-down, crunch and simmer .

Yifan-Hu

Yifan-Hu is also a force-directed multilevel algorithm, well suited for large graphs - up to 100,000 nodes. It makes use of the Barnes-Hut optimization [68] and treats clusters of distant nodes as single super nodes, thus reducing the complexity of standard force-directed algorithms [69].

Kamada and Kawai

Kamada Kawai algorithm is based on the same metaphorical representation of a graph as Eades and Fruchterman-Reingold [67]. However, besides the rings and springs along with calculating the attracting forces only among connected nodes, he solves some partial differential equations (based on Hooke’s law) and introduces the notion of ideal distance between nodes, which is proportional to the length of the shortest path between two nodes.

$$\sum_{\leq i<j\leq |V|} k_{ij}\left(|p_i - p_j| - l_{ij}\right)^2 \quad (2.21)$$

where p_i is the position of the ring corresponding to vertex v_i, k_{ij} is the spring constant for the spring between p_i and p_j , and l_{ij} is the optimum distance between vertices v_i and v_j [70].

According to Kamada Kawai, drawing a graph is a matter of reducing the amount of energy of a system (exerted by compression and repulsion forces) and the nearest that two nodes would ever be is equal to the ideal distance calculated for two rings.

An important aspect to be noted here is that every ring’s location is calculated iteratively and not in parallel to calculating others’ locations, and as a result, in one step, only the impact of one node’s forces over the system is being modified. This step repeats for every ring until the result reaches a pre-established threshold.

2.6.2 Visualization tools

Gephi

Gephi is a standalone data visualization tool launched in 2010. It has greatly developed over time, offering more and more features for data layout and graph metrics/statistics. The available metrics are: Betweenness Centrality, Closeness, Diameter, Clustering Coefficient, PageRank, Community detection (Modularity), Random generators, Shortest path.

Regarding graph layout algorithms, Gephi implements two types of layout categories: force-based algorithms and optimize for graph readability. Among these: Contraction, Expansion, Force Atlas, Force Atlas 2, Fruchterman-Reingold, Noverlap, OpenOrd, Random Layout, Yifan Hu. Algorithm parameters can easily be customized through the user interface, to achieve the best layout [71].

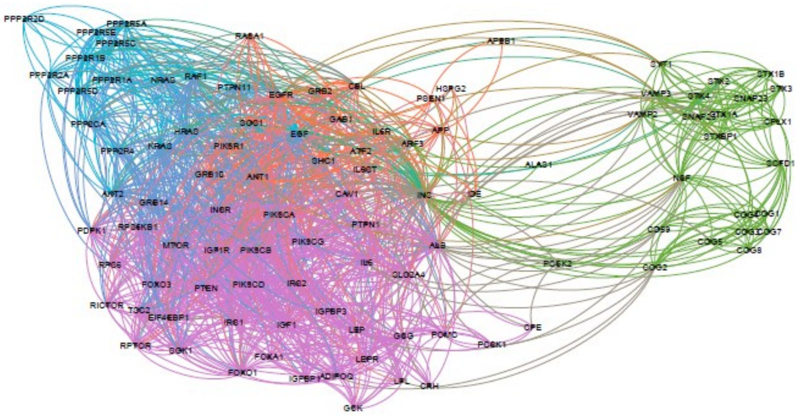


Figure 2.9 Gephi network sample of a genetic network

The main challenge with Gephi layouts is that it is difficult to accurately measure the amount of time needed for rendering any type of layout. In the case of Force Atlas 2 layouts, given that Force Atlas is a continuous algorithm, it can run indefinitely until the user decides to stop it from running. In addition, depending on the available processing power and the available resources overall, the algorithm can run faster or slower. However, the dynamic graph clearly reaches a stability phase, not long into the rendering process, when most clusters are already compactly grouped and node positions stop varying dramatically. Graph movement inside the drawing canvas becomes less and less evident, and no drastic changes can occur anymore. Yet stabilization time clearly depends on the size of the graph.

Python (NetworkX)

NetworkX is a graph layout library implemented in Python. It too implements two categories of node positioning algorithms, structural and force-directed: circular layout, random layout, shell layout (concentric circles), spring layout (Fruchterman-Reingold), spectral layout(positioning nodes based on the eigenvectors of the graph Laplacian). Graph analysis metrics like centrality,and network density are also available in Python. As its creators state, NetworkX is not designed with the purpose of competing with tools like Gephi or Cytoscape in terms of graph rendering. However, when dealing with large sets of data, trading visual appeal for computation speed and efficiency may become a compromise worth making [72].

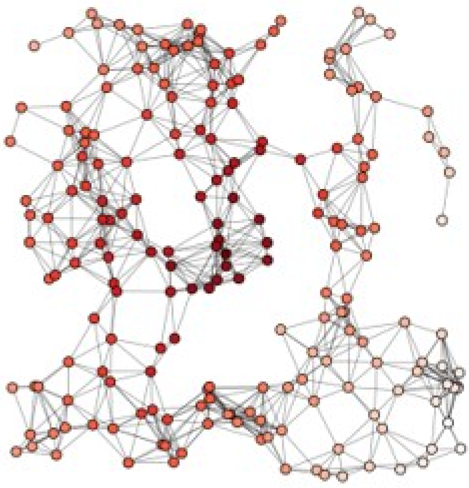


Figure 2.10 NetworkX sample network [73]

R (iGraph)

R comes with its own graph layout library iGraph. This library is also available in Python. Among the implemented 2D layout algorithms: igraph_layout_graphopt — Optimizes vertex layout via the graphopt algorithm, Kamada Kawai, multidimensional scaling, LGL (Large graph layout algorithm), Reingold-Tilford layout for tree graphs, Circular Reingold-Tilford layout for trees, Sugiyama layout algorithm for layered directed acyclic graphs. Some of these algorithms are also used for 3D layouts (Fruchterman-Reingold, Kamada Kawai).

Similar to the NetworkX in Python, iGraph is also a programmatic solution to graph rendering and lacks a user interface. Despite not being best in class at user friendliness or visual aspect, it does however deliver good results in terms of speed and efficiency, which makes it also a good alternative to the previously mentioned tools [74].

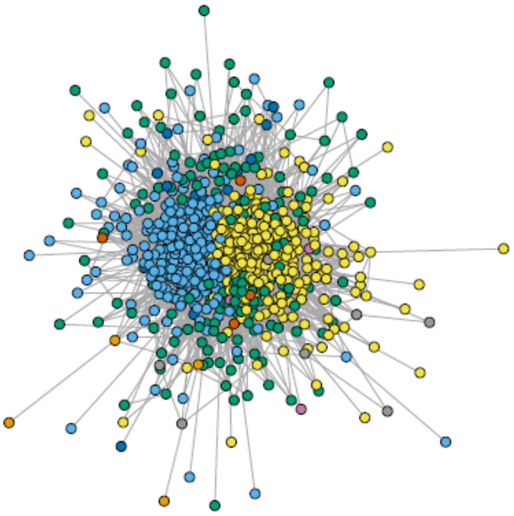


Figure 2.11 iGraph sample network of DNA mice genes

Cytoscape

Cytoscape is an autonomous 2D data visualization tool. It implements force-directed, grid or circular layout algorithms [75]. However, it is not recommended for large datasets as it becomes very slow in rendering and quite resource-consuming [76].

Cytoscape comes as an alternative to Gephi. It is similar to it in the fact that it was first created as a standalone application offering multiple network layout options. However, it now offers a Javascript library that has similar functionalities to the application. Similar to Gephi, graphs can be imported into the application in numerous ways: as a list of nodes, as a list of edges, as unformatted files, as formatted text files, or Excel Workbooks. All this is packaged into a fairly intuitive user interface. In any case, the available documentation extensively covers all functionalities, being a user-friendly application.

A few of the helpful features in terms of graph visibility and clarity include grouping nodes into clusters and reducing them to one element to decrease complexity, eliminating node labels if the number of nodes is above a certain predefined threshold (200), or if the sum of nodes and edges is bigger than 4000.

Cytoscape aims to be a versatile network visualization tool given the palette of network layouts it implements.

The grid layout is a default one and is one of the most simplistic ways of displaying data. Just like its name suggests, the elements represented using this algorithm are dispersed in a grid shape. This view is less efficient when the number of nodes and edges is high. However, if dealing with a reasonable graph density, it might come as a means of viewing data in a table-like manner, giving it more structure.

Attribute Circle Layout is simple yet effective when dealing with small networks and it places nodes on a circle, grouping them according to a user-selected criterion.

Group Attributes Layout resembles the Circle Layout with the sole difference that nodes are split into multiple circles. Nodes having the same attribute value are placed on the same circle.

Edge-weighted Spring-Embedded Layout is one of the most interesting in Cytoscape's arsenal. It is based on the principle of electrically charged particles as implemented by Kamada Kawai. This means that nodes play the roles of particles that attract or repel according to certain forces. The bigger the similarity between two nodes, the stronger the attraction force and the closer together they end up on the graph. Conversely, the more two vertices differ, the bigger the repelling force and the farther apart the nodes will be placed.

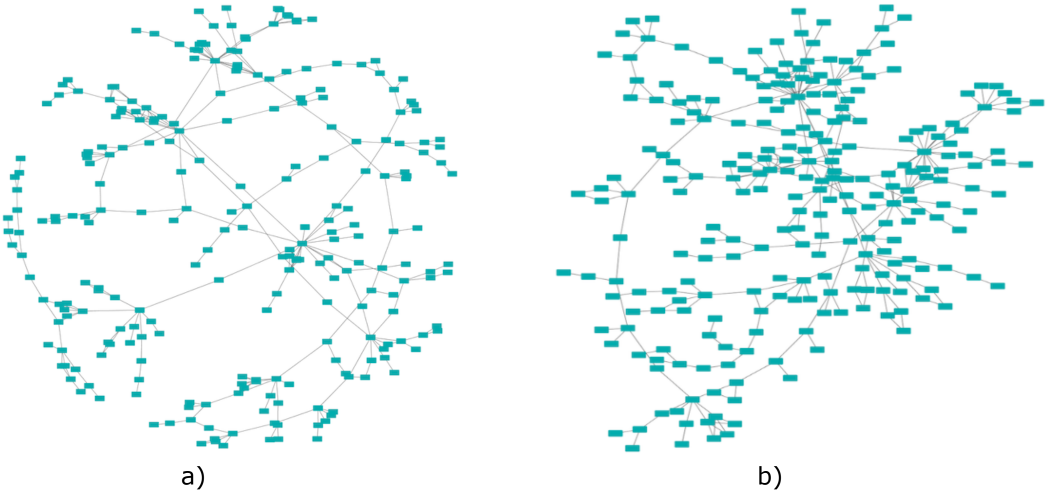


Figure 2.12 Cytoscape layouts (a) Edge Weighted (b) Prefuse force directed [77]

Prefuse Force Directed Layout is also a force-directed layout inspired by Jeff Heer. It claims to have a very good performance in terms of generation speed and it generates interesting results.

The advantage of the Compound Spring embedded Layout is the fact that it maps well to graphs that require compound nesting of elements.

Circular Layout is another version of Attribute Circle Layout which groups the nodes into circular shapes. These circles are then dispersed into a tree shape.

The Hierarchical layout does exactly what its name suggests. It structures the element according to a certain chosen hierarchical criterion. This approach resembles the grid layout, except the rule by which nodes are arranged ensures minimum edge crossing and conveys the way information flows within such a network.

The Copycat Layout is an interesting feature that allows the arrangement of network nodes by copying the structure and layout of another. This theoretically allows "importing" other types of layouts into Cytoscape and applying them to the desired network without having those layouts already implemented by the application [77].

As an extra package, yFiles addon can be downloaded and added to Cytoscape, thus adding a few more to the existing ones: Circular Layout, Hierarchic Layout, Hierarchic Layout Selected Nodes, Organic Layout, Orthogonal Layout, Radial Layout, Tree Layout, Orthogonal Edge Router, Organic Edge Router.

Nodetrix

NodeTrix is a hybrid network visualization tool designed especially for social networks. It aims to address the problem of graph readability in a grid-like manner.

NodeTrix is an application that claims to offer a user-friendly interface, with an accent on interactivity capabilities. It allows dragging of nodes around the graph and rearranging elements so that the user can better group them or highlight certain aspects [78].

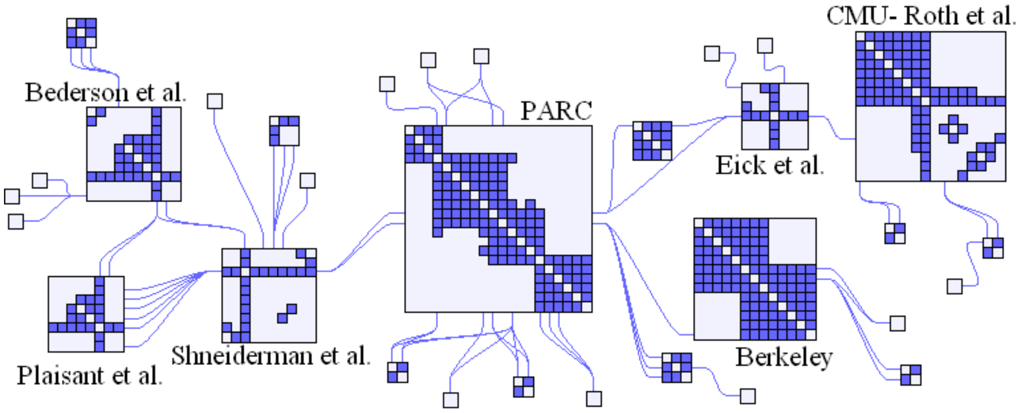


Figure 2.13 Co-Authorship network - NodeTrix layout [78]

Although it uses matrices for intra-cluster links and a classical node-edge approach for the overall graph, this looks more like a mathematical approach to network visualization. This might be a less popular feature among researchers given that for intra-cluster visualization, we do not get an idea about how close nodes are to each other or how related they are. One can only see that there is a connection between adjacent nodes.

Depending on the purposes this tool is used for, it could be helpful to some extent yet it does have limitations. In a network based on a genetic data set, this is an incomplete layout, as it does not offer the complete image. Genetic pathways are for example an incompatible set of data, as it is important to follow the flux of information or interaction paths among all network nodes.

IGraph

IGraph is a network visualization library created to work with Python and R. It implements multiple 2D and 3D visualization algorithms

Its Random layout (1.1 igraph_layout_random) is one of the simplest options, which distributes nodes randomly and uniformly in a 2D space. [74]

The Circle layout (igraph_layout_circle) is quite a simplistic or rudimentary view, given that the strategy for placing nodes around a circle is solely based on ids, which is not very relevant from a user perspective.

There are numerous other algorithms implemented by iGraph among which:

- igraph_layout_star — Generate a star-like layout
- igraph_layout_grid — Places the vertices on a regular grid on the plane.
- igraph_layout_graphopt — Optimizes vertex layout via the graphopt algorithm.
- igraph_layout_bipartite — Simple layout for bipartite graphs
- igraph_layout_drl — The DrL layout generator, igraph_layout_drl_3d — The DrL layout generator, 3d version.
- igraph_layout_fruchterman_reingold — Places the vertices on a plane according to the Fruchterman-Reingold algorithm.
- igraph_layout_kamada_kawai — Places the vertices on a plane according to the Kamada-Kawai algorithm.
- igraph_layout_gem — The GEM layout algorithm, as described in Arne Frick, Andreas Ludwig,
- igraph_layout_davidson_harel — Davidson-Harel layout algorithm
- igraph_layout_mds — Places the vertices on a plane using multidimensional scaling.
- igraph_layout_lgl — Force based layout algorithm for large graphs.
- igraph_layout_reingold_tilford — Reingold-Tilford layout for tree graphs
- igraph_layout_reingold_tilford_circular — Circular Reingold-Tilford layout for trees
- igraph_layout_sugiyama — Sugiyama layout algorithm for layered directed acyclic graphs.

It is worth mentioning here that some of them are more popular than others and can also be found in other network visualization tools.

Fruchterman Reingold and Kamada Kawai are among Gephi's list of force directed algorithms as well.

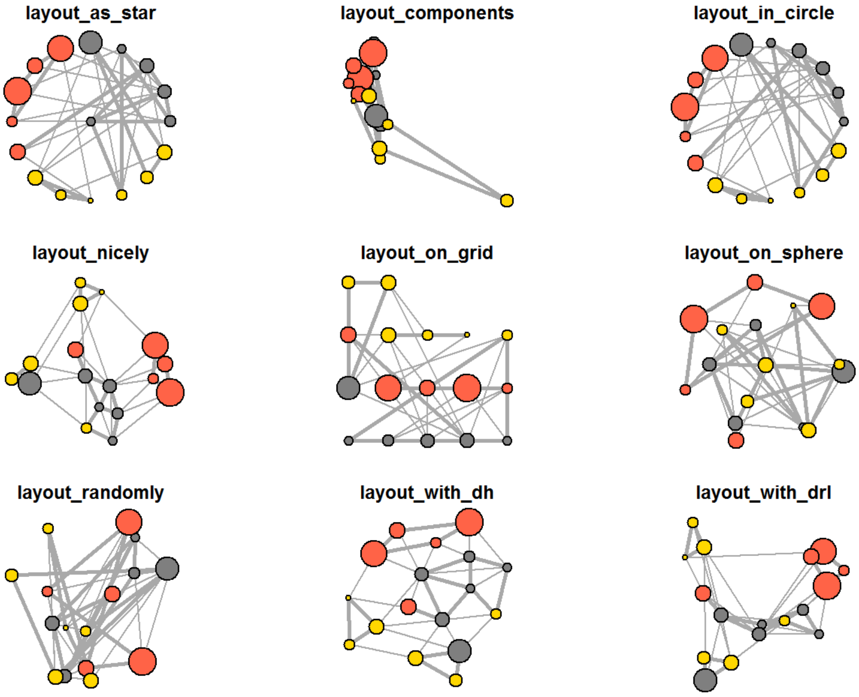


Figure 2.14 Igraph3D layouts [79]

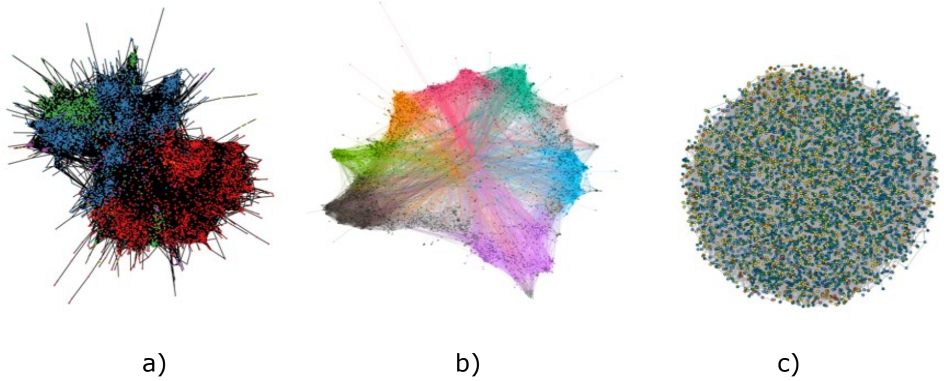
In addition to the 2D layouts, iGraph also implements an array of 3D alternatives. From simple ones such as a random 3D node dispersion or a sphere/grid dispersion to a more structured placement of nodes such as Fruchterman-Reingold or Kamada-Kawai.

- igraph_layout_random_3d — Random layout in 3D
- igraph_layout_sphere — Places vertices (more or less) uniformly on a sphere.
- igraph_layout_grid_3d — Places the vertices on a regular grid in the 3D space.
- igraph_layout_fruchterman_reingold_3d — 3D Fruchterman-Reingold algorithm.
- igraph_layout_kamada_kawai_3d — 3D version of the Kamada-Kawai layout generator

2.6.3 Defining the niche

All the aforementioned software packages and algorithms claim to be able to generate graphs of fairly large complex networks, leading to the natural question of whether there is one best suited to efficiently generate the most relevant and qualitative rendering of a biological system. Apart from that, there is also a need to determine the disadvantages of such tools and detect areas of improvement to be taken into consideration when building such software solutions. Consequently, an array of criteria and a scale have been proposed for the purpose of this evaluation.

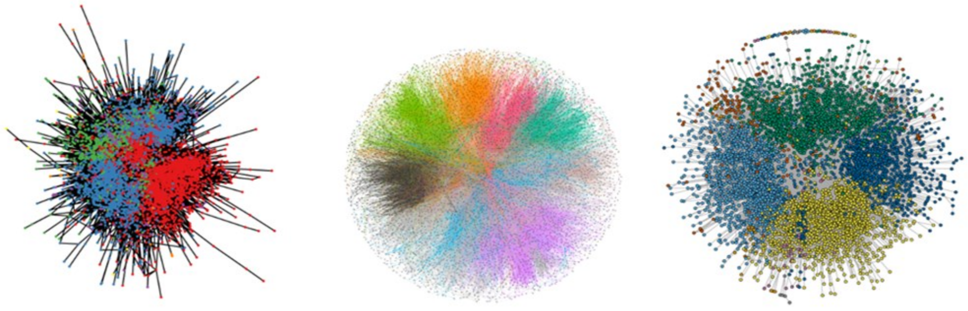
Firstly a fairly large biological dataset has been chosen for benchmarking purposes, to compare the performance of three visualization tools/software libraries: Python, Gephi, and R. The dataset dimensions consist of 5168 nodes and 42087 edges, and it represents a system of mice DNA genes, some of which are suspected to be responsible for the early emergence of genetic abnormalities or illnesses. Data is prefiltered with the help of a domain specialist. The selected batch of data is also considered a good test candidate due to its scale-free characteristics, which is a trait of most biology networks.



a) b) c)

Figure 2.15 FA2 Network layout rendering of a mice DNA data set in a) Python (NetworkX) b) Gephi c) R [13]

Fig. 2.15 shows the different types of visualizations generated by the benchmarked tools: NetworkX, Gephi, and R. Strictly from a visual point of view, there is already a major difference between the three renderings and Gephi seems to come up ahead in this aspect. The R figure (Fig 2.15 c) does not manage to group nodes into communities, thus ending up with a chaotic display of elements.



a) b) c)

Figure 2.16 FR Network layout rendering of a mice DNA data set in a) Python (NetworkX) b) Gephi c) R [13]

From an FR point of view, (Fig. 2.16) the R rendering manages to cope with the data size and the output with this run is more rewarding than the Python one.

In terms of performance, the results for the three tested tools are wildly different.

Alg.	**Visualization tools**		
	NetworkX	***Gephi***	***R***
FA2	11752.9 s	34 s	4376.916 s
FR	134.11 s	588 s	387.11 s

Table 2.1 Execution times for each tool [13]

Table 2.1 shows that although Gephi performs better than its two competitors when generating visualizations with the Force Atlas 2 (FA2) algorithm, there are still cases when it does not achieve the same rendering speed, like with the Fruchterman Reingold (FR) algorithm. The result is also dependent on the graph size and density, yet it is clear that other tools like Python's NetworkX can surpass it. In opposition, NetworkX behaves better in the case of FR, yet in terms of visual aspect (Fig. 2.16 a), the layout looks rather simplistic and does not seem to convey as much relevant information as the other two. Needless to say, despite R having a better FA2 rendering time than NetworkX, there is a clear failure in terms of cluster aggregation there (Fig. 2.15 c) and the user cannot identify separate components of the network.

Given the results in Table 2.1, it is clear that there is no single tool that can successfully achieve both a good performance and a pleasing visual aspect without compromising. This leaves way for tools that can better comply with the domain requirements in order to deliver better speed and visual appeal.

Criteria	**Visualization tools**					
	Python		***Gephi***		***R***	
	FA2	***FR***	***FA2***	***FR***	***FA2***	***FR***
Speed	1	2	2	3	2	3
Visual appeal	2	1	4.5	3	3	2
Relevance/ Utility	1	1	4	3	1	2
User friendliness	2	2	5	5	2	2
Customizability	5	5	3	3	5	5
ML capability	1	1	0	0	1	1
No programming skills required	0	0	1	1	0	0
Interactivity 0 (no) / 1 (yes)	0	0	1	1	1	1
Totals	5	5	16.5	15	9	10

Table 2.2 Visualization tools comparison [13]

Table 2.2 defines a wider range of comparison criteria and the tools are graded on a scale from 1 to 5 (1 representing the lowest grade and 5 being the highest) by a group of domain specialists in different.

In terms of speed, marks have been normalized proportionally to the run times of each algorithm. Values higher than 10.000 s correspond to a 1, while values ranging from 100 s to 10.000 s are the equivalent of grades 2 or 3. Any values smaller than 100 s will receive a grade between 4 and 5.

Visual appeal can be considered subjective, yet comparing the renderings in Fig. 2.15 and Fig. 2.16 there is no doubt that there are major differences that comfortably put Gephi at the top of the list. The Relevance/Utility metric suggests the extent to which researchers are able to use these tools for detecting patterns or using them for scientific discoveries. Usefulness is a reflection of clarity and relevant details, together with a solid aggregation of data. User-friendliness defines how easily a researcher would be able to use such tools and how intuitive the Graphical User Interface is. Customizability reflects the ability of each tool to adapt to the users' needs. Machine Learning (ML) Capability is another criterion that has proven to be very important in the recent past, as domain specialists seek to make use of ML algorithms more and more often. Yet this aspect requires the user to have programming skills, which is why the rating shows that Python and R have an edge, Gephi being an out-of-the-box tool. Finally, interactivity tells whether the researcher can interact with the generated visualization or not, and in this aspect, only Gephi and R comply.

In conclusion, there are multiple directions for improvement in the area of complex networks visualization for biology datasets, especially since neither of them manages to fully cover the majority (or a fair part) of the proposed criteria.

In addition, there is one extra criterion not included in the table, and that is: having a clear and visually measurable reference system included in the visualization. None of the assessed tools offer a fixed reference system or a clear ranking system which would allow the user to better understand the network structure, as well as grasp the dimensions of different elements or network components.

3. Visual Patterns in Bioinformatics

3.1. A Hybrid 3D Visualization Algorithm for Complex Networks

The present algorithm proposes a new hybrid approach to network layouts which aims to solve some of the most important needs identified with the existing layout algorithms in the field. The starting point when implementing such frameworks is considering what is the main type of network it addresses. Small networks (0 <|V| <~200 elements, 0 < |E| < ~500 edges) are fairly well suited for the majority of the readily available network tools. However, for medium to large networks (|V| > 200, |E| > 500), not all of them are capable of displaying data just as well [13]. A large number of nodes together with a high edge density pose quite a challenge from four major points of view:

- Structurally - or how to represent data in such a way that the human eye can perceive and grasp major interacting components
- Graphically - whether it is pleasant enough (subjective, yet follows some well-defined criteria), or whether it brings out key elements in the graph
- Resource consumption - is it dependent on the processing power, or does it convey good results even with a sub-optimal workstation
- Time consumption - whether the runtime is acceptable with regard to the quality it delivers. Good quality does not justify very long running times. As a result, a compromise needs to be made and the right balance should be established.

Given the above-mentioned hypotheses, the current algorithm has been defined as a solution for medium to large complex networks, an area that forces researchers to experiment with different types of layouts almost exhaustively, in the search for the perfect 'visual angle' to enhance and support intrinsic network dynamics and characteristics.

It is designed to solve a structural requirement, very important when dealing with large networks, it uses simple graphics without overcrowding the canvas and delivers an overall qualitative network layout without consuming excessive resources. Running time is also addressed in comparison with other tools, but without a major cost in terms of used resources.

This hybrid algorithm also aims to bring to the table an innovative compound feature, which cannot be found in other current layout types: a new concept in the display of data and at the same time 3D reference system as an important visual queue for graph interpretation.

The new layout concept proposed here started with the desire of adding more meaning to 2D and even 3D graphs. The core idea emerged from the concept of a 3D heatmap (Fig. 3.1), which is basically a 3D representation of a mathematical function. Beyond the simplicity of it, this way of visualizing information is strongly tied to the notion of depth (third dimension) but also to the concreteness sensation one gets when looking at data as a sort of geographic map. This type of layout involves hierarchy, subordination, and prioritization of the represented information according to certain criteria which can be either generic but are customizable enough to allow for specific domain-related restrictions.

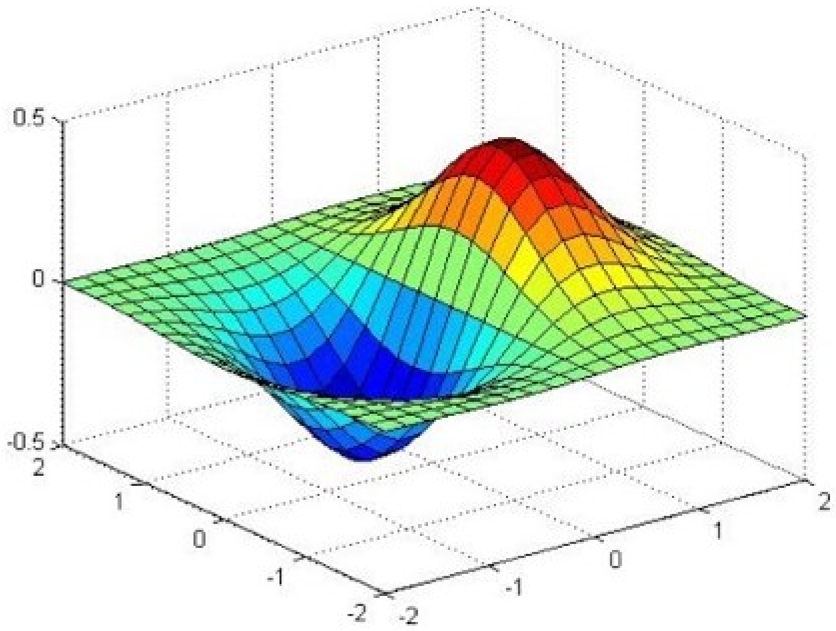


Figure 3.1 Sample 3D heatmap in Octave

The idea of a 3D network is indeed one step further than 2D graphs, yet only if it brings added value to the layout from a scientific point of view. In other words, the person looking at such an image should be able to use the extra dimension to determine and extract pieces of information that would not normally be obvious from a 2D perspective. This is what most other 3D network visualization tools lack, despite an appealing image. The Z axis does not mean anything if nodes are randomly placed in space but there is no means for the user to associate its placement to an additional characteristic whether it is domain-specific, or an aggregation of multiple metrics.

3.2. Layout structure

It is worth mentioning from the start that this type of layout is based on a predefined set of visual rules so that there is a specific targeted structure that all rendered layouts will comply with.

There are a couple of principles behind this visualization type.

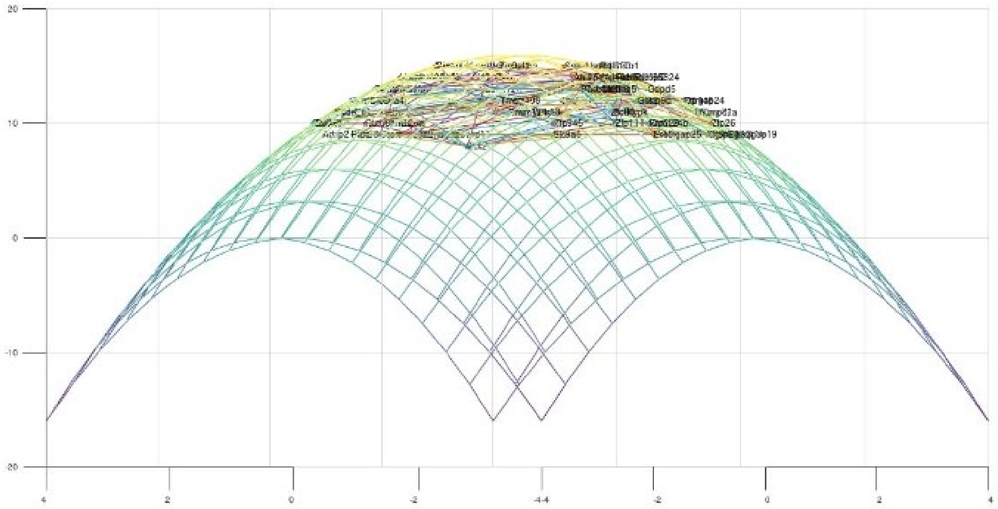
All graph clusters and elements should be well-spaced and the structure should be clearly visible. Creating a visual backbone of the network allows for a good overall comprehension of the image. Previous research shows that the prerequisite for a layout to be qualitative, it needs to check as many of these points as possible (at once) [13]:

- Easy to understand
- Reduce edge crossings
- Well-delimited clusters and nodes

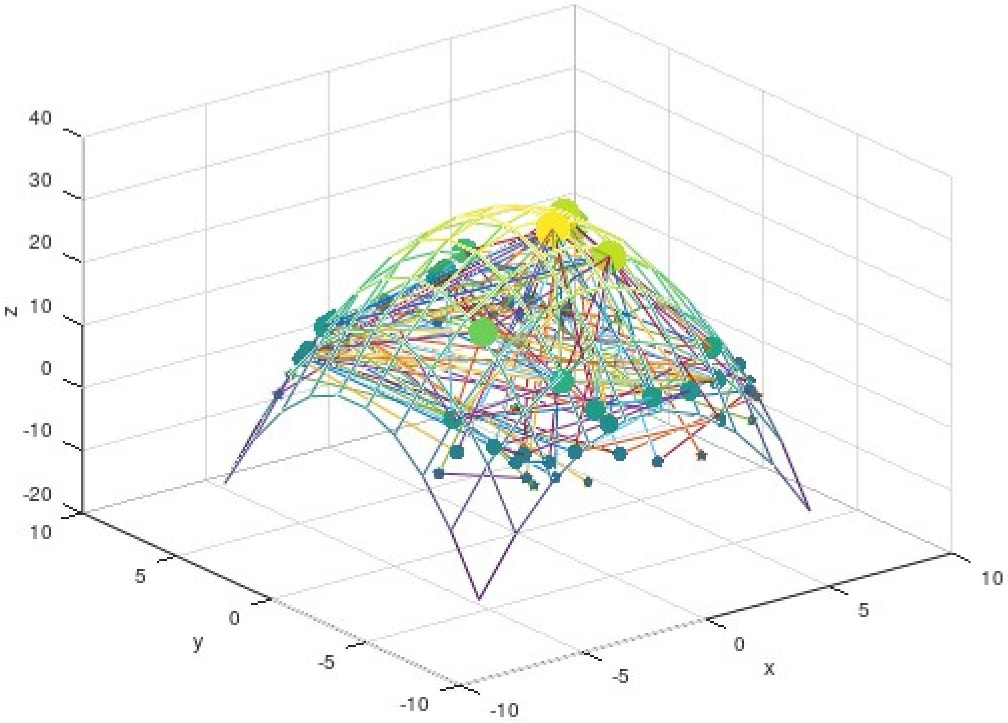
In terms of edge crossings, this problem has been long studied along with the Barycenter Heuristic[80], [81]. In practice, when implementing such a network visualization algorithm, one must tend towards such a desideratum, yet compromise is, most often than not, inevitable given the long list of functional and structural criteria these types of networks should respect.

3.3. Layout Algorithm Version 1

Based on the desideratum above, the first representation of such a network was decided. The complete network would be dispersed along a 3D inverse paraboloid (composed of concentric circles with decreasing radiuses). This would already be an improvement over its 2D version, as their hierarchy would be much clearer.



(a)



(b)

Figure 3.2 A 3D layout of a complex network rendered with the Hybrid algorithm V1

Although the paraboloid support figure gives the network a 3D structure, the figure seems too crowded. There are no clearly delimited clusters yet, but that is due to the early stage of development this algorithm was in. At this point there are two options for going forward, neither of them developed so far in the field:

1. Keep the whole network distributed across one single 3D paraboloid.
2. Split the network into its composing clusters and distribute each community across its own paraboloid, or even choose a hybrid layout containing both 2D and 3D clusters.

3.4. Layout algorithm version 2

The algorithm proposed here is based on two structural techniques used for plotting complex networks:

1. At a macro level, clusters are well-spaced and all of them have a circular/radial disposition. All clusters are placed around a center cluster - either a random one or preselected by the user.
2. At an intra-cluster level, all nodes are placed in concentric circles along the Oz axis. If clusters are only displayed in 2D, then the z-coordinate will be null for all of them and the result is a classical 2D layout.

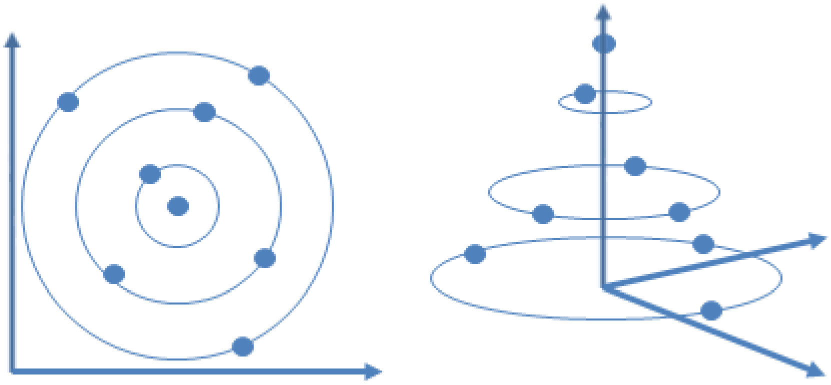


Figure 3.3 Cluster view a) 2D and b) 3D view

The rules for their individual placement are similar to a force-directed algorithm and are defined as follows:

1. First, identify the potential positions of a vertice in correlation with all its external connections. This means that each element should always have a position within its cluster which allows it to be reasonably well placed among community neighbor nodes, but at the same time fairly close to external clusters/nodes it is most connected to. To calculate these candidates for a node's intra-cluster position, we determine the weighted mass center of all other clusters (Cluster 2, Cluster 3 in this case) excluding the current one. The weights are proportional to the number of outgoing edges from point P to Cluster 2 and Cluster 3 respectively [Fig. 1]. Given that node P has two edges to Cluster 2 and three edges to Cluster 3, the resulting mass center will be closer to Cluster 2's origin. Once the mass center point is determined (*Mc(x_1,y_1)*), we draw an imaginary line defined by it and the cluster center (*O_1(x_2,y_2)*):

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) \tag{3.1}$$

On the other hand, as per the intended design, each community node will stand on one of the concentric circles of Cluster 1. The circle it will reside on is the circle corresponding to the node's importance within the cluster. In other words, the origin of the circle is the same as all other circles for Cluster 1 and the radius will be proportional to the weight of that node. This intersection of this line with the intra-cluster concentric circle to which the current node pertains will define two points: PP1, and PP2. These points constitute the potential positions starting from which we can further refine this node's position.

2. The result of the first step is a pair of geometric coordinates based on which intracluster positioning is adjusted. There are multiple options starting from here: either keeping the closest point with regards to external clusters or keeping the farthest point, provided that the node in discussion has little or no connections to external clusters. Once one of the two gets selected, its position is adjusted towards other intra-cluster adjacent nodes depending on the coexpression of the current node and nodes which have already been placed on the graph (i.e. they already have a stable position)

This process is repeated until every vertice of the graph has been positioned within its cluster.

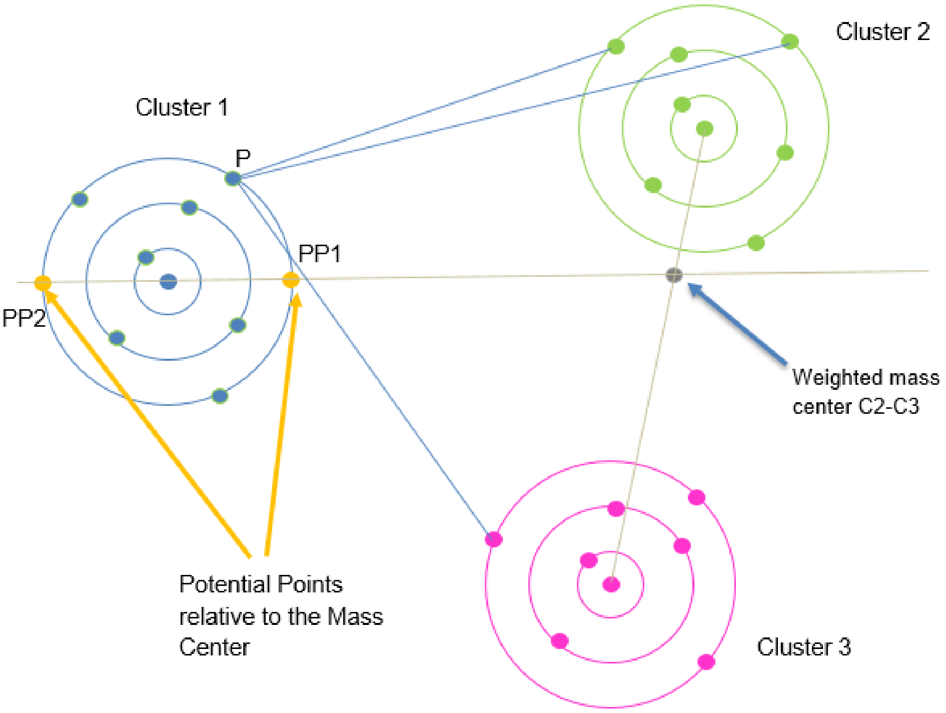

Cluster 2
Cluster 1
P
PP1
PP2
Weighted mass center C2-C3
Potential Points relative to the Mass Center
Cluster 3

Figure 3.4 Node placement phase – potential position calculation

3.5. Algorithm implementation overview

The proposed algorithm consists of three main stages, as per the following:

1. DB and data curation
2. Graph structure creation and spatial positioning
3. Layout generation

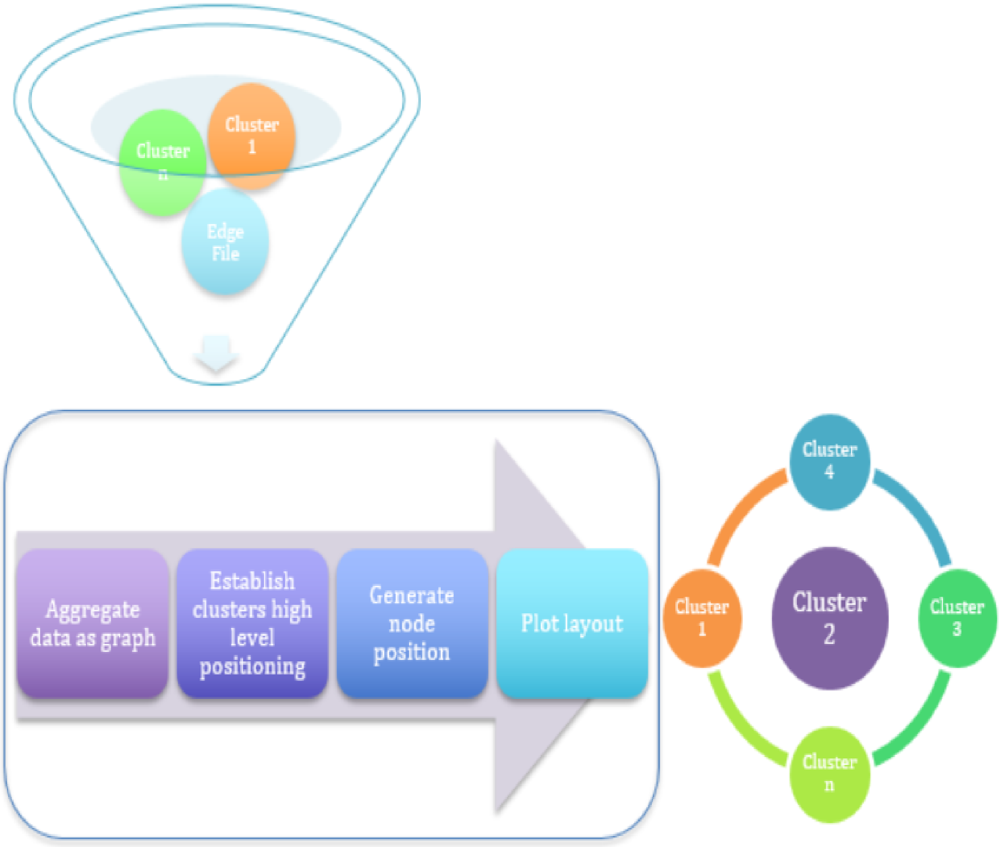


Figure 3.5 Algorithm overview

DB and data preparation refers to the raw data being used for the current experiment and fed to the proposed visualization algorithm. Most often used data sources are medical research institutes that regularly publish observation data resulting from their studies or experiments.

The most consistent part of the network layout generation process consists of the hybrid layout algorithm. This has been devised in the form of a process split into multiple steps, each one responsible for the aggregation, clusterization, and spatial

geometric placement of the network nodes, in this particular order. The main concept is based on a pipeline with the purpose of producing a useful enough graphical image so that the researcher can extract more valuable information from it.
Interestingly enough, layout generation - the last step of this process - is the most costly part of the algorithm, as it consumes the most graphical resources and it claims the majority of the total time spent.

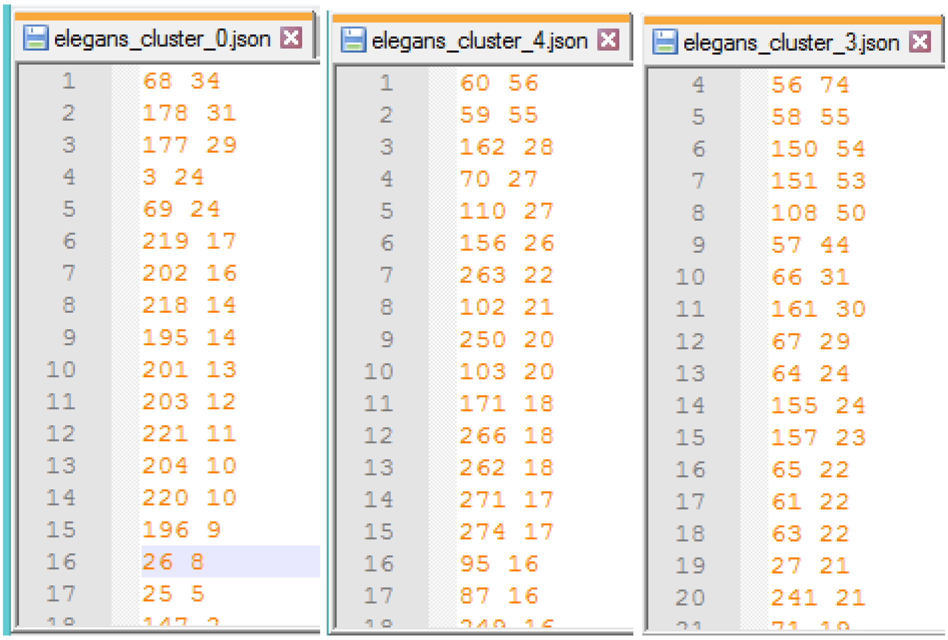
3.5.2 Data sets

The solution chosen for this research consists of several curated files containing graph metadata. This metadata describes the minimum features of a network so that one can recreate it using the given information.
There are two types of files being used:

1. Cluster files – multiple JSON files containing information about all cluster nodes, grouped by cluster. These also contain the number of connections each vertex has (e.g. node degree). All vertices in a cluster file are sorted in a descending order based on node degree/ Betweenness Centrality / or any other domain-specific metric such as the molecular mass of the cluster elements. The number of files is equal to the number of clusters within a network. The data is structured using the following format:
    - <Node_ID>␣<Degree/Betweenness Centrality/Any Meaningful Metric>
2. One edge file - containing all connections (links) between cluster nodes. No ordering is needed here. Depending on the type of network we are creating, the edges could be directed or undirected. Although for the purpose of this research, mostly undirected connections are being used.

The clusterization of the networks analyzed in the current thesis has been achieved with the use of the Louvain community detection algorithm. This is the exact same algorithm used by Gephi and has become increasingly popular among clustering methods. This is used as an external tool for the prefiltering and arrangement of the data and must be run as a pre-step to obtain the cluster files necessary as input material. The result of running this algorithm is a series of cluster files containing only vertices pertaining to the same cluster.

Another important aspect here is the naming convention established for all cluster files and follows the following pattern: *[graph_name]_cluster_[cluster_no].json*. This is how the application identifies all cluster files pertaining to a network name given as input.



elegans_cluster_0.json

```
68 34
178 31
177 29
3 24
69 24
219 17
202 16
218 14
195 14
201 13
203 12
221 11
204 10
220 10
196 9
26 8
25 5
```

elegans_cluster_4.json

```
60 56
59 55
162 28
70 27
110 27
156 26
263 22
102 21
250 20
103 20
171 18
266 18
262 18
271 17
274 17
95 16
87 16
```

elegans_cluster_3.json

```
56 74
58 55
150 54
151 53
108 50
57 44
66 31
161 30
67 29
64 24
155 24
157 23
65 22
61 22
63 22
27 21
241 21
```

Figure 3.6 JSON Cluster files

Future potential improvements could consist of relational or non-relational databases which could improve data filtering and manipulation by queries tailored to the needs of the user. However, this aspect implies either using some of the existing databases made available by research institutes or entities, or curating some of the file-based medical information. The lack of consistency between researchers in terms of storage formats can lead to more work being done in order to obtain a consistent and wholesome data foundation.

3.5.3 Data sets characteristics

For this research experiment, the data sets have been picked from the biology domain, and they belong to the C. Elegans nematode [82]. More precisely, this is a collection of neurons and they are represented as a complex network, given that the nature of their interaction resembles an interconnected and interactive system.

These complex networks shall be defined as follows:

G = (V, E) where V is the set of C. Elegans neurons and E is the collection of interconnecting edges. In this case, the total number of nodes and respectively total number of edges are |V| = 279 and |E| = 2287.

The average degree is then calculated to be a=16.39 considering that it is an undirected graph, and as such, the appropriate formula for the average degree is a=2*|E|/|V|.

The degree distribution scattered graph (Fig. 3.7) for this specific data set shows that the largest part of the edges is split between a small number of vertices, rather than being evenly distributed. What this means in terms of a visual display, is that the high-value nodes (the ones with the highest degree) will have a cluttering effect wherever they reside among the clusters, all the more if they are part of the same community. And thus, the final layout will be difficult to make sense of, for the human eye.

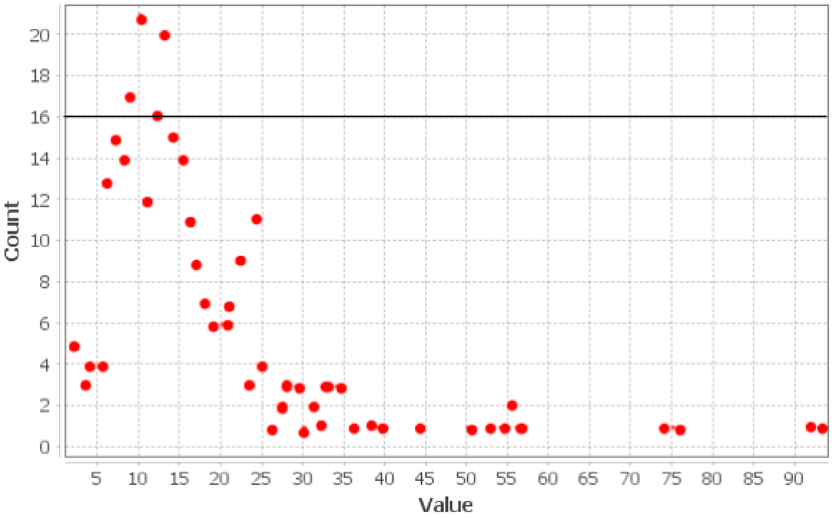


Figure 3.7 Node degree distribution for a C. Elegans network [83]

Here, the Louvain community detection algorithm was used to group vertices into clusters, being one of the popularly used algorithms of its kind [59]. Gephi networks layout tool uses the same algorithm. Thus, the resulting number of communities (|*M*|) is in this case |*M*| = 5.

To determine the complexity of the graph and the „crowdedness" we calculate the average number of vertices per community, respectively the average number of edges per cluster as follows:

$$avg(|V|/|M|)=56$$
$$avg(|E|/|M|)= 457.4$$

What these numbers say about the structure of the network is that its layout is going to be inevitably quite heavy, yet it seems that links are distributed fairly equally between vertices from separate clusters.

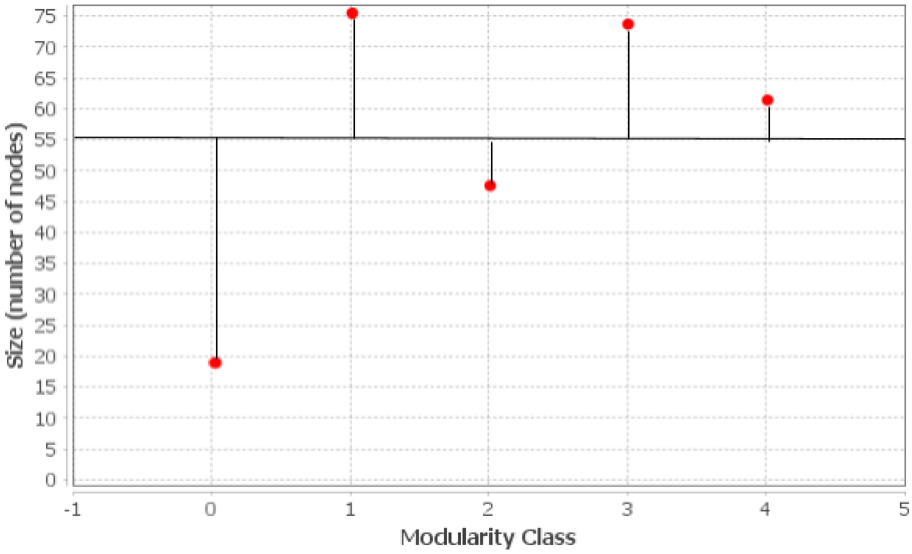


Figure 3.8 Vertex distribution per community (Modularity class) [83]

Since it is estimated that the resulting network layout will be quite congested this aspect will make it complicated for any viewer to distinguish separate elements or assess the importance of single vertices within the whole picture. As a result, the algorithm should be able to adapt to the network specificities or be able to disperse vertices based on the density of nodes in certain areas or clusters. This type of coefficient should not be a global one, but it should be a community-dependent one. In terms of inter-cluster interaction or closeness, this should also take into consideration the number of connections two clusters might have, and increase or decrease the distance between them to accommodate the large number of edges that need to be plotted and make them easily visible.

3.6. Algorithm steps

The layout algorithm is, as previously mentioned, a pipeline type of operation composed of multiple steps designed to process data and send it further to the next step in line.

3.6.1 Input parsing

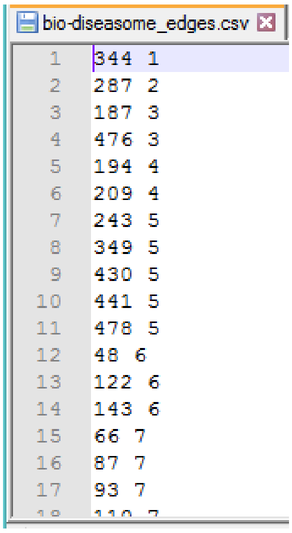
The first stage is in charge of parsing the input files (cluster files) one by one and creating a graph structure object containing all nodes grouped by cluster, as well as storing metainformation about each node.
This vertice meta information consists of:

- node id
- weight: could be node degree, node betweenness centrality, or any discriminating metric
- clusterId - the number indicated in the file name, corresponding to the cluster a node pertains to
- color - specific color associated with each cluster
- potentialPositions - an array of intermediary positions associated with each node to be used in determining the final placement
- position - final position of each node

Clusters attributes are also stored at this point:

- id - unique cluster id associated with each individual cluster;
- weight - cluster weight is equal to the sum of all node weights in the cluster
- nodesNbr - number of nodes in each cluster;
- avgDegree - average degree;
- maxCoEx - maximum coexpression;
- maxZ - maximum height (in 3D) per each cluster

At this stage, the adjacency file (edges file) is parsed into an adjacency matrix. The edge file respects the following naming convention: *[graph_name]_edges.csv*.



```
344 1
287 2
187 3
476 3
194 4
209 4
243 5
349 5
430 5
441 5
478 5
48 6
122 6
143 6
66 7
87 7
93 7
110 7
```

Figure 3.9 Edge file sample

3.6.2 Covariance based positioning

To determine the position of a vertice within its community, a hierarchical criterion is defined to help with node placement in a cluster.

The idea behind vertice placement is based on the 3D paraboloid shape. A 3D paraboloid consists of an infinite number of concentric circles of various radiuses which, placed one on top of the other, form the resulting shape.

Based on the newly created adjacency matrix, a co-expression matrix is created. This coexpression table computes the similarities between nodes of the same cluster. The covariance of two nodes is an indicator of how much two nodes change at the same time. Applied to this concrete example, the more neighbors they have in common, the higher the covariance. The resulting values are not (necessarily) standardized and can greatly vary.

Thus, we define covariance as follows.

Given *x, y* two nodes pertaining to the same graph *G (V, E)*.

$$A = \{z \mid z \in V \wedge \{x,z\} \in E \wedge \{y,z\} \in E \wedge x \neq y \neq z\} \qquad (3.2)$$

Where *A* is the set of nodes connected to both x and y.

Then we can define the covariance of x and y as:

$$cov(x,y) = |A| \qquad (3.3)$$

In other words, the cardinality of the set consisting of vertices common to both *x* and *y* represents the covariance of the two.

3.6.3 Vertex 3D positioning

In calculating a node's potential position within its community, three important factors are being taken into consideration:

1. The number of connections a node has to all other clusters.
2. The positioning of connected clusters excluding the one the current node is part of.
3. The node's connectivity within its own community.

Let

$$X_{i,k} = \begin{cases} 1 \; if \; \exists\{i,k\} \in E \\ 0 \quad otherwise \end{cases} \qquad (3.4)$$

where *community(i)* != *community(k)*

Then

$$D_i = \sum_{k=1}^{n-1} x_{i,k} \qquad (3.5)$$

Is the number of edges from vertex *i*.

We define the weighted center of all communities excluding C_i as follows:

$$G = [w_1 * x_1 + w_2 * x_2 + \cdots + \; w_n * x_n, w_1 * y_1 + w_2 * y_2 + \cdots + w_n * y_n] \qquad (3.6)$$

$$G = \sum_{i=1}^{n} w_i * x_i \, , \sum_{i=1}^{n} w_i * y_i \qquad (3.7)$$

Where

$$\sum_{i=1}^{n} w_i = 1 \qquad (3.8)$$

Once *G* has been established, then it will stand as a point of reference for the positioning of node *i* in relation to its outgoing connections.

Let *O1G* be the line defined by points *O1* (Cluster center 1) and *G*. Then:

$$O1G\colon \frac{y - y_{O1}}{y_G - y_{O1}} = \frac{x - x_{O1}}{x_G - x_{O1}} \qquad (3.10)$$

where *O1* and *G* are distinct points.

In addition, each vertice within the cluster must follow the following rules in terms of inner cluster positioning:

1. Establish feature/metric associated with the Z axis - this hierarchy will help determine a node's vertical priority. Sort nodes by value, in descending order.
2. Consider each distinct Z value (Z_v) as the squared radius of an imaginary circle placed in a 3D space as follows:

   $$(x - O_x)^2 + (y - O_y)^2 = r^2 \qquad (3.11)$$

   Where $r^2 = Z_v$

   - Given the known radius, the position of a point $P(x_p, y_p)$ on the previously defined circle should comply with the circle equation:

     $$(x - O_x)^2 + (y - O_y)^2 = Z_v \qquad (3.12)$$

3. Calculate the intersection of *O1G* and circle *Ci*:

$$OiG\ Ci = \{PPi_1, PPi_2\} \qquad (3.13)$$

Based on the two potential points of intersection, and the covariance matrix the following choice is made: if covariance between node *i* and previously placed node i-1 (more important node, higher Z value) is greater than the average covariance per cluster (Ci) then keep *PPi1*. Else keep *PPi2*.

$$P_i = \begin{cases} PPi_1,\ cov(V_i, V_{i-1}) > avg(cov(C)) \\ PPi_2,\ cov(V_i, V_{i-1}) \leq avg(cov(C)) \end{cases} \qquad (3.14)$$

3.6.4 Position adjustment phase

Once the potential starting position has been chosen (PPi1 or PPi2) the placement of the node on the circle is adjusted by an angle proportional to the node's betweenness centrality within the graph. If the centrality value is smaller than average outgoing degree per Ci, then move the node farther apart from other clusters. Otherwise, bring it closer to the middle of the graph.

In the position adjustment phase, when establishing the final node 3D coordinates, the density of nodes within a certain circle area is considered. Given that at some point multiple vertices in cluster C1 might get roughly the same potential position considering their connections to the other external clusters, there may be cases where one circle area becomes too crowded. And while the idea of staying close to external connection remains, the internal positioning within the cluster changes.

Thus, given the previous node position is always stored (P_{i-1}), when calculating the position for the current processed node P_i (assuming both nodes pertain to the same cluster) the coexpression between the two is compared to the average coexpression within the cluster. The closer these values are, the closer the final positioning will be. Otherwise, node P_i coordinate adjustment will begin from PPi2 (the potential position calculated in the previous step and the one farthest away).

Having decided the starting position to adjust (PPi1 or PPi2), the angle between the two is calculated first with the help of the cosine law. Assuming the two points could have a different z-index, in other words pertaining to two different circles, the cosine of the angle between the two can be calculated as follows:

$$prevR = euclideanDist(Po, P_{i-1}) \qquad (3.15)$$

$$currentR = euclideanDist(Po, P_i) \qquad (3.16)$$

$$cosA = \frac{(prevR^2 + currentR^2) - d(P_i, P_{i-1})}{2 * prevR * currentR} \qquad (3.17)$$

Where *prevR* is the radius of the circle where the previous point resides on, *currentR* is the radius of the current circle and current point, and *cosA* is the cosine of the angle formed by the two points.

Based on the obtained cosine, the angle ɑ can be deduced by applying an arccosine function $\alpha = \mathrm{acos}(cosA)$. The angle is then increased with a coefficient that is directly proportional to the difference between the two node’s coexpression and the maximum coexpression within that cluster. In other words, the more interconnected the two nodes are, the closer they should be placed to one another.

$$simCoef = \frac{maxCo - Co(P_{i-1}, P_i)}{maxCo} \tag{3.18}$$

In the above equation, simCoef is the normalized similarity coefficient (*simCoef*) defined as the subunitary difference between the maximum coexpression value within the cluster (*maxCo*) and the current coexpression value between P_i and P_{i-1}.
Thus, the obtained angle is increased with the similarity coefficient defined by their coexpression.

$$\alpha' = \alpha(1 + simCoef) \tag{3.19}$$

Having obtained a new angle for the node repositioning, the new coordinates of P_i are computed:

$$xP_i = (xPP_i - xP_O) * \cos(\alpha') - (yPP_i - yP_O) * \sin(\alpha') + xP_O \tag{3.20}$$

$$yP_i = (yPP_i - yP_O) * \cos(\alpha') - (xPP_i - xP_O) * \sin(\alpha') + yP_O \tag{3.21}$$

where $[xP_i\ , yP_i]$ are the new coordinates of P_i, $[xPP_i,\ yPP_i]$ are the coordinates of the predetermined potential position (one of two), and $[xP_o, yP_o]$ are the coordinates of the cluster center.

3.6.5 Edge plotting

Once node positions have been calculated across all clusters of the networks, and the color palette has been chosen (each cluster is assigned a color from the color palette), the edges are plotted in a 3D space.
There are multiple approaches that can be taken in terms of plotting:

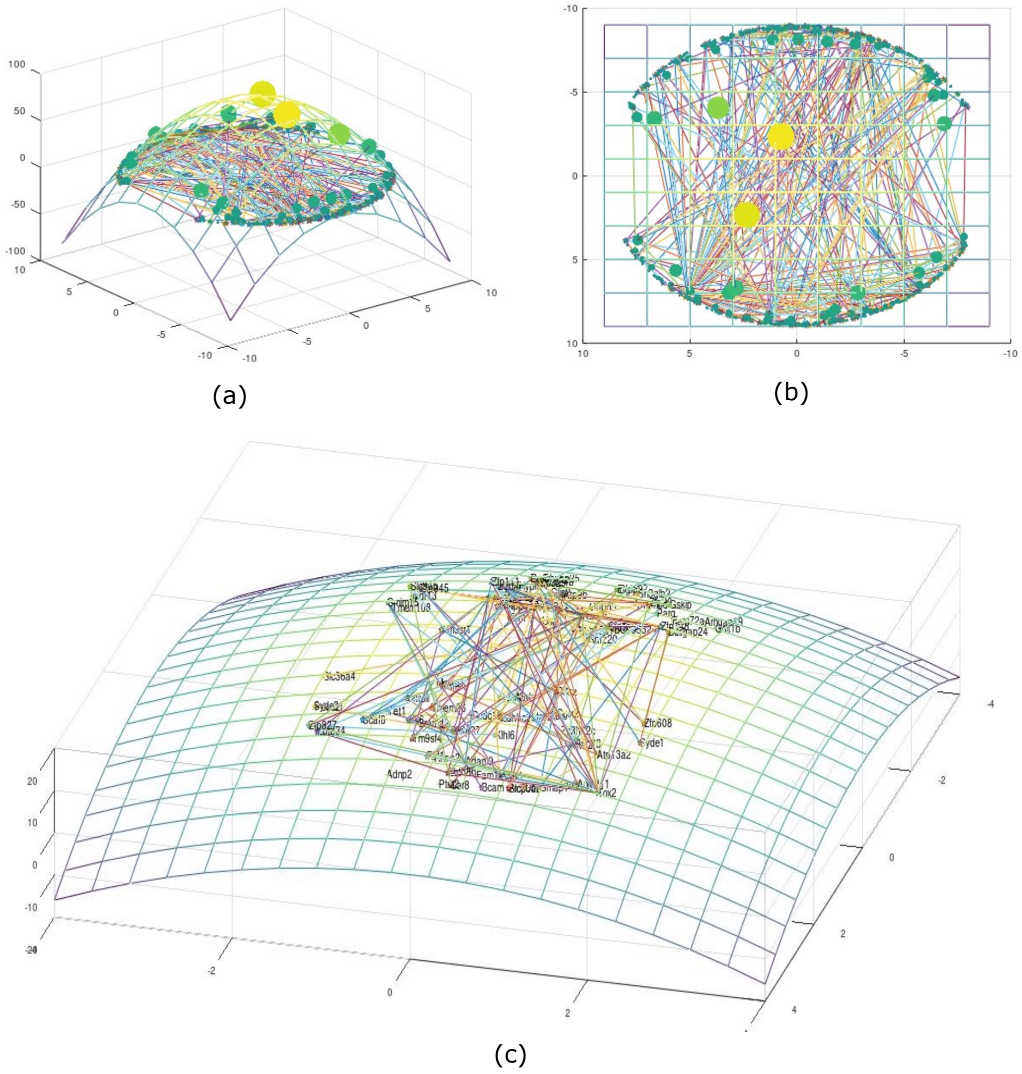
1. Plotting all clusters in a 3D space against its paraboloid
2. Plotting only one cluster at a time in 3D while keeping all other clusters in 2D
3. Displaying only vertices in 3D while keeping edges on a 2D plane and connecting all the node’s 2D projections together
4. Removing inter-cluster edges and only showing intra-cluster connections

Depending on the desired outcome, one of these approaches can be used due to the algorithm being highly customizable.

3.7. Results

With the first version of the algorithm, the following graph sample layouts were obtained.

3.7.1 Hybrid 3D algorithm V1



(a)

(b)

(c)

Figure 3.10 3D Layout algorithm V1 – all nodes distributed across one single paraboloid (a) paraboloid view from the side[84] (b) 2D view of the paraboloid and nodes[84] (c) Zoom in on the 3D network from a side, with node labels

All nodes are placed within the same 3D space, and the same paraboloid (Fig. 3.10). They are still placed in concentric circles, yet there is no grouping nor is there a specific order to them.

One aspect which gives an idea of how important nodes are within their network is node dimension, color as well as position.

The closed a node is to the center of the paraboloid, the higher a degree it has, and from a maximum degree perspective, it can be considered the highest ranked within the whole network.

In terms of color, the lighter the color (bright yellow) the higher a node is ranked, while the darker ones (dark green) assume a less important role within the community.

Node dimension (bubble size) also gives a visual indicator of the same factor and is directly proportional to node degree. Node labels can also be displayed, however, at a certain point, this feature might become undesirable, given the number of elements in a network layout.

3.7.2 Hybrid 3D algorithm V2

With the second version of the algorithm, communities are all separated, keeping one of them in a 3D space (nodes distributed across the paraboloid) while all others are placed around the cluster of interest. Fig. 3.11 shows the first stage of the algorithm (node rendering) part of them in 3D, and the other part of them in 2D.

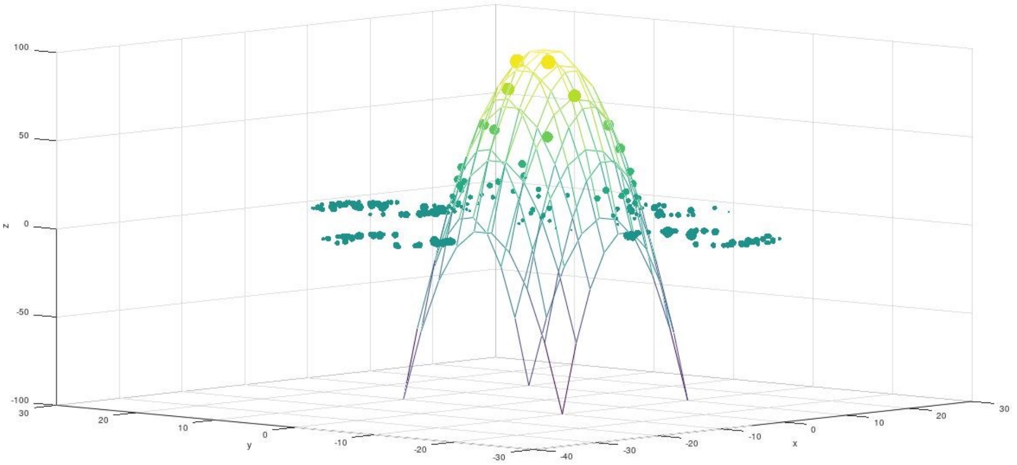
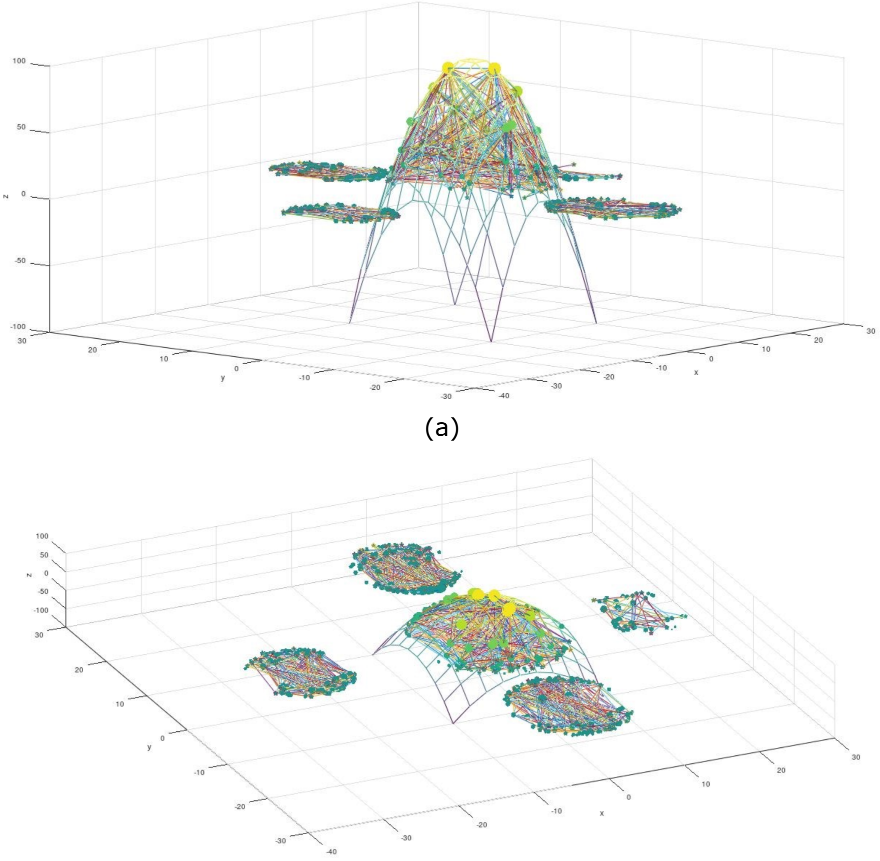


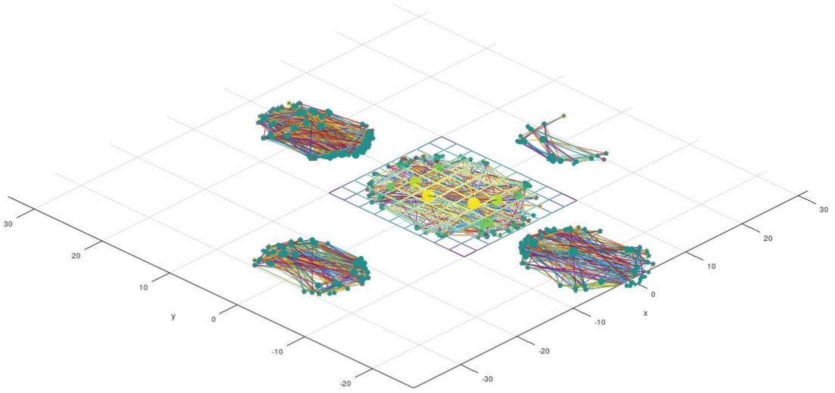
Figure 3.11 Hybrid 3D layout algorithm V2 (E. Coli network) – stage 1 (only nodes) [83]

The chosen cluster is either preselected by the user or it can be a random one if no preference is specified. The gradient of the nodes also changes as they get placed higher up on the top of the paraboloid. The colors can be changed depending on the selected color palette. Vertex size is proportional to their degree – the fewer connections, the smaller the node bubble. Conversely, the more links a node has, the bigger it appears within the layout.



(a)

(b)



(c)

Figure 3.12 Hybrid 3D Layout algorithm V2 (E. Coli network) – Stage 2: all communities with intra-cluster edges (a) 3D view from the side [83] (b) 3D view from above (c) 2D view from above

Figure 3.12 (a,b,c) shows the second stage of the Hybrid algorithm (V2) where intra-cluster edges are plotted. For now, communities are not interconnected, and it is possible that if the user wishes, they can be kept like this. There is also the option of zooming in and out of the clusters or changing the color of cluster nodes.

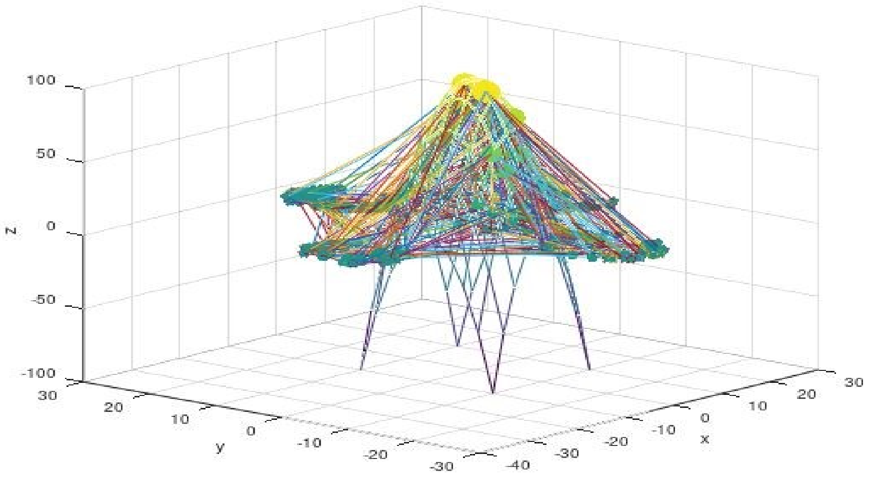
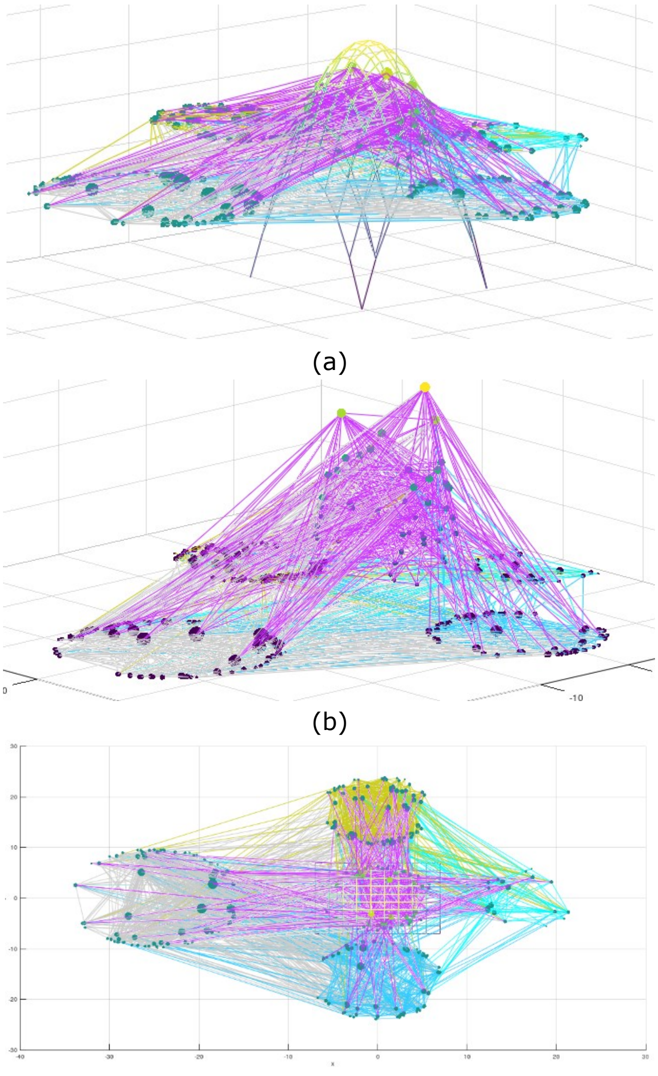


Figure 3.13 Hybrid 3D Layout algorithm V2 (E. Coli network) – Stage 3: all communities with intra and inter-cluster connections [83]

Figure 3.13 above shows the final layout of the network with all clusters interconnected. Given this is quite a dense network, it is expected that the final result will be quite busy. However, this 3D model can be zoomed in and manipulated so as to better distinguish the network elements.



(a)

(b)

(c)

Figure 3.14 3D Hybrid Layout of the C. Elegans network [83] (a) 3D view, communities have different colors (b) 3D view layout, no paraboloid support for main cluster (c) View from above

A visual improvement to the Hybrid layout consists of different colors for the clusters, so as to differentiate them better from one another (Fig. 3.14). This also helps for an improved structure of the entire graph.

While sometimes the supporting paraboloid is an interesting addition to the layout, other times, when dealing with large networks, any extra element taking up space (even 3D space) might make the design difficult to understand and could create an opposite efect. That is why this feature does not always have to be present, but it is there merely to show the logic behind node placement.

Still, even with all the visual adjustments, one cannot help but wonder if there are better, clearer ways to approach the representation of a complex network.

Another attempt at a more exotic approach of displaying a graph might be one where all clusters are spread in a 3D space, all of them distributed over different meshes – emulating a continuous platform underneath the nodes. While this may be an interesting approach to smaller graphs, this is again something that might not work for large and highly interconnected networks.

Figure 3.15 below shows an alternative 3D layout where clusters are all plotted on 3D meshes. Again, the cluster of interest can be placed higher than others, or they can all be placed at the height dictated by their node degrees.

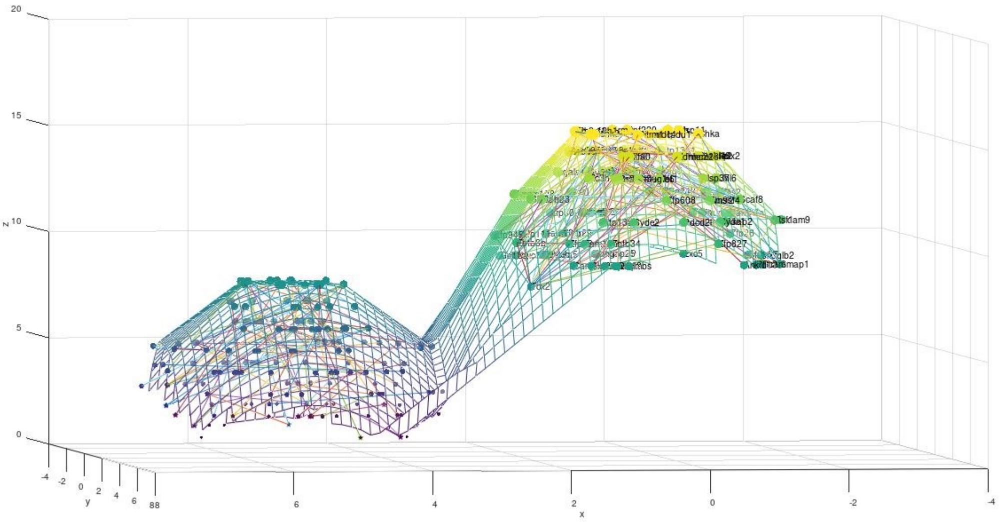
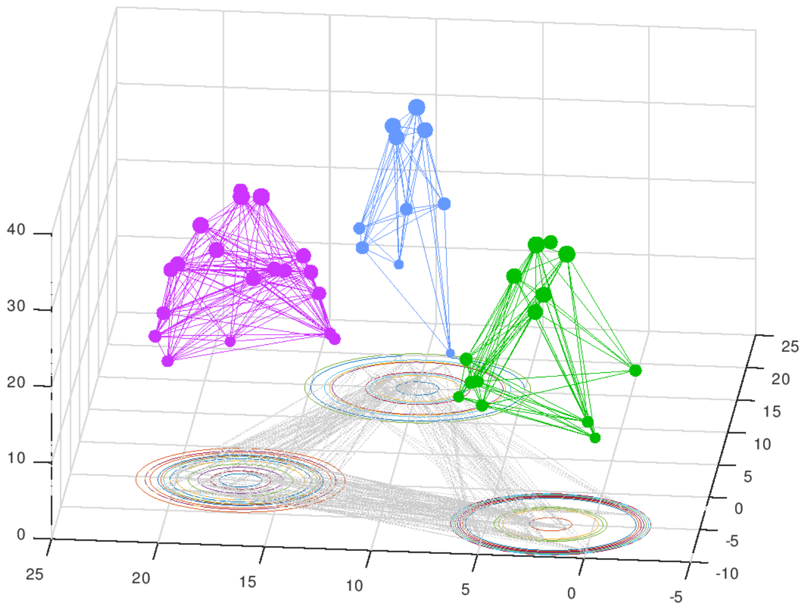
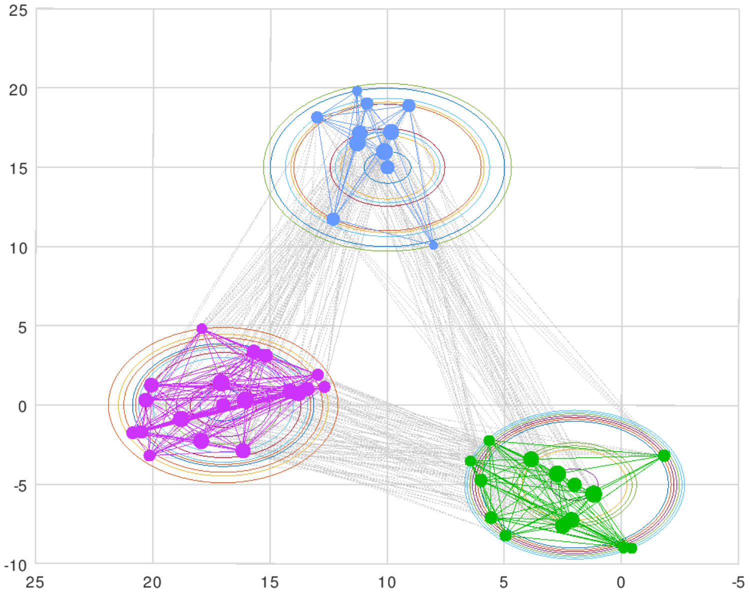


Figure 3.15 Alternative Hybrid 3D Layout (E. Coli network - partial) – all clusters distributed on 3D meshes

Another possibility to unclutter the resulting layout is, as shown in Fig 3.16 (a) and (b) below, to plot all communities and intra-cluster edges in 3D, yet keep all inter-community edges on a 2D plane.


40
30
20
10
0
25
20
15
10
5
0
-5
25
20
15
10
5
0
-5
-10

(a)


25
20
15
10
5
0
-5
-10
25
20
15
10
5
0
-5

(b)

Figure 3.16 Hybrid 3D Layout V3 - all clusters in 3D with intra-community edges, 2D inter-cluster edges

That way, the network elements seem more evenly spread out, and the whole ensemble is looking more loosely coupled. Inter-cluster edges can also be colored according to their communities, however, if the aim is to put an accent on the individual clusters, then there is no use in adding too many unnecessary features which might distract the user from the points of interest.

3.8. Discussion

The 3D Hybrid Layout proposed here aims to be a viable option apart from the already well-known visualization tools or algorithms such as Gephi, NetworkX, NodeTrix, etc. This type of 2D and 3D combined approach is quite uncommon among network layout software and proposes a different approach in emphasizing different network properties, elements, and communities. This comes in handy, especially with biological networks where the maximum degrees, as well as average degrees, are high.

The advantage of such a technique consists in adding a reference system (xOyz axes) which aims to help the viewer better understand the difference in densities, size, and the structure of communities altogether within a 3D gridded space. This does not happen with the most common visualization tools (Gephi – Fig. 3.17) where visual appeal takes precedence over functional aspects, leaving behind valuable information.

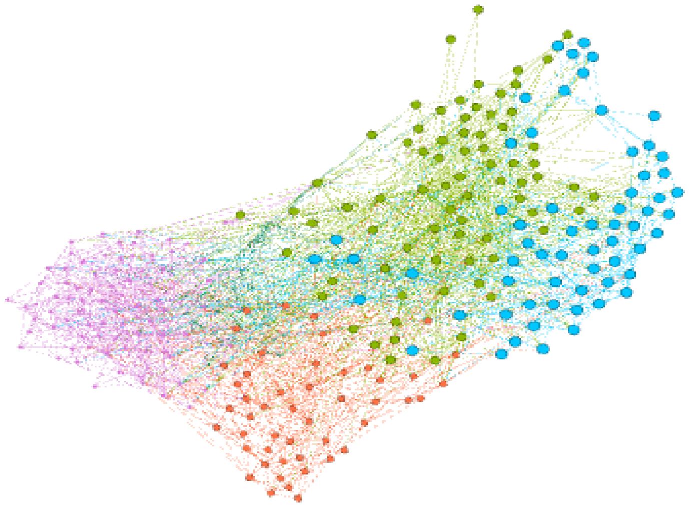


Figure 3.17 Gephi (Force Atlas 2) layout of the C. Elegans network [83]

In the case of C. Elegans network, there is an important difference between the Gephi layout and the Hybrid 2D/3D one. While in Figure 3.17 nodes can be seen grouped into fairly irregular clusters, which might as well be considered a 2D plane, Figures 3.14. and 3.16 produce a better structure of the same network. Obvious visual cues include node color and dimension, as well as placement with regard to all other nodes and communities.

A sample rendering of the same network, but this time produced by Python's Force Atlas 2 algorithm, shows a very cramped graph, where different communities can be difficult to visually separate from one another (Fig. 3.18).

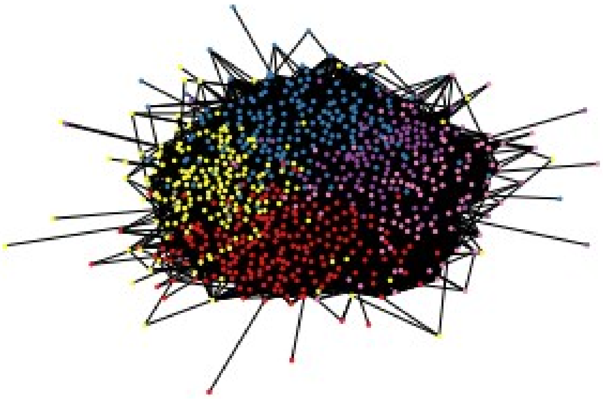


Figure 3.18 Python rendering with Force Atlas 2 of the C. Elegans network

On the other hand, with the Hybrid approach, there are more visual cues that allow for a better understanding of the whole ecosystem: a third degree of liberty (the *Oz* axis), as well as the entire reference system which stands as a ruler and a clear and absolute reference point. This way, clusters are not only comparable to one another but can be interpreted on their own as well. While typical traits such as node size could be an indicator of node degree, the third dimension (z coordinate) could stand for a totally different metric, either one of the standard ones (betweenness centrality) it could just as well be an entirely new measurement, user-defined. This is not possible with the other types of tools, as they are missing the extra dimension.

When it comes to performance, the run time of the proposed Hybrid algorithm can be seen in Table 3.1, for the proposed data set.

Time/ Phase	**Layout computation time**		
	Position calculation	***Nodes render***	***Edges render***
Avg time	0.035 s	0.6 s	23 s

Table 3.1 Break down of Hybrid layout rendering time for a C. Elegans network [83]

This gives a total run time of T_{hybrid}=23.635 s. Comparing this time with Gephi’s layout render time is slightly challenging given that the FA2 algorithm does not give the user the possibility of measuring each step of the process. However, the whole generation time for the same network has been measured to T_{Gephi}=10s.

The algorithms have been tested against a wider range of networks so as to determine each of their strongest and weakest points. These sample networks have different characteristics in terms of network size, density, number of communities, etc. They have been named Set1, Set2 and Set3 (Table 3.2) and they all represent biological processes (mouse DNA gene sets).

	#nodes	#edges	Density	Avg deg	Network diameter	Modularity	#clusters
Set 1	768	15786	0.054	41.109	6	0.329	9
Set 2	2393	5705	0.002	4.768	9	0.538	22
Set 3	5168	42087	0.003	16.288	11	0.759	38

Table 3.2 Test data sets (biological networks)

Average run times have been recorded for the above data sets, with all 3 algorithms, per 100 renditions.

Layout tool	**Average Time per 100 renditions**		
	Set1	**Set2**	**Set3**
Python (NetworkX)	921.3	1843 s	11523.95 s
Gephi	20 s	142 s	34 s
Hybrid 2D/3D	155 s	63.37 s	439 s

Table 3.3 Performance comparison between Python's FA2 layout algorithm, Gephi (FA2) layout algorithm, Hybrid layout algorithm

Table 3.4 shows a comparison between this Hybrid layout, Gephi's Force atlas 2 visualization and Python's NetworkX FA2 algorithm including a number of both subjective and objective criteria. All criteria have been graded on a scale from 1 to 5, where 1 is the lowest score and 5 is the highest.

Criteria	**Tools**		
	Gephi (FA2)	***NetworkX***	***Hybrid 2D&3D***
Speed	4	2	3
Visual appeal	4.5	2	3.5
Relevance/ Utility	3.5	2	4
Node distribution	Semi 3D	2D	2D + 3D

Criteria	**Tools**		
	Gephi (FA2)	***NetworkX***	***Hybrid 2D&3D***
Fixed reference system 0(no)/1(yes)	0	0	1
Allows user defined metrics	0	1	1
Customizability	4	3	5
ML Capability	0	1	1
User Friendliness	5	0	3.5
Interactivity	1	0	1
Totals	22	11	23

Table 3.4 Hybrid layout comparison to Gephi and NetworkX

The chosen criteria were graded based on empirical evaluations. With regards to speed, the hybrid algorithm is still behind Gephi, yet both of them are considerably faster than the Python one. Visual appeal is, admittedly subjective, yet its grading has been calculated based on the opinion of a group of subjects from different domains (10 data specialists, in the medical and engineering field). The hybrid algorithm provides more visual clarity than NetworkX, however, when compared to Gephi, it seems slightly less structured. Still, looking at Fig 3.14 and Fig 3.16 above, there is more depth to the Hybrid one, compared to Gephi.

Relevance/Utility is the category where the proposed layout algorithm aims to bring a plus over the others. The fact that it can convey more visual queues than its competitors gives it an advantage. In terms of node distribution, or better yet, graph dimensions, it is clear that the only one having a real advantage of an extra dimension is the hybrid approach and this applies to the next criteria as well (Reference system).

Allowing user-defined metrics is a plus which only programmatic approaches like Python or the Hybrid algorithm (Octave) have.

In terms of customizability, Gephi's abilities only go up to a certain point, however, being an out-of-the-box tool, it is not meant to let users change any definition of the set of metrics already defined within it. With NetworkX this can be done, however still, being a predefined type of algorithm, its flexibility only lets the user modify certain aspects.

Machine Learning (ML) capabilities are definitely an aspect desired by most tools nowadays, however Gephi is not yet at that point. NetworkX can be adjusted accordingly and so can the Hybrid algorithm. User friendliness is not one of the strong points of any programmatic solution, and thus, Gephi gets a maximum score in this criterion.

With interactivity, both Gephi and the hybrid algorithm allow users to manipulate the generated graph, while NetworkX offers a static one. A plus here can be considered the Hybrid algorithm’s capability of rotating the network around the three axes (Ox, Oy, Oz) which cannot be done with any of the other ones, as well as zooming in and out of it.

4. Modeling Numerical Patterns

4.1. A novel method of Computer Tomography image interpretation

Diffuse interstitial lung diseases are a category of lung pathologies that despite their similarities, can be treated differently if diagnosed correctly in due time[85]. A correct and prompt diagnosis may offer patients a much longer life prospect[86]. This is why specialists need to use every paraclinical tool – such as X-Rays or HRCTs[87] – to be able to accurately identify lung affections[88]–[90]. Even with the level of detail offered by an HRCT, doctors still have to rely on the keen eyes of radiologists for this purpose.

Nonetheless, there is a limit to the precision level radiologists can achieve even disregarding the inevitable variations among doctors' diagnoses [91]–[94]. To compensate for this process, modern software such as CALIPER or AI-based tools have been developed and are being used more frequently[29]–[31], [95], [96].

This thesis chapter proposes a novel technique aiming to help radiologists perform an in-depth analysis of HRCT images. The algorithm takes small samples of lung snapshots as an input, translates them into complex networks, and analyzes their texture in 3 dimensions: emphysema, ground glass opacity, and consolidation. Two sets of lung HRCTs have been processed and the resulting degree distributions show a clear difference between healthy and affected lungs. The function forms describing each type of network are a key factor in determining whether the patient suffers from an illness or even finer details such as lung deterioration due to aging. These findings confirm that such software could become part of a clinician's tools and greatly increase diagnosis precision with its fine-grained analysis.

4.2. HRCT and advanced imaging tools used for computer-aided diagnostics

HRCTs (High Resolution Computer Tomography) have been used for over 50 years now and have had a great impact on patient diagnosis and consequently, their recovery and survival rate. Whether a doctor needs to confirm a supposed diagnosis, or whether they just cannot determine the nature of a patient's illness, they ultimately call on CTs to enhance and enrich the amount of information they have, especially when it comes to diagnosing Interstitial Lung Diseases (ILD)[97], [98].

The HRCT patterns which diagnosis is typically based on, are analyzed in terms of spatial distribution within the lung as well as in its basic functional unit – the secondary pulmonary lobule (SPL). Basically, the categories of pulmonary lesions involved in creating these patterns are four main ones: reticular pattern, nodular pattern, high attenuation, and low attenuation. The combination and quantity of these lesions offer an indication of how pathologies can be interpreted and diagnosed[98].

The HRCT technique is based on taking a number of X-Ray images of the body (or a specific part of it, such as the lungs) and is carried out by producing very thin slice images which are then processed using reconstruction algorithms in order to produce a very detailed and precise representation of that body part. It can be used to either detect and evaluate illnesses, monitor patient progression while under treatment or even help decide on the area where a biopsy should be performed to determine the nature of affected tissue.

Fundamental technical HRCT protocols [99]:

- **slice thickness:** 0.625-1.25 mm
- **scan time:** 0.5-1 second
- **kV:** 120
- **mAs:** 100-200
- **collimation:** 1.5-3 mm
- **matrix size:** 768 x 768 or the largest available
- **FOV:** 35 cm
- **reconstruction algorithm:** high spatial frequency
- **window:** lung window
- **patient position:** supine (routinely) or prone (if suspected ILD)
- **level of inspiration:** full inspiration (routinely recommended) expiratory HRCT scans in patients with obstructive lung diseases

Unlike the old generation CTs, HRCTs produce thinner sections: less than 1.5 mm compared to the thicker < 3mm offered by the former, due to hardware limitations. In terms of radiation exposure, there are two types of techniques for generating an HRCT: either a sequential spaced acquisition (less exposure, but less precision and detail) or a volumetric acquisition (even thinner slices, usually ~1mm or less, combined with a post-processing algorithm which sharpens and improves image quality but exposes patients to a higher dose of radiation).

This technology is considered to be the most sensitive type of radiologic evaluation when assessing the lung parenchyma in search of ILD traces. The interstitium and SPL are the two most important components of the lung tissue which are analyzed for IPF[100].

The type of algorithm proposed here is inherently different from the classical Computer-Aided Diagnosis (CAD) approaches[24]. The majority of CAD involve ML and heuristics, yet they lack an analytical process and they are rather focused on classifying data sets rather than offering an insight into the origin of such ailments. These techniques do not offer the possibility of assessing the illness evolution or severity[29]–[31], [95], [96]. There are other programmatic approaches or software which take a more anatomically oriented angle yet to provide a valuable assessment, they require extra input information such as PFTs [101]–[103].

The HRCT composing slices also encode a type of visual indicator which cannot be guessed by the human eyes, and it reflects the color gradient or the pixel shades. Although a human observer could intuitively tell whether there is a difference between two pixels (in terms of color), they could not say what it means, or how to quantify this pragmatically. If analyzed from a density point of view, pixel shades also represent different tissue textures, which is a very important perspective when assessing an HRCT. These types of markers are stored as grayscale gradient values in the DICOM files, yet they can easily be converted into Hounsfield Units (HU) – a type of unit representing the lung radiation absorption capacity. The different types of densities are spatially intertwined within the lung tissue and create characteristic textures. The resulting patterns can be better highlighted when using complex network techniques[104], [105] and this thesis proposes and implements such an approach.

4.3. Image processing algorithms

In terms of image processing algorithms, there are quite a few tools currently available such as CALIPER or machine learning algorithms.

The proposed algorithm is composed of a set of steps that, combined, help in transforming or translating a plain HRCT image into multiple complex networks, and consequently, determining the modeling functions which best fit the characteristics of each one.

Medical practice has been relying more and more on paraclinical tools for an accurate diagnosis as well as an objective observation of patient's health state.

In diagnosing DILD for instance, there are several broadly used tools or instruments used with the purpose of painting a clear picture of the patient's clinical state. Among these, it is worth mentioning peripheral blood tests, chest X-Rays, spirometry, etc. However, for more than a decade, HRCTs (High-Resolution Computer Tomography) have also become an important part of the clinical diagnosis process, given that it presents itself with a a few advantages over some other invasive tests (e.g. biopsy) such as noninvasiveness, high visual precision, and low-level details, minimal preparation required, no anesthetic involved (with a few exceptions), the procedure lasts for no more than 10 minutes.

4.4. DICOM Image format

As with most modern medical imaging machines, HRCTs have become digital, and are created and stored in an international and standardized file format called DICOM (Digital Imaging and Communications in Medicine standard) which has been broadly accepted and adopted by most radiological equipment producers. Consequently, most, if not all, of the latest medical technologies (MRI, CT, HRCT, ultrasound) are now using this file format to store information. The National Electrical Manufacturers' Association (NEMA) created DICOM as a generic standard in order to establish and maintain compatibility between different types of medical devices used in healthcare [106].

These types of files generally have the ".dcm" file extension and they set themselves apart from other formats through the fact that they store information as data sets. The general structure of such a file consists of two sections: header and image data sets. The header holds numerous tags and values representing both the physical machine setup parameters (e.g. manufacturer, slice thickness, pixel spacing, rescale intercept, rescale slope, etc) as well as the patient's demographic data (e.g. name, age, weight, additional history, etc), all grouped into categories. Given that this type of information is sensitive and is subject to GDPR regulations, test data should be anonymized when shared for scientific purposes.

The header is embedded into the DICOM file and cannot be separated, given that it holds precious information on both the subject of the study, as well as parameters required for any DICOM viewer to be able to correctly interpret the file contents. These parameters are a type of metadata that tell the real image dimensions, matrix size, gradient, or intensity.

The rest of the file consists of the actual content, or the pixel intensities stored in a binary format, and can only be reconstructed into a visible image with the help of header data. It can be either one single image or a set of images, in 2D or 3D.

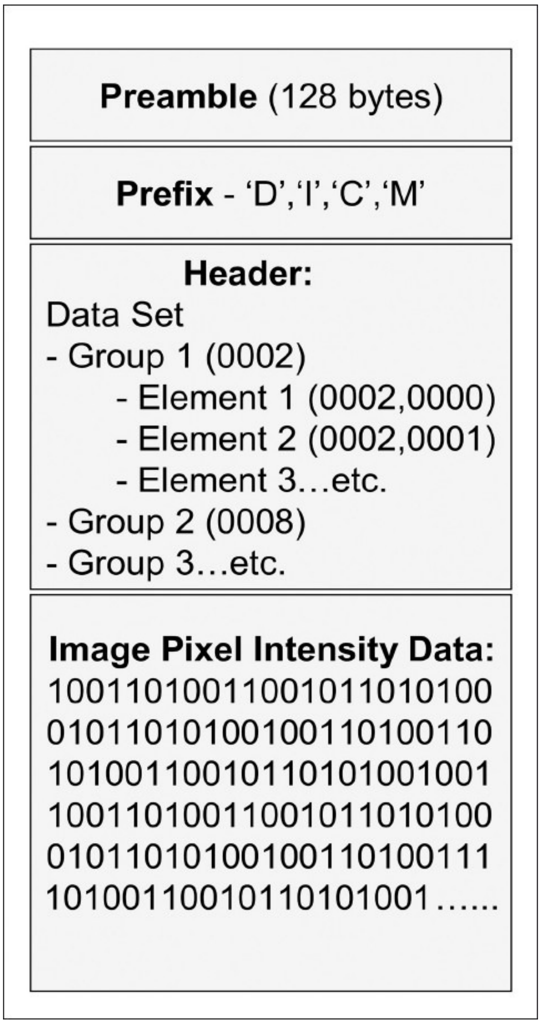


Figure 4.1 DICOM file structure [107]

In terms of visualizing the actual image represented by a DICOM file, this usually requires a type of proprietary DICOM browser to be installed, unlike well-known file types like jpeg, png, or tiff which can be easily opened with a default viewer application on a personal computer. These DICOM browsers are software applications that can be either free or may require purchasing a license. Some of the most widely used browsers in the field are: PostDICOM, Horos, RadiAnt, DICOM Viewer, Reader, MicroDicom, OsiriX DICOM Viewer [108].

4.5. Data sets

Multiple data sets were chosen for this experiment. The categories of HRCTs belong to a number of chosen pathologies, all related to Interstitial Lung Diseases: honeycombing, fibrosis, and sarcoidosis.

Depending on the specifics of each illness, the resulting complex networks greatly differ from one another and certain characteristics can be outlined for each of them.

The collection of HRCTs consists of 60 samples from the private cloud repository of 'Victor Babes' Infectious Diseases and Pneumoftiziology Clinical Hospital Timisoara as follows:

- 30 cases of patient lungs suffering from pathologies in the DILD category (all of which included CT exams and exploratory function tests)
- 30 cases of patients categorized as having normal lungs – acting as the control group

All participants had given written consent for the use of their anonymized medical data for research purposes.

Inclusion criteria for the selected cases were considered as follows:

- Every patient had been diagnosed by at least 3 pulmonologists with a medical experience of over 5 years in DILD
- Every CT has the same technical characteristics and quality as the one established across the entire lot
- All pathological patients have had imagistic evaluations for at least one year
- Every selected pathological patient also has additional investigations within their medical record, such as: DLco, FEV, clinical evaluation details, and result
- All selected CTs have been annotated by medical experts with descriptions and indications of affected lung areas

Exclusion criteria for the selected lots included:

- Patients refusing recurrent imagistic evaluation were excluded due to a lack of medical data
- Low-quality HRCT images presenting artifacts or having a slice thickness of more than 1.5 mm
- The presence of other associated serious pathologies such as neurodegenerative diseases, neuropsychiatric diseases, heart conditions, etc.
- The lack of written consent offered by patients with regard to using their medical records for research studies.

4.6. Imaging parameters

All selected HRCTs and their respective DICOM files were the result of a General Electric (GE) Healthcare Optima 520 16 slices with 32 slices reconstruction. The scanner is a 0.5 mm x 16 detector-row allowing for an 8 mm total z-axis length. The machine settings for the whole lot are as follows: slice thickness: 1.25 mm, scan time 1 second, kV: 120, mAs: 130, collimation: 2.5 mm, matrix size: 769 x 768, Field of View (FOV): 35 mmm reconstruction algorithm: high spatial frequency, window: lung window, patient position: supine or prone.

4.7. PC capabilities

All experiments carried out throughout this research paper were performed on a personal computer (PC) with the following system capabilities and specifications:

- Processor: Intel(R) Core(TM) i7-4710HQ CPU @2.50GHz 2.50GHz
- Installed memory (RAM): 12.0GB (11.9 GB usable)
- System type: 64-bit Operating System, x64-based processor
- Hard Disk capacity: 224GB; Usable: 223 GB
- Operating System: Windows 8.1 Pro

4.8. Algorithm Overview

The central idea behind this medical image processing algorithm is that it uses complex networks as a means of representing HRCTs and extracting additional information and metrics, which would otherwise require other types of machine learning algorithms specialized in image processing.

The algorithm consists of the following steps:

1. Gathering a set of healthy lung HRCTs (DICOMs)
2. Curating said DICOM files:
   a. Selecting one or more samples out of each DICOM (one sample per person)
   b. Out of each sample, cropping out a patch of 65 x 65 pixels (healthy lung sample)
3. For each sample convert the grayscale image into a Hounsfield Unit matrix:
   a. Treat every square sample as a pixel matrix
   b. Convert grayscale pixel values into Hounsfield Units
   c. Copy all Hounsfield Units corresponding to studied illnesses into new matrices (emphysema, ground glass opacity, consolidation) - practically constructing new images only with lung tissue corresponding to the studied aspects
4. Each new matrix is converted into an adjacency matrix (complex network) according to the following rules:
   a. Every pixel is counted as a vertex
   b. Links (edges) between two nodes exist only if:
      i. The radial distance between them is $rd \leq 4$
      ii. The gradient difference (Hounsfield unit delta) is $\Delta \leq 50$
      iii. Hounsfield units values for both nodes fall into the same HU band
5. Each of the 4 matrices is analyzed from a mathematical perspective, defining approximating functions for all of them. These functions will constitute a baseline for all other studies datasets.
6. Repeat the whole process with HRCTs (DICOMs) of illness-affected lungs
7. Compare the resulting functions of healthy lungs to functions corresponding to damaged lungs to determine similarities and differences.

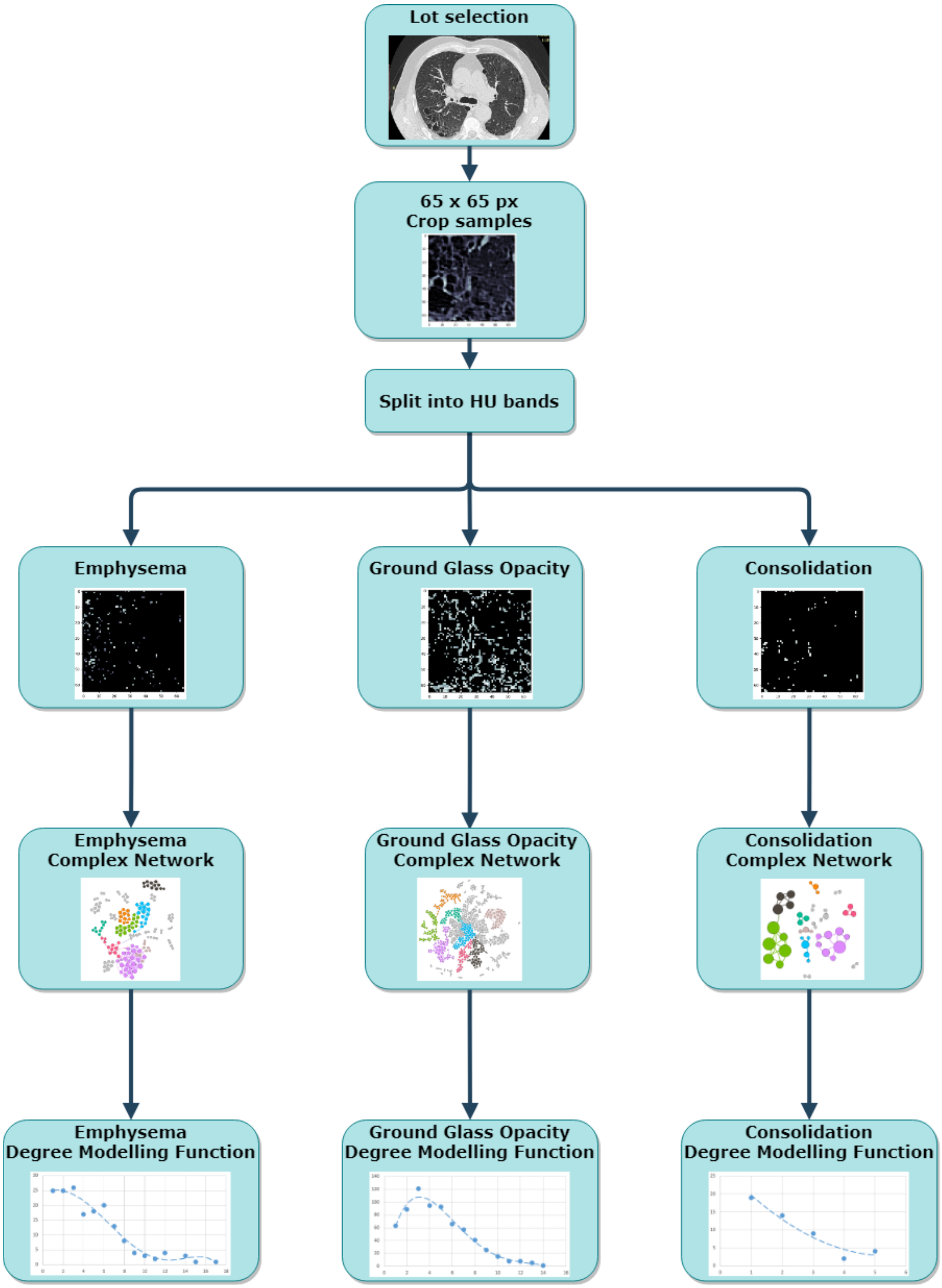


Figure 4.2 High-level overview of DICOM to Complex Networks processing algorithm

4.9. Technical HRCT selection

Two sets of DICOM files were put together with the help of a team of medical specialists which provided an entire collection of HRCTs split into categories: healthy lungs, fibrosi- affected lungs, lungs affected by sarcoidosis, lungs affected by COVID-19, etc.

For each of the categories, a set of characteristics was provided along with concrete samples illustrating a snapshot of the lung which would be most relevant to the case in the study (Figure 4.3). In addition, most patients had not one, but two, or three CTs taken at different points in time, and these were used to analyze and assess the evolution of the pathologies dynamically. These curated samples were then ingested into the proposed algorithm and transformed as per the aforementioned steps.

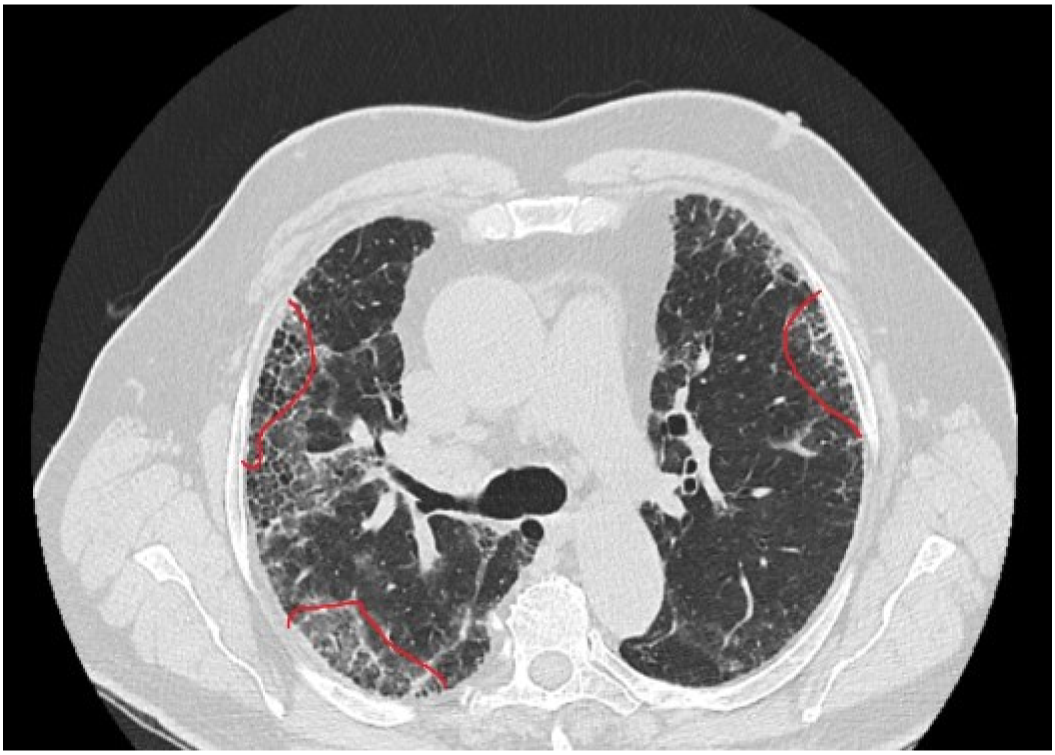


Figure 4.3 Sample DICOM image provided by medical specialists, indicating affected areas to be analyzed

4.10. Curating DICOM files

Each DICOM file was cropped to a 65x65 pixel area containing only a certain portion of the lung which would best display the medical diagnostic. Medical indications featured in Figure X above show the best areas to be processed for a certain ailment.

4.11. Crop area definition

So far, there have been few attempts at slicing and screening HRCTs in such a way. There are multiple reasons behind this starting with the fact that not many people have followed this type of approach, and the ones that did, used different techniques or algorithms, some in the machine learning area. Some studies show that even an 11x11 px area could be enough to identify affected tissue and different patterns in the lung. On the other hand, one might argue that such a patch would be too small to be able to give any guarantees or high probabilities as to the accuracy of the output. This is why, in the pursuit of building a more encompassing algorithm, a larger area was chosen for this study, i.e. 65x65 px. The exact dimensions of this square patch were selected due to a few factors, also confirmed by clinicians. Among these: the area should be large enough to capture an entire secondary lobule - which is the fundamental unit of the lung and its contour is indicative of how disease manifests itself or how much it has expanded. In reality, the dimension of such a unit is normally half the area of the crop, but choosing a larger observation window ensures there is at least one secondary lobule in it. Some standard measurements used in this case: a regular secondary lobule has a square surface between [1 cm; 2.5 cm] in diameter [109]. Converting this value into pixels requires knowing the pixel spacing (PS) and this is a value encoded as a metadata parameter into every HRCT and it typically varies between [0.70, 0.80]. Within the current research experiment, all DICOM files had PS = 0.74 mm. Considering that the maximum secondary lobule area is 2.5 cm^2 x 2.5 cm^2 then a simple calculus shows that the minimum sample size should be 25 / 0.74 = 33.7837 px. However, for a better probability of framing at least one entire secondary lobule within the crop sample, the analyzed area dimensions are almost doubled (65 px).

A larger crop could and probably would offer more information, however, this is where processing power capabilities impose limits. The larger the studied area, the more it takes for the application to process it. Similar research approaches consider areas of 11 x 11 px in trying to assess lung tissue, however, the decision for choosing such dimensions is not clearly stated [102], [103].

Despite this, one of the future goals is to extend said area along with the usage of processing units which would increase the performance.

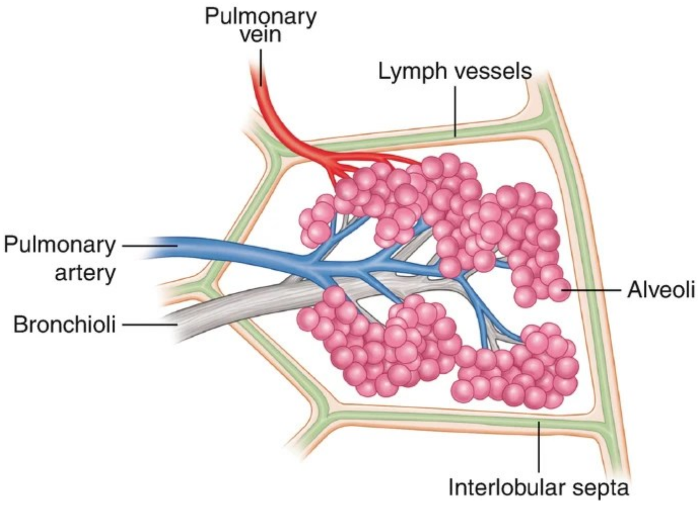


Figure 4.4 Secondary pulmonary lobule structure [110]

Figure 4.4 shows a simplified depiction of the basic lung unit – the secondary pulmonary lobule. Its shape and respective size can be better grasped when looking at Figure 4.5 along the outer edge of the lungs.

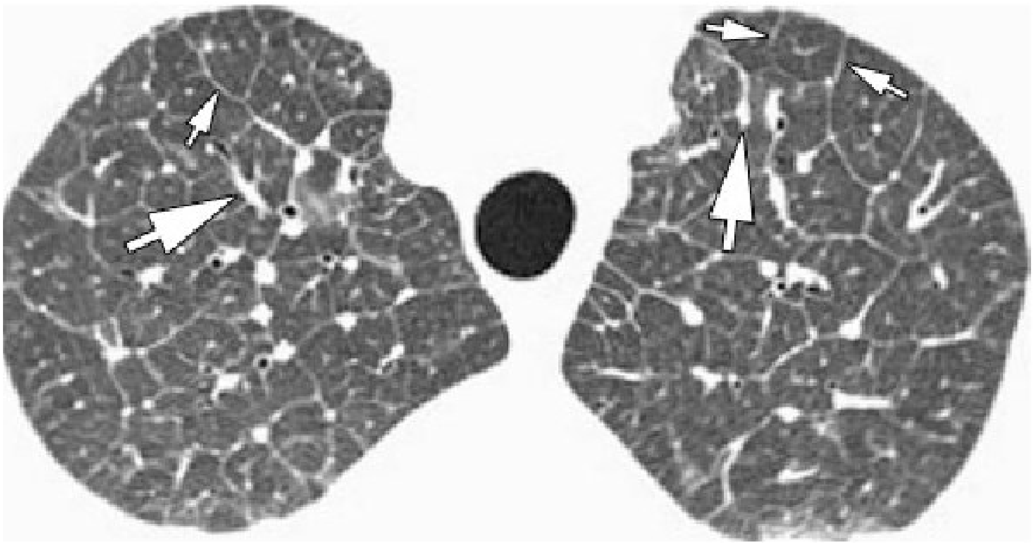


Figure 4.5 Secondary Pulmonary Lobule as seen on an HRCT [109]

4.12. Radial distance selection

Radial distance (rd) has been defined as the maximum distance (linear, euclidean distance) up to which a certain network vertex Vx can be considered potentially connected to another node Vy. In other words:

$$\exists(\mathrm{Vx}, \mathrm{Vy}) \rightarrow \mathrm{d}(\mathrm{Vx}, \mathrm{Vy}) \leq \ rd \qquad (4.1)$$

where *rd* is a nonarbitrary value chosen according to multiple tests and experiments. In this case, an array of discrete values of $rd \in \{1;8\}$ has been tested in order to find the most proper fit, meaning that the chosen value would have to lead to the generation of a medically relevant complex network, with the characteristics of a biological process. A value of *rd* = 1 can be interpreted as: any vertex Vx can only be linked to another Vy if the euclidean distance between them is less than or equal to 1, in other words, they can be linked only if they are adjacent, while an *rd* = 8 allows for potential neighbors at a distance of 8 pixels away.

The difference between choosing the most useful *rd* value is made by the resulting complex networks. An *rd = 1* could make for a very sparse network (thin, scattered, disconnected clusters, even single pixels) which would not be very relevant medically and could easily be confused with noise - which often happens due to patients moving while being scanned. Conversely, an *rd* = 8 would allow for too much inclusion, resulting in a too-dense network, which in medical terms suggests that there is no texture difference between all vertices, since they are all connected, and as a result, they are treated as being very similar. Another consequence of having a too-dense network is that there is an agglomeration of closely interconnected pixels, which would not reveal anything new to the researcher, because a high density of similar pixels would also be visible to the naked eye.

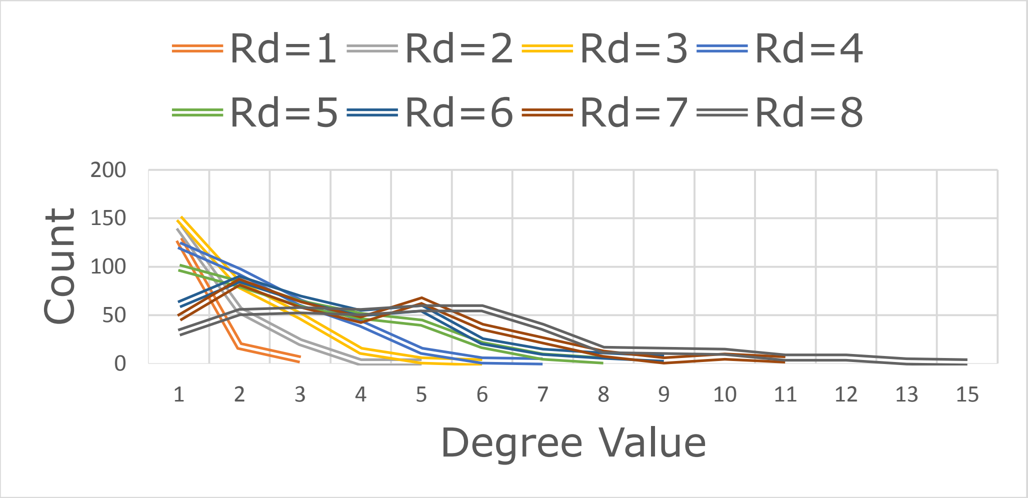


Figure 4.6 Degree distributions for various radial distances [111]

Based on the tested *rd* values, the most suitable value for such a radial distance has been chosen as *rd=4*, meaning that the sensibility of this approach would detect lesions with a size of 4*0.74 = 2.96 mm. Indeed, this measurement also corresponds to the minimum detectable lesion size of 3-17 mm suggested by other previous research [112].

4.13. Hounsfield Units - selection of Hounsfield Bands

Hounsfield Units (HU) are the basic units used to quantify and evaluate the scale of X-ray absorption and attenuation into human tissue. Medical specialists make use of this type of analysis when interpreting HRCTs in order to confirm certain diagnostic suspicions which would be too difficult to evaluate visually[113]. The spectrum of HU values varies between -1000 HU in the case of air to 3000 HU in the case of hard metals such as steel. To the human eye, these values can be associated with different shades from white to black according to the amount of radiation the tissue can absorb. While air has a very dark color (dark gray or even black), lung tissue, organs, and vessels tend to be on the other side of the spectrum (white, light gray, up to a darker gray).

Pulmonary tissue	**HU intervals**
Emphysema	[-1024, -977)
Normal pulmonary parenchyma	[-977, -703)
Ground-glass opacities	[-703, -368)
Others (crazy-paving, pleural fat)	[-368, -100)
Consolidation	[-100, 5)
Others (interstitial vessels)	>5 HU

Table 4.1 HU intervals (bands) spectrum per type of pulmonary lesion / density[27], [114], [115]

An affected lung might have either one or a combination of multiple such tissue densities, according to the pathology it may suffer from. For the purpose of this study, only three types of HU bands were analyzed in the context of multiple illnesses: Emphysema, Ground-glass opacities and Consolidation.

4.14. Vertex similarity based on maximum gradient delta

As mentioned, for two vertices to be considered similar, the maximum gradient delta must be less than or equal to a certain delta = 50. This value was chosen based on two criteria:

1. Given the range of values each of the HU bands can take, they should be split into multiple sub-bands in order to differentiate between pixels pertaining to the same HU band. This gives an insight into subtle differences, as well as allows the inclusion of connections where pixels are tightly coupled. While the Emphysema layer is a narrow one and given the max possible delta = 47 meaning that all pixels with shades in these bands could be connected to each other, the other two layers (GGO and Consolidation) can both be split into multiple 50-unit range sub-bands.
2. Admitting a similarity difference greater than *Δ=50* (or none at all) would mean that, for instance, a pixel with a HU value HvX=-702 and another one HvY=-369 represent the same kind of biological tissue, which would be a great overstatement, given that there is such a large range of discrete values that needs to be covered between the two. Another case that is eliminated with this similarity delta is the one where the HRCT contains a lot of noise, meaning that some pixels might be brighter due to the patient moving while being scanned. What happens in these situations is that the whole image will often look brighter, introducing a lot of so-called "noise". Eliminating the outliers, in this case, the much too bright vertices is obtained if we narrow down the search and look for a more specific small range of values when generating edges between same-band pixels.

4.15. Converting DICOM grayscale values into Hounsfield values

As previously mentioned, the dcm image is stored as an array of grayscale values. In order to convert such data into Hounsfield Units, which would give an insight into the tissue capacity of X-Ray absorption, and implicitly, understanding the nature of lung cells in a certain area, these values need to be converted using a simple formula:

$$\text{pixel\_hu\_value} = \text{pixel\_value} * \text{RescaleSlope} + \text{RescaleIntercept} \quad (4.2)$$

where pixel_hu_value is the HU converted value, pixel_value is the original grayscale pixel value stored by the dcm file, RescaleSlope and RescaleIntercept are constants embedded in the dcm header dependent on the HRCT machine settings.

For the current study, all HRCTs have the following default rescale parameters presented in Table 4.2.

Tag ID	Description	Value
(0028,1052)	Rescale Intercept	-1024
(0028,1053)	Rescale Slope	1
(0028,1054)	Rescale Type	HU

Table 4.2 Rescale parameters – DICOM Metadata Parameters [116]

In table 4.2 Rescale Type defines the output unit of pixel_hu_value in (4.2) above. It can have one out of multiple values such as: OD (Optical Density), HU (Hounsfield Units), US (Unspecified), MGML (mg/ml), Z_EFF (Effective Atomic Number), ED (Electron Density), EDW (Electron Density Normalized), HU_MOD (Modified Hounsfield Unit), PCT (Percentage %).

The Tag ID for each of the parameters is the official documented ID and it is part of the standard for all DICOM files.

4.16. Converting HU bands into complex networks

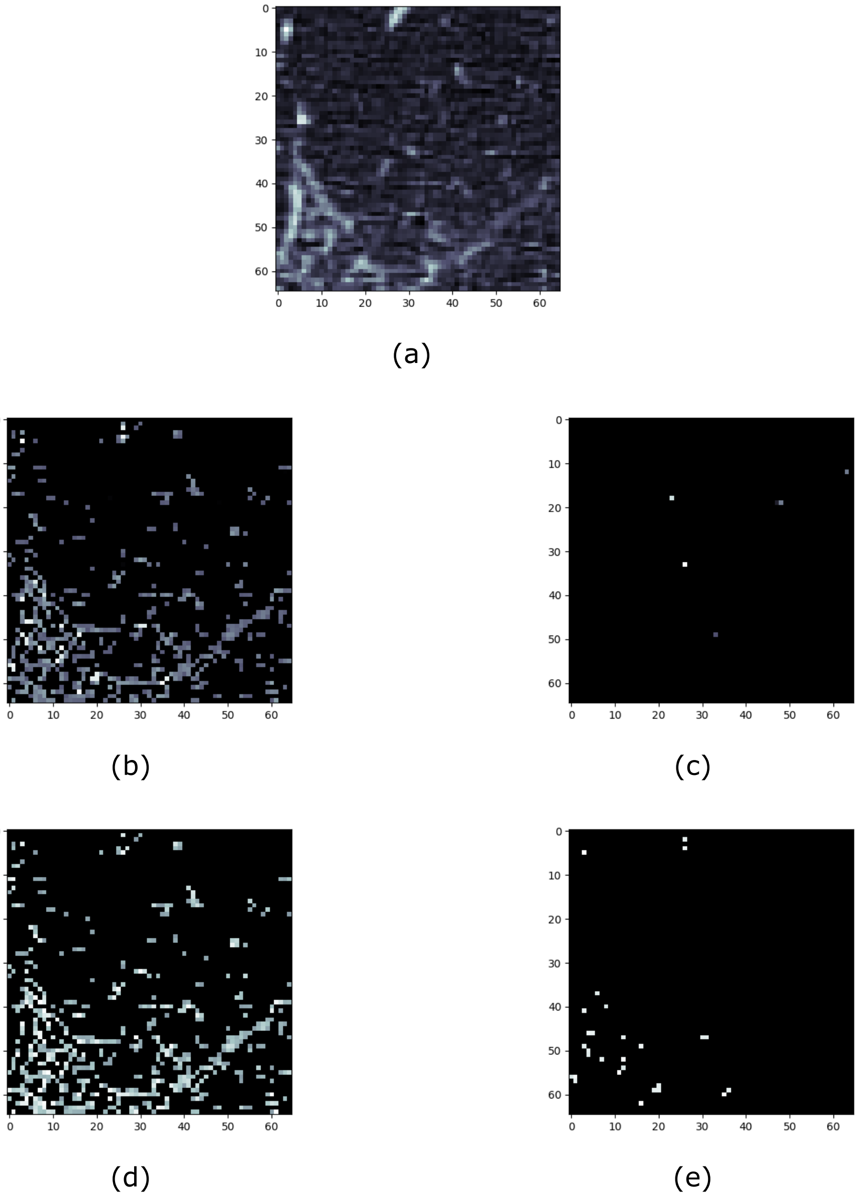
For this study, only three HU bands were selected as being relevant, and as a consequence, the proposed algorithm generates one adjacency matrix for each one, as well as one containing all of them.

The adjacency matrix is constructed based on the previously defined rules:

1. Two vertices are linked if the maximum radial distance (rd) between them is rd = 4
2. Two nodes are adjacent if the maximum gradient difference between them is Δ=50

Figure 4.7 shows an original piece of a DICOM image, with a dimension of 65x65 pixels. While Fig 4.7 (a) shows the original 65 x 65 px sample image, Figures 4.7 (b), (c), (d) and (e) show the transformation it suffers throughout the layering process: first the noise is eliminated (any other HU values not pertaining to the three selected bands are left out), then the resulting image is further split into 3 separate images, each one containing only pixels whose HU value pertain to one HU band: Emphysema, GGO or Consolidation.

A more in-depth example is described in Figure 4.8 from a programmatic point of view.



(a)

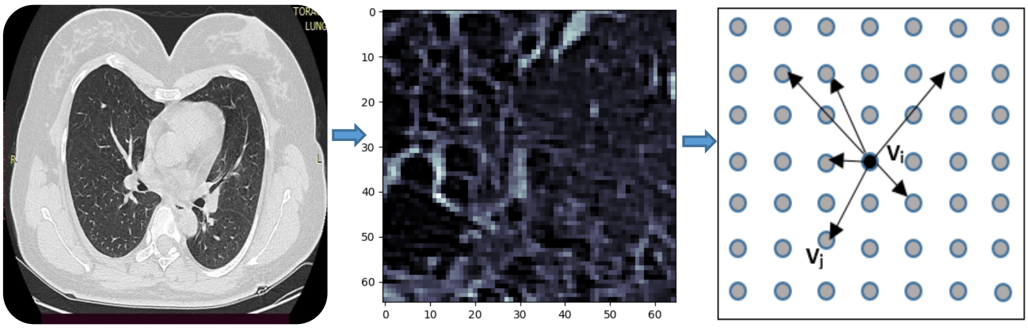
(b)

(c)

(d)

(e)

Figure 4.7 Original HRCT deconstructed into multiple HU layers: a) Sample crop b) Emphysema layer, GGO layer, Consolidation layer only, without “noise” c) Emphysema layer d) GGO layer e) Consolidation layer



(a) (b) (c)

367	241	230	149	75	130	83
269	180	195	190	189	228	123
382	479	533	500	589	387	267
419	517	502	549	460	537	491
258	326	421	575	257	522	146
323	403	334	533	487	262	234
373	371	292	233	243	270	219

(d)

		230	149	75		
	180	195	190	189	228	
382	479	533	500	589	387	267
419	517	502	549	460	537	491
258	326	421	575	257	522	146
	403	334	533	487	262	
		292	233	243		

(e)

		319	400	474		
	369	354	359	360	321	
167	70	16	49	40	162	282
130	32	47	■	89	12	58
291	223	128	26	292	27	403
	146	215	16	62	287	
		257	316	306		

(f)

		0	0	0		
	0	0	0	0	0	
0	0	1	1	1	0	0
0	1	1	■	0	1	0
0	0	0	1	0	1	0
	0	0	1	0	1	
		0	0	0		

(g)

Figure 4.8 Image to matrix conversion for an rd≤4 and Δ≤50

a) Original DICOM image b) Sample crop c) Sample equivalent matrix V_{MxN} d) HU values e) Limit neighborhood to rd ≤4 f) Calculate Δ between HU_{vi} and HU rest of the pixels g) Mark adjacency matrix with 1 where the Δ≤50 condition is met and 0 otherwise

Figure 4.8 Shows how lung samples are converted from grayscale to HU values and then based on the conversion rules, are turned into an adjacency matrix.

For each of the composing pixels, the algorithm evaluates all neighbors within the rd≤4 and Δ≤50. If these conditions are met, then the edge is marked in the adjacency matrix.

First (Fig. 4.8 d) the selected image crop pixels are all converted into their HU equivalents according to equation 4.2.

Each of the composing pixels is then considered the center of an area and all its neighbors are evaluated in terms of similarity and radial distance.

Subsequently, any other pixels beyond the *rd=4* pixel area of the current pixel is eliminated for not complying with the distance criterion. (Fig. 4.8. e).

Secondly, for the similarity criterion to be fulfilled, all potentially linked pixels should have a HU value within 50 units difference from the main pixel. In other words, the delta calculated between the two HU values should be Δ≤50. If this condition is met, then the edge is marked with 1 in the adjacency matrix (equivalent to the image crop), otherwise, no edge is stored. Edges are undirected and unweighted.

When all composing pixels have been evaluated, the adjacency matrix is complete and it can be used as raw material for the complex network representation of the selected sample. Based on this adjacency table, the degree distribution can be computed and different network characteristics can be highlighted.

4.17. Algorithm implementation

The described algorithm has been implemented in Python, a flexible programming language that offers an array of libraries with different functions. One of the libraries in particular has the ability to read and interpret HRCTs and medical images stored in a Dicom format: Pydicom.

4.17.1 Main function

```
def process_cts(ctNumber, fibrosisLungPath, threshold, n, radialDistance, patchOrigin):

       start_time = time.time()

          #ctNumber = 1
          #fibrosisLungPath = r"PATH/TO/DICOM/FILE.DCM";
          #threshold = 50 #gradient threshold
          #n = 65 # sample area width/length

          # read hrct file
       ds = dcmread(fibrosisLungPath)

       ct = ds.pixel_array
       networkParams = NetworkParams(n, threshold, radialDistance)

       adjacencyMatrix = []
       xOrigin = patchOrigin.x
       yOrigin = patchOrigin.y

          # sampled area coordinates
       sampleCT = ct[xOrigin: xOrigin + n, yOrigin: yOrigin + n]
          # calculate ggo/emphisema/consolidation matrix
       huMatrices = convertToHu(n, ds, sampleCT)

       convertedImageGgo = huMatrices.huMatrixAll
       convertedImageGgo_e = huMatrices.huMatrixE
       convertedImageGgo_g = huMatrices.huMatrixG
       convertedImageGgo_c = huMatrices.huMatrixC


      plt.imshow(convertedImageGgo, cmap=plt.cm.bone)
      plt.show()
      plt.clf()
```

```
       # Defining output file path for layer images
       imagePath = "C:\\PATH\\TO\\OUTPUT\\" + str(ctNumber) + "CT" +
str(ctNumber)

       # Generate curated images for all HU layers
       generate_separated_images(imagePath, huMatrices)

        adjFilePath   =   "C:\\PATH\\TO\\OUTPUT\\"   +   str(ctNumber)   +
"\\edgeList_nsip_CT_" + str(ctNumber)

       generate_all_adj_matrices_at_once(adjFilePath,networkParams,
huMatrices)
         end_time = time.time()
         print("--- %s seconds ---" % (end_time - start_time))
```

Code Snippet 4.1. The main function of the algorithm

Given the complexity of the algorithm and having in mind the idea of scalability, the performance of the algorithm needs to be evaluated. The high number of dicom files that need to be processed and converted into complex networks requires a fast algorithm and quick results. For this reason, all reruns are tracked and timed.

All the HRCTs are numbered and split into multiple folders/categories depending on the illnesses. (`ctNumber`).

The path to all Dicom files is passed as an argument and varies according to the category. (`fibrosisLungPath`)

`Threshold` represents the gradient threshold beyond which two vertices are not considered linked anymore.

`n` represents the dimension of the lung patch to be processed. In this case, has been chosen as `n = 65` pixels.

`patchOrigin` is the origin of the selected 65 x 65 px square framing of the analyzed lung area, indicated by medical specialists and radiologists.

`convertToHu`, `convertToHuEmphysema`, `convertToHuGgo`, `convertToHuConsolid` are all functions converting the dicom image into its equivalent HU values matrices.

Generates an image containing only the pixels pertaining to the different separated HU bands.

4.17.2 Converting pixel values to HU values

```
def convertToHu(n, ds, sampleCT):
         convertedImage_egc = []
         convertedImageGgo_e = []
         convertedImageGgo_g = []
         convertedImageGgo_c = []
         rescaleSlope = ds.RescaleSlope
         rescaleIntercept = ds.RescaleIntercept

         for i in range(0, n):
             currentLine = []
             currentLine_e = []
             currentLine_g = []
             currentLine_c = []
             for j in range(0, n):
                 gradient = sampleCT[i, j]
                 nodeHu = rescaleSlope * gradient + rescaleIntercept

                 if (-1024 < nodeHu < -984): #emphysema
                     currentLine.append(gradient)
                     currentLine_e.append(gradient)
                 else:
                     if (-634 <= nodeHu <= -368):    #GGO
                         currentLine.append(gradient)
                         currentLine_g.append(gradient)
                     else:
                         if (-109 <= nodeHu <= 9):   #Consolidation
                             currentLine.append(gradient)
                             currentLine_c.append(gradient)
                         else:
                             currentLine.append(0)
                             currentLine_e.append(0)
                             currentLine_g.append(0)
                             currentLine_c.append(0)

             convertedImageGgo.append(np.array(currentLine))
             convertedImageGgo_e.append(np.array(currentLine))
             convertedImageGgo_g.append(np.array(currentLine))
             convertedImageGgo_c.append(np.array(currentLine))

         huMatrices          =          HuMatrices(convertedImage_egc,
convertedImageGgo_e, convertedImageGgo_g, convertedImageGgo_c)
         return huMatrices
```

Code Snippet 4.2 - Converting DICOM image into HU layers

In code Snipped 4.2, the function does exactly what its name suggests: it receives as an argument the dicom image, the dimension of the analyzed patch (in this case n = 65), and the two HRCT parameters for converting pixels into HU values. All pixels are iterated through, converted, and then depending on their value, either stored in one of the new HU layer matrices or discarded. Finally, all resulting matrices are stored in a HuMatrices object and returned.

```
def generate_separated_images(imagePath, huMatrices):
         png = ".png"
         plt.imshow(huMatrices.huMatrixAll, cmap=plt.cm.bone)
         plt.savefig(imagePath + "_snapshot_e_g_c" + png)

         plt.imshow(huMatrices.huMatrixE, cmap=plt.cm.bone)
         plt.savefig(imagePath + "_snapshot_e" + png)

         plt.imshow(huMatrices.huMatrixG, cmap=plt.cm.bone)
         plt.savefig(imagePath + "_snapshot_g" + png)

         plt.imshow(huMatrices.huMatrixC, cmap=plt.cm.bone)
         plt.savefig(imagePath + "_snapshot_c" + png)
```

Code Snippet 4.3 - Generating sample images out of each HU layer matrix

Generating HU layer images is quite simple once the matrices have been generated. Cmap specifies the color map to be used when printing the pixel values.

4.17.3 Generating adjacency matrices

```
def    generate_all_adj_matrices_at_once(adjFilePath,    networkParams,
huMatrices):

         # ALL
          adjFileName1 = adjFilePath + "_e_ggo_c.csv"
          adjFileName2 = adjFilePath + "_e.csv"
          adjFileName3 = adjFilePath + "_ggo.csv"
          adjFileName4 = adjFilePath + "_c.csv"
         fEGC = open(adjFileName1 , "w")
         fE = open(adjFileName2 , "w")
         fG = open(adjFileName3 , "w")
         fC = open(adjFileName4 , "w")
         adjacencyMatrixEGC = []
         adjacencyMatrixE = []
         adjacencyMatrixG = []
         adjacencyMatrixC = []
```

```
      adjList1 = []
      adjList2 = []
      adjList3 = []
      adjList4 = []

      n = networkParams.n
      radialDistance = networkParams.radialDistance
      threshold = networkParams.threshold

      for i in range(0, n * n - 1):
         currentLine1 = []
         currentLine2 = []
         currentLine3 = []
         currentLine4 = []
         for j in range(0, n * n - 1):
             xI = i // n
             yI = i % n
             xJ = j // n
             yJ = j % n
             a = (xI, yI)
             b = (xJ, yJ)
             distIJ = distance.euclidean(a, b)

       delta1 = abs(huMatrices.huMatrixAll[xI][yI] -
huMatrices.huMatrixAll[xJ][yJ])
       delta2 = abs(huMatrices.huMatrixE[xI][yI] -
huMatrices.huMatrixE[xJ][yJ])
       delta3 = abs(huMatrices.huMatrixG[xI][yI] -
huMatrices.huMatrixG[xJ][yJ])
       delta4 = abs(huMatrices.huMatrixC[xI][yI] -
huMatrices.huMatrixC[xJ][yJ])

             # e_ggo_c
             if radialDistance >= distIJ > 0 and delta1 <
threshold and huMatrices.huMatrixAll[xI][yI] != 0 and
huMatrices.huMatrixAll[xJ][yJ] != 0:
                 currentLine1.append(1)
                 edge = str(i) + ' ' + str(j)
                 altEdge1 = str(j) + ' ' + str(i)
                 if edge not in adjList1:
                     adjList1.append(edge)
                     adjList1.append(altEdge1)
                     fEGC.write(str(i) + ' ' + str(j))
                     fEGC.write('\n')
             else:
                 currentLine1.append(0)
```

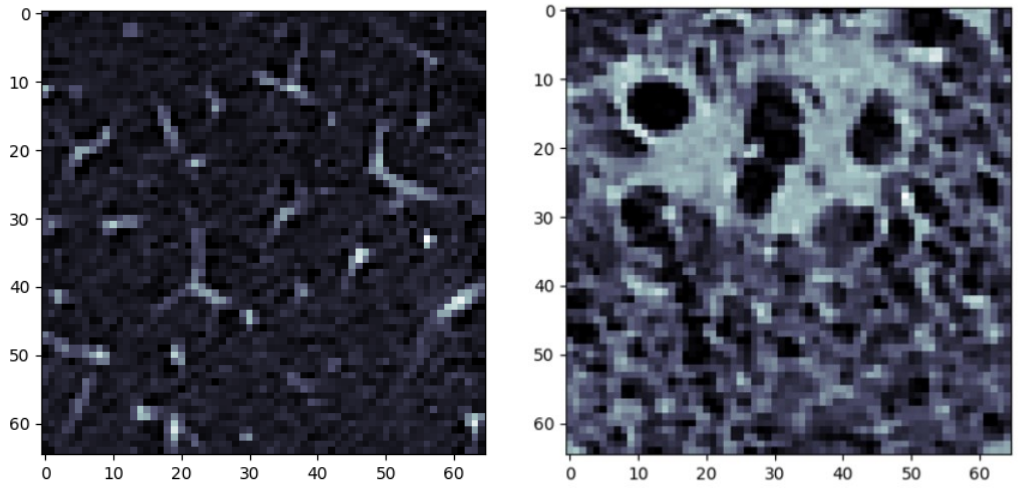
```
            # emphysema
            if radialDistance >= distIJ > 0 and delta2 <
threshold and huMatrices.huMatrixE[xI][yI] != 0 and
huMatrices.huMatrixE[xJ][yJ] != 0:
                currentLine2.append(1)
                edge = str(i) + ' ' + str(j)
                altEdge2 = str(j) + ' ' + str(i)
                if edge not in adjList2:
                    adjList2.append(edge)
                    adjList2.append(altEdge2)
                    fE.write(str(i) + ' ' + str(j))
                    fE.write('\n')
            else:
                currentLine2.append(0)

            # ggo
            if radialDistance >= distIJ > 0 and delta3 <
threshold and huMatrices.huMatrixG[xI][yI] != 0 and
huMatrices.huMatrixG[xJ][yJ] != 0:
                currentLine3.append(1)
                edge = str(i) + ' ' + str(j)
                altEdge3 = str(j) + ' ' + str(i)
                if edge not in adjList3:
                    adjList3.append(edge)
                    adjList3.append(altEdge3)
                    fG.write(str(i) + ' ' + str(j))
                    fG.write('\n')
            else:
                currentLine3.append(0)

            # consolidation
            if radialDistance >= distIJ > 0 and delta4 <
threshold and huMatrices.huMatrixC[xI][yI] != 0 and
huMatrices.huMatrixC[xJ][yJ] != 0:
                currentLine4.append(1)
                edge = str(i) + ' ' + str(j)
                altEdge4 = str(j) + ' ' + str(i)
                if edge not in adjList4:
                    adjList4.append(edge)
                    adjList4.append(altEdge4)
                    fC.write(str(i) + ' ' + str(j))
                    fC.write('\n')
            else:
                currentLine4.append(0)

        adjacencyMatrix1.append(np.array(currentLine1))
        adjacencyMatrix2.append(np.array(currentLine2))
```

```
       adjacencyMatrix3.append(np.array(currentLine3))
       adjacencyMatrix4.append(np.array(currentLine4))

    fEGC.close()
    fE.close()
    fG.close()
    fC.close()

    adjMatrices = NetworkAdjacencyMatrices()
    adjMatrices.adjMatrixAll = np.array(adjacencyMatrix1)
    adjMatrices.adjMatrixE = np.array(adjacencyMatrix2)
    adjMatrices.adjMatrixG = np.array(adjacencyMatrix3)
    adjMatrices.adjMatrixC = np.array(adjacencyMatrix4)

    return adjMatrices
```

Code Snippet 4.4 - Converting HU layers into Network Adjacency Matrices

4.18. Results

All DICOM sets were analyzed through the above algorithm. The following samples illustrate how this process is carried out for two different patients: a clinically healthy one as well as one with a DILD (Diffuse Interstitial Lung Disease) diagnosis.

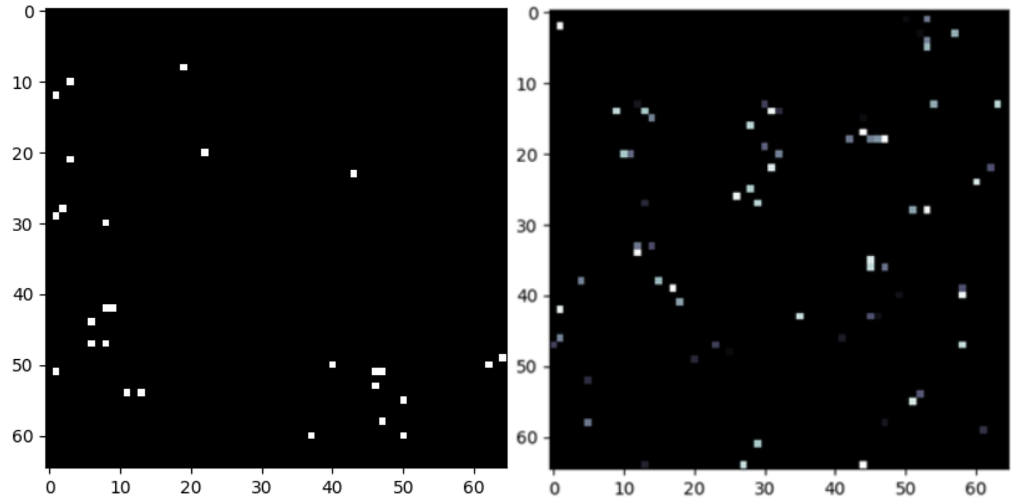


(a) (b)

Figure 4.9 Sample selection for two different patients (a) Normal (control) sample of a healthy lung; (b) DILD (IPF) diagnosed lung sample

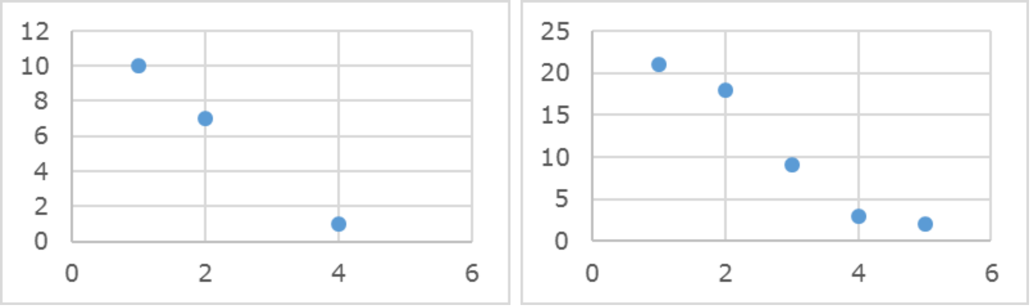
The following steps of the algorithm produce the 3 layers corresponding to the HU bands selected as the main interest for this research experiment. Pixels pertaining to different categories are separated and aggregated as individual images, then converted into complex networks adjacency matrices according to the established conditions.

The Emphysema layer is the first analyzed HU band, resulting in two different images for each of the two samples. Although they may not seem to be a visible difference when looking at them with the naked eye, Fig 4.10 (b) is still showing more pixels than Fig 4.10 (a), meaning that the affected lung is displaying signs of abnormality. This can also be observed in Fig 4.11 (b) where the degree distribution and the number of Emphysema pixels for the ill lungs are quite higher and denser.



(a) (b)

Figure 4.10 Emphysema layers for (a) normal lungs sample (b) DILD lung sample



(a) (b)

Figure 4.11 Emphysema layer degree distribution for (a) normal lung sample (b) DILD lung sample

The degree distributions (Fig. 4.11) of both samples can tell apart which of the samples is the normal lung and which has a pathology.

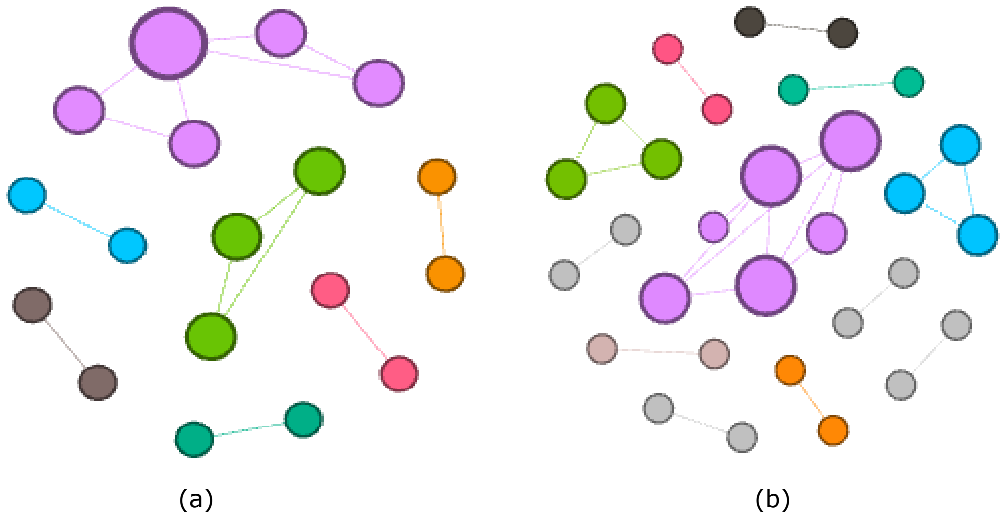


Figure 4.12 Emphysema layer converted into complex networks for (a) normal lung sample (b) DILD lung sample

The resulting complex networks for the Emphysema layer (Fig 4.12) are also visibly different and this confirms that this type of tissue is more likely to be found in the affected lung rather than the healthy one.
The second layer of interest is the GGO, again, with the three steps involved: generating the snapshots, creating the adjacency matrices, and displaying the resulting equivalent complex networks.

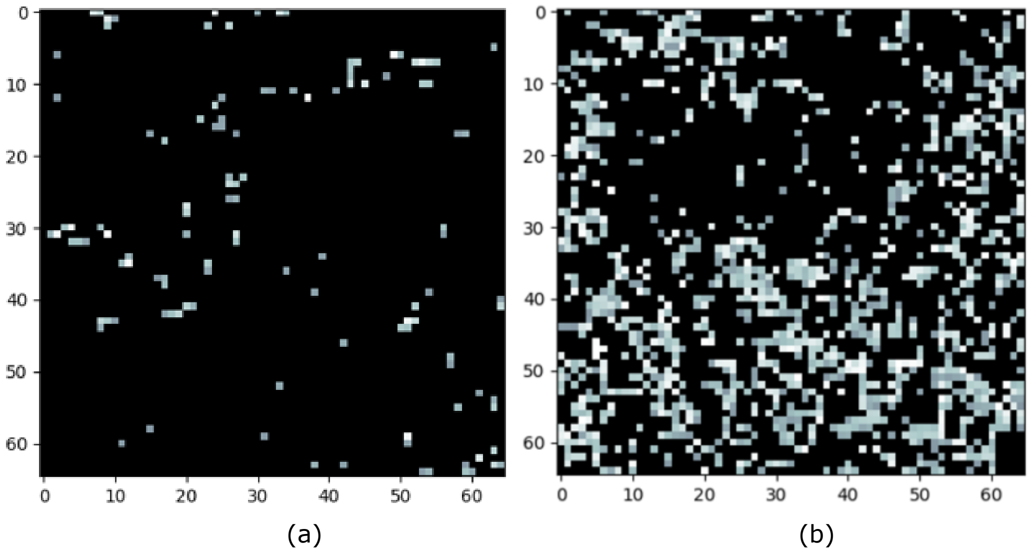
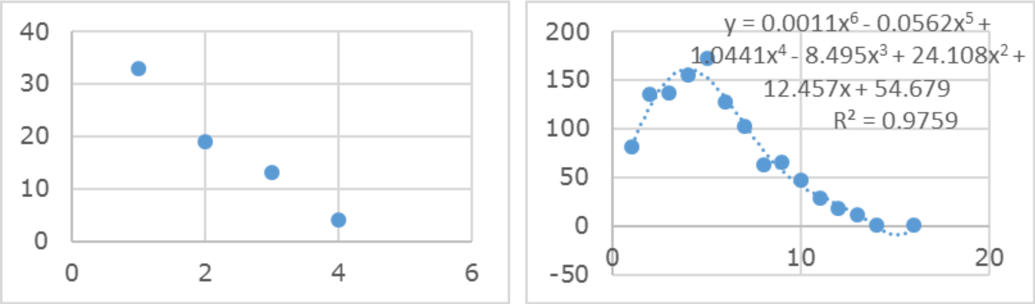


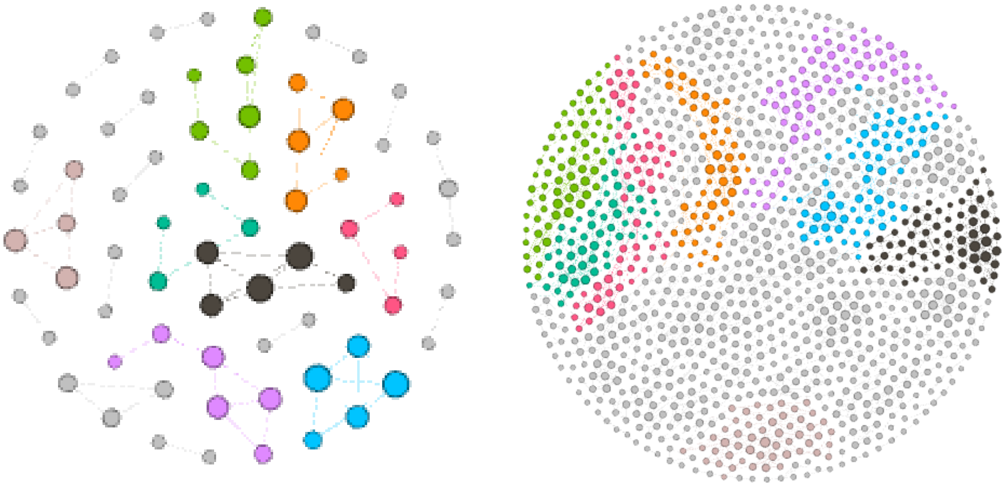
Figure 4.13 GGO layers for (a) normal lungs sample (b) DILD lung sample

There is quite a visible difference between the two layers in terms of GGO (Fig. 4.13), and indeed, medical specialists confirm that this type of tissue density forms in the case of DILD diseases. In contrast, healthy lungs have much less of it, but depending on the patient's particularities (age, associated diseases, lifestyle) they may also display small areas of such tissue.



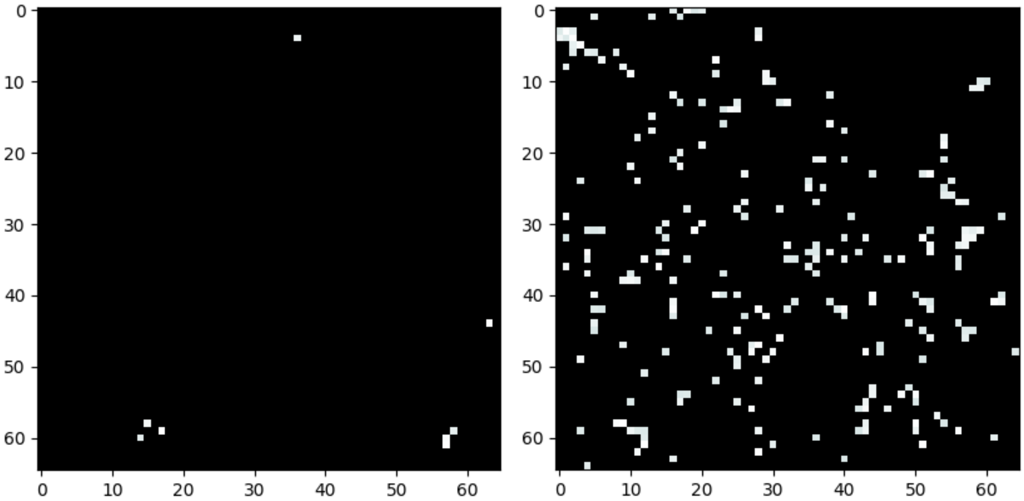
(a) (b)

Figure 4.14 GGO layer degree distribution for (a) normal lung sample (b) DILD lung sample



(a) (b)

Figure 4.15 GGO layer converted into complex networks for (a) normal lung sample (b) DILD lung sample



(b) (b)

Figure 4.16 Consolidation layers for (a) normal lungs sample (b) DILD lung sample

Similar to the GGO layer, the Consolidation samples are showing a clear disproportion regarding the density of affected tissue which can be found in a DILD lung and this will also be reflected in the degree distributions of the equivalent complex networks.

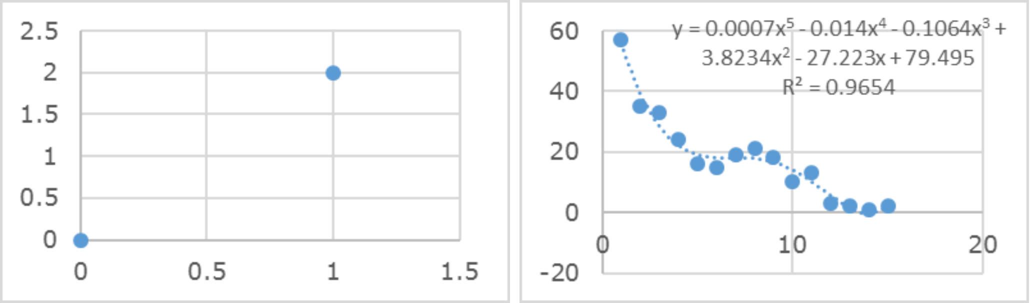


Figure 4.17 Consolidation layer degree distribution for (a) normal lung sample (b) DILD lung sample

While in Fig. 4.16 (a) Consolidation pixels barely exist, Fig 4.16 (b) shows a whole different story, which is coherent and consistent with what one might assume.

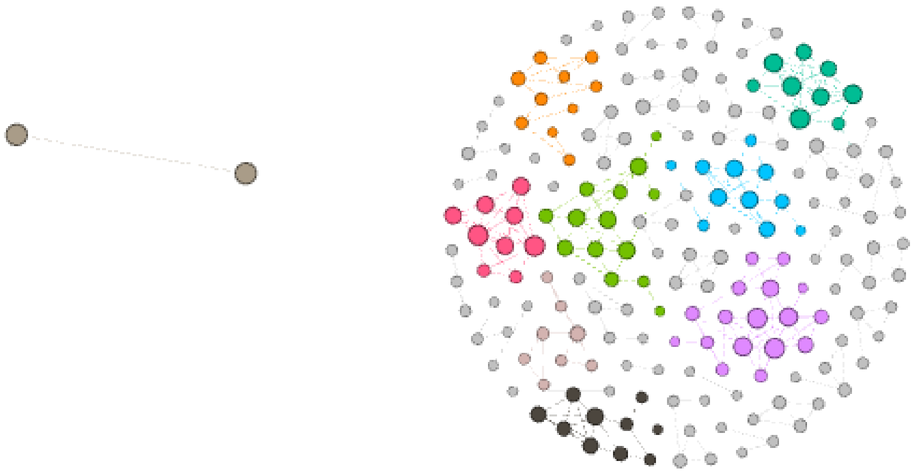
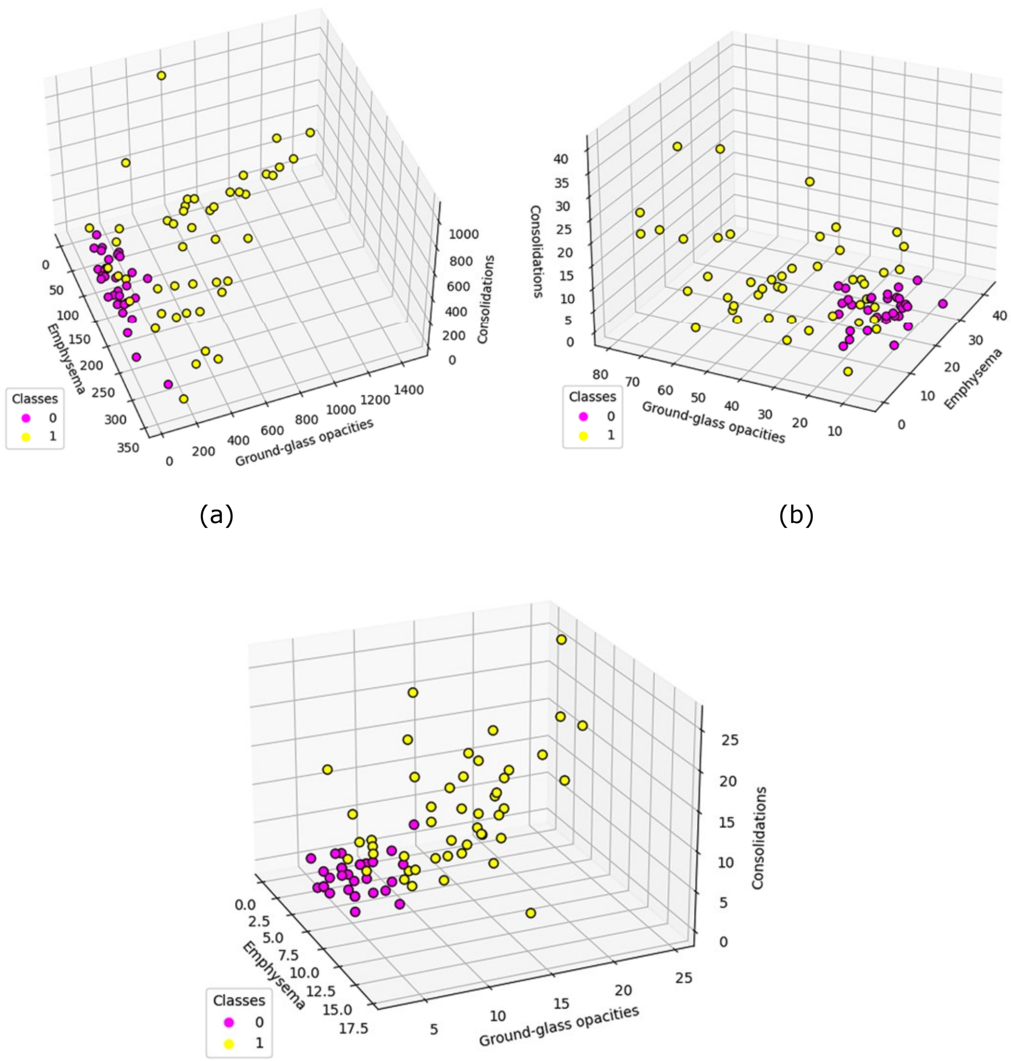


Figure 4.18 Consolidation layer converted into complex networks for (a) normal lung sample (b) DILD lung sample

Lastly, complex networks representing the two GGO samples (Fig 4.18 (a) and (b)) could not be more different. Multiple clusters can be seen in the DILD sample (marked in purple, light green, blue, pink, and black).

4.19. Fitting Complex Networks metrics

Given the initial results for the whole lot revealed an obvious distinction between the control set of HRCTs and the other types of disease categories, the following step would naturally be to determine a pattern or a certain network metric that could best approximate the different types of network models.



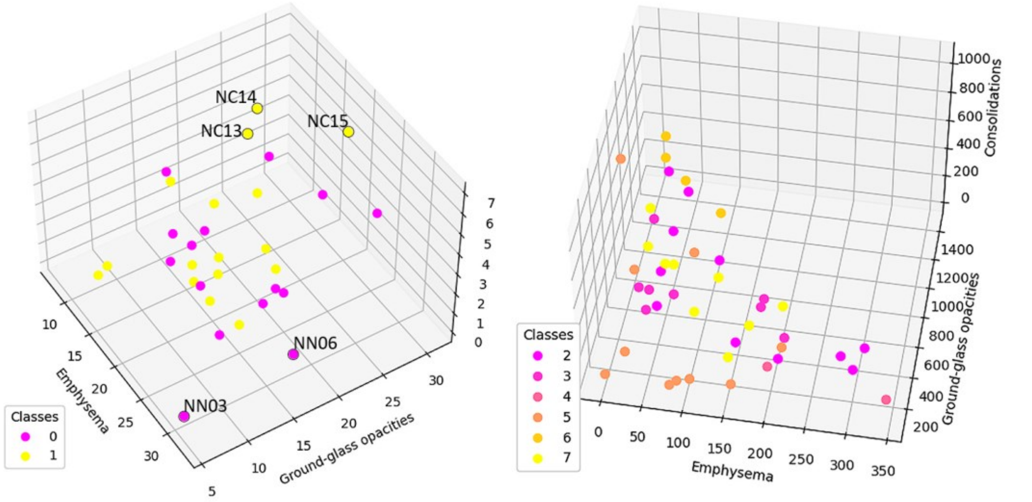
(a)

(b)

(c)

Figure 4.19 Degree distributions comparisons when different network metrics are considered: (a) Total count (b) Average count (c) Maximum degree [111]

In Figure 4.19, healthy lungs (control group) are represented as Class 0 (bright pink), while affected DILD lungs are symbolized by Class 1 (yellow).



(a) (b)

Figure 4.20 (a) Normal set distribution based on average degree. The two different classes (bright pink – Class 0 and yellow – Class 1) both represent healthy lungs but they are split into two categories: normal lungs diagnosed before the Covid pandemic, respectively during the Covid era. (b) DILD diagnosed population distribution based on average degree. The six different classes represent different types of lung affections (UIP, probable UIP, UIP and Emphysema, OP, HP, Sarcoidosis) [111]

Fig. 4.20. shows two very distinct population lots. While the normal population (Fig. 4.20 a) is relatively tight, there are still a few outliers which will be explained separately. Fig 4.20. (b) however tells a different story, with all affected lung CTs grouped by individual pathologies. These pathologies, part of the ILD category, all present the three HU analyzed bands (Emphysema, GGO, Consolidation), yet the combination and quantity differs from one to another

4.20. Discussion

The starting point of the above research material was the intention to generate and analyze a complex network model which would represent real HRCT lung images.

From a network science perspective, one approach to representing real-world systems is through their degree distributions. Understanding what these degree distributions reveal about the system itself is a matter of mathematically and programmatically pinpointing the function types which would best approximate these types of distributions.

4.20.1 Normal lungs model function

Previous research shows that these ecosystems fall into the categories of logarithmic or power law functions [117]. Indeed, the research hereby conducted (Fig. 4.21) comes as a confirmation that, for example, normal lungs samples (Fig 4.11 a, Fig 4.14 a) comply with a logarithmic distribution given that the predetermined biological resolution is used (Rd = 4).

A question arises then, whether these systems could also fit into a power law function, even with an Rd variation, in order to cover for all the biological variations. This is not the case, however, because the attempts to fit a power function to the normal population set concluded in a less precise approximation of the degree distributions, and thus, proved to be less relevant than the logarithmic one.

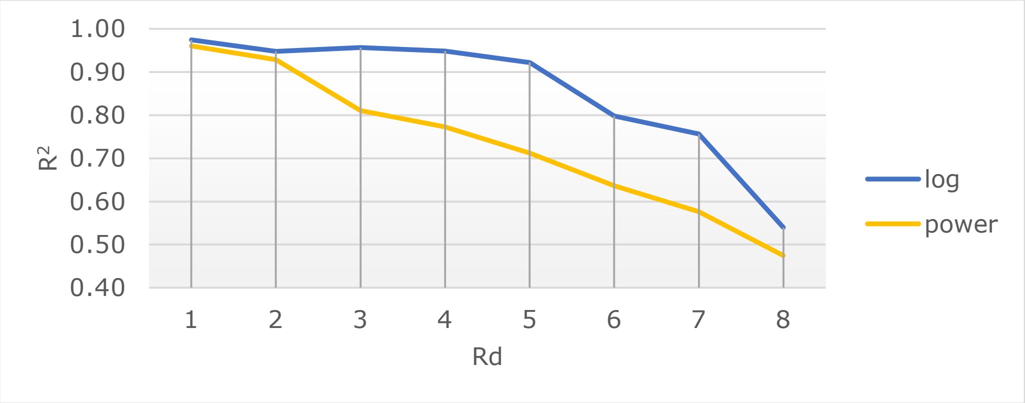


Figure 4.21 The average coefficient of determination (R^2) for logarithmic and power distributions based on radial distance (Rd)[111]

These findings can be explained by the fact that biological systems are diverse and have different particularities based on whether they are feedback systems or not. Previous research shows that biological systems without tightly coupled feedback loops can be represented as a logarithmic model rather than a power distribution [118], and this proves to be the case with lungs, as well. This is indeed also validated through the current proposed model.

4.20.2 Pathological lungs model function

On the opposite side, affected lungs describe a completely different function model (Fig 4.11 b, Fig. 4.14 b, Fig. 4.17 b). In this case, polynomial functions best fit the model rather than logarithmic ones, and this is also validated by previous literature results [1], [119]. These show that DILD pathologies are considered proliferative processes in terms of inflammation and fibrosis, and although the cause of such processes may not be a virus, their development and propagation behave similarly.

It is worth mentioning that the polynomial function best approximating all studied cases is not the same for all, and the maximum degree may also differ within the range [2,8]. This is explained by the fact that the pathologies were multiple and each of them is a combination of the 3 HU bands. A more in-depth study is required to be able to differentiate between types of illnesses, together with far larger data sets, that is representative of each of them.

So far, this study formulates a powerful indicator in the fact that normal lungs have different modeling functions from affected lungs, however, we cannot yet tell which specific pathology a patient suffers from.

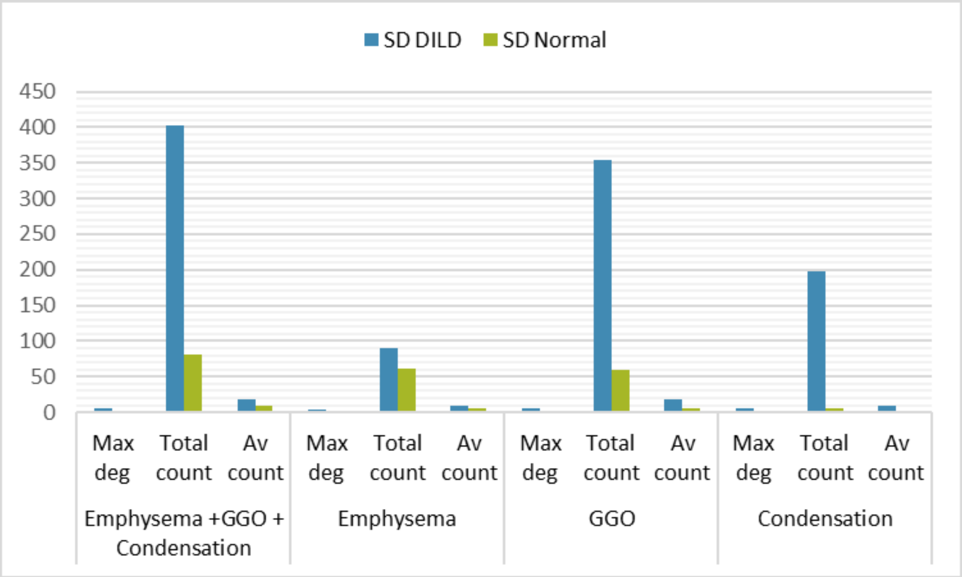


Figure 4.22 Relative comparisons of standard deviation for DILD-affected lungs and normal lungs considering all HU bands, based on maximum degree, total count, and average degree

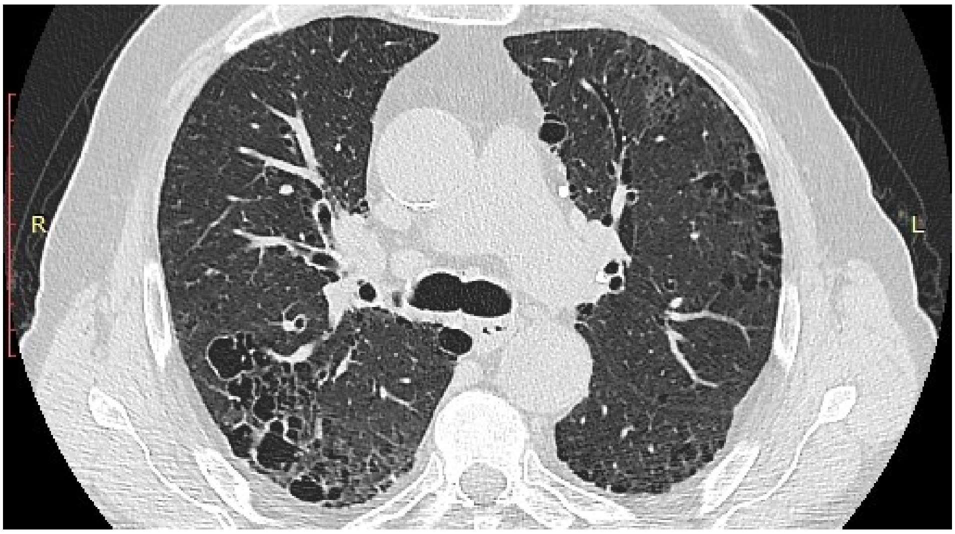
Further analysis (Fig. 4.22 above) confirms that the chosen models are valid, and also validates what one could intuitively observe with the naked eye in some cases (Fig 4.19 and 4.20). When comparing standard deviations for all patients' data series, the result is as shown in Fig 4.22 (above). Key indicators (maximum degree, total count, average count) evaluated for both categories of lungs – healthy and DILD- and for all 3 HU bands (separately and combined), reveal that there is a clear delimitation between normal lung networks and pathological ones.

4.20.3 Medical validation of models

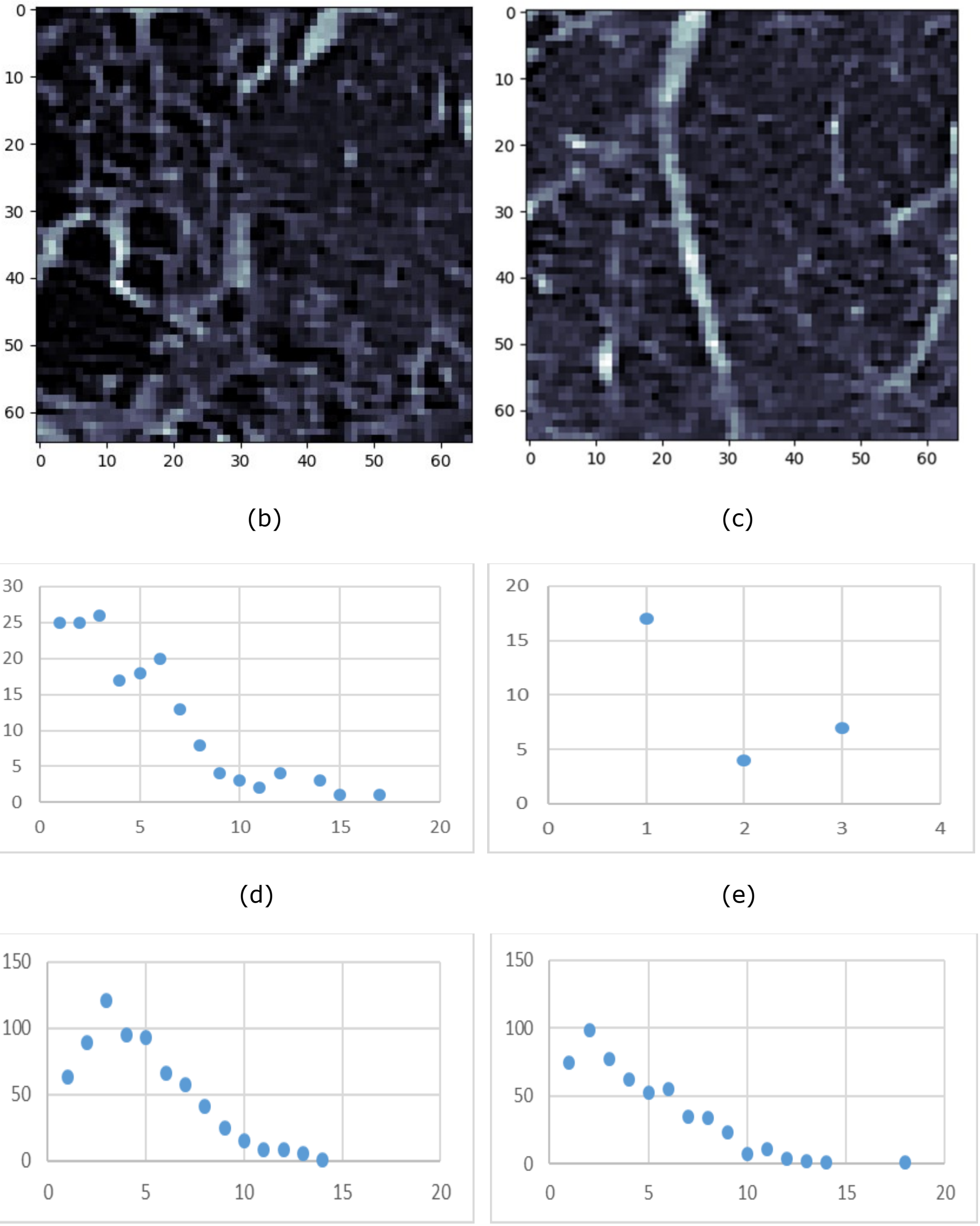
From a medical science perspective, there are a number of cases (outliers) that need to be addressed in order to validate the aforementioned models. Figure 4.20 a) shows a few cases that might seem to fall out of the proposed models. However, there are multiple factors to be taken into consideration with nodes NC13, NC14, and NC15 (higher GGO and consolidation). First, the clinical definition of a healthy (or normal) was understood differently due to the fact that NC13 and NC14 are patients recovering from covid. As a result, it is considered normal that scattered artifacts of GGO and consolidation might still be found in the lungs, given the disease evolution. On the other hand, NC15, also an outlier, reflects the case of a patient whose HRCT was performed prior to developing COVID, when a negative PCR test also infirmed the disease. Later on (approximately two days later), the patient did develop a severe case of COVID, and this diagnosis was also confirmed by a positive PCR test. These cases show that the models defined for both the control group and the pathological one could be used to detect early signs of potential lung illnesses reflected by modifications in the lung tissue.

In terms of pre-COVID outliers (NN group), although the algorithm showed some of them as being close to the boundary of normal, these differences can be explained by patient particularities such as heavy smoking (NN06, NN03). These aspects are often treated as such when diagnosing patients.

The proposed model also deals well with overlapping patterns. Real-life cases may display a combination of HU bands which would normally not be a characteristic to one single pathology. This has also been observed in the following case study, where the patient suffered from a mix of IPF (reflected through a pronounced GGO layer) and Emphysema.



(a)



(b)

(c)

(d)

(e)

(f)

(g)

Figure 4.23 (a) HRCT slice under analysis (b) Sample 1 (c) Sample 2 (d) Degree distribution for sample 1 on the emphysema layer (e) Degree distribution for sample 2 on the emphysema layer (f) Degree distribution for sample 1 on the GGO layer (g) Degree distribution for sample 2 on the GGO layer

Figure 4.23 shows a case of different samples taken from the same patient, where one of them also includes an additional affection (an emphysema bubble Fig 4.23 (b)). However, the decomposition of layers offered by the proposed algorithm offers the advantage of deconstructing the lung tissue into multiple layers and allowing the medical specialist to analyze them either separately or in combination, depending on the case. This supports a better diagnosis process, given that here, an examiner could confirm that although the underlying disease (IPF) has been indicated by the GGO degree distributions (Fig 4.23 f, g), there may always be additional overlapping patterns (Fig 4.23 d, e) which can easily be observed in isolation, thus reducing the complexity or the difficulty of interpreting HRCTs with the naked eye. The CN algorithm has proven in this case that it can successfully be applied to overlapping patterns.

4.20.4 T-test model validation

A t-test is further presented as a statistical demonstration that the algorithm and model work in a comprehensive manner. This independent sample t-test assuming unequal variances was performed against the two sets of samples: normal and DILD.

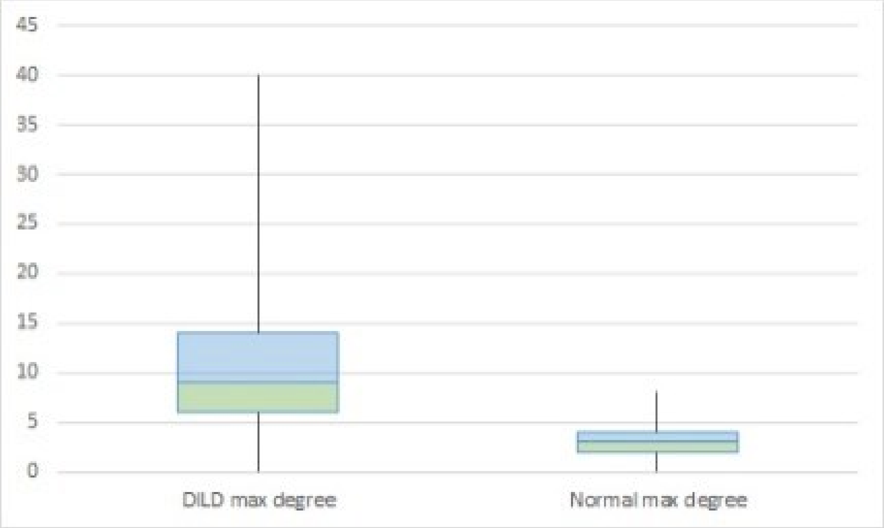


Figure 4.24 Box plot for DILD (left) and Normal (right) for complex network parameter maximum degree [111]

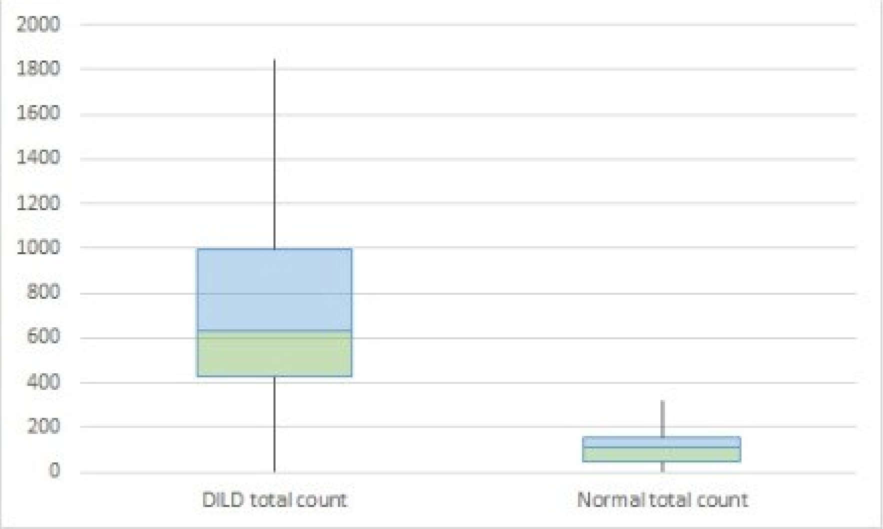


Figure 4.25 Box plot for DILD (left) and Normal (right) for complex network parameter total count[111]

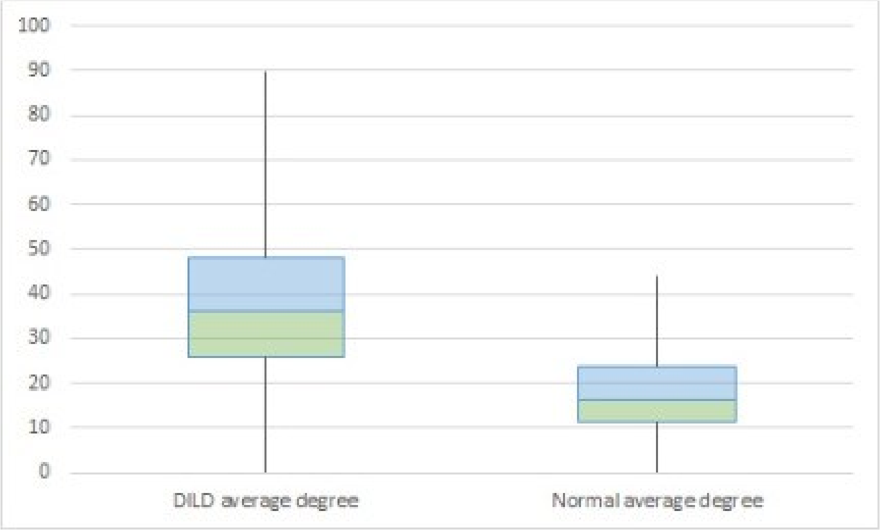


Figure 4.26 Box plot for DILD (left) and Normal (right) for complex network parameter average degree[111]

	Maximum degree		**Total count**		**Average count**	
	DILD	Normal	DILD	Normal	DILD	Normal
Mean	15.96875	7.032258	846.5692	7.1	51.65253	32.53397
Variance	39.45933	3.365591	206084.5	3.334483	362.9068	113.4483
Obervations	30	30	30	30	30	30
Hypothesized Mean Difference	0		0		0	
Df	82		64		92	
T Stat	10.49451		14.9084		6.288591	
P (T≤t) one-tail	3.97E-17		8.52E-23		5.31E-09	
T Critical one-tail	1.663649		1.669013		1.661585	
P (T≤t) two-tail	7.93E-17		1.7E-22		1.06E-08	
T Critical two-tail	1.989319		1.99773		1.986086	

Table 4.3 Statistical comparisons[111]

Table 4.4 concludes the outcome of the test, revealing that the measured *p* is smaller than 0.05 (3.97 × 10−17, 8.52 × 10−23, and 5.31 × 10−9). Having all the pieces, we then calculate the test statistic. Knowing that *t* statistic can be calculated with the following formula:

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2 + s_2^2 - 2\rho s_1 s_2}{n}}} \tag{4.1}$$

it can be concluded that the *t-statistic* (*t Stat* in Table 4.4) set of values [10.49, 14.91, 6.29] is larger than the *t-critical* set [1.98, 1.99, 1.98].
This result proves that, indeed, the null hypothesis is rejected. In other words, the differences observed between the two analysed sets (groups) have a 95% confidence of not being due to chance.

4.20.5 Comparisons with other HRCT analysis tools

This section aims to analyze the advantages and disadvantages of the proposed algorithm (complex networks model) against other field technologies or methodologies. When considering the classical diagnosis process, it resembles an information-gathering procedure, where medical specialists (radiologists and doctors) assess the HRCT but also acquire medical data from multiple sources or lung tests to complete the picture. This is not necessarily a standardized process, but involves a combination of analytical thinking, and "clinical sense" (based on experience and prior knowledge), adding to this a subjective evaluation of the patient's illness progression[91]–[94].

In terms of tools and computer-aided diagnosis, one of the most popular ones is Caliper [101], a paid standalone off-the-shelf solution, which uses a mix of HRCTs and extra Pulmonary Function Tests (PFT) to be able to measure lung expansion and give a potential diagnosis.

Method/ Tool	**HRCT only**	**Analytical**	**Empirical**	**Standalone**	**Measurement**
Doctor	N	Y	Y ("clinical sense")	Y	Subjective
Caliper [101]	N, PFT	Y	N	Y	Yes, 1 dimensional size
Zrimec [102], [103]	Y	Y	N	Mostly	Maybe
Machine learning	Y	N	Y	Maybe	Maybe
Complex networks model	Y	Y	N	N	Yes, 3 dimensional

Table 4.4 Methodology comparisons[111]

There is a major drawback in research-stemmed approaches such as Zrimec [102], [103] or even machine learning (ML) algorithms[29]–[31], [95] in the fact that these methods rely on programmatically processing the HRCT yet they do not yield a quantifyable measurement of the illness, or if they do, it is at best a volumetric one. The majority of ML algorithms are based on proper classification and pattern recognition but no means of quantification.

Unlike the hereby proposed model, these other methods (Table 4.5) do not offer a way of mathematically representing affected lung patches. The complex networks approach proposed here, however, offers a way to quantify and qualify pathological processes in three dimensions (3 axes). Admittedly, this method is not a standalone one yet, although the implementation of future improvements is already underway, and needs to be validated against a larger training set in order to improve its classification capabilities.

5. Analysing Numerical Patterns

5.1. Enhancing Imagistic Interstitial Lung Disease Diagnostic

The proposed algorithm (described in Chapter 4) aims to provide more than an analysis tool for helping medical specialists assess lung HRCTs. Early detection is one other goal this complex networks model strives to play a role in. When it comes to Diffuse Interstitial Lung Diseases, it is of paramount importance that the primary diagnosis is corroborated with a series of previous HRCT scans from the patient's medical records. This allows for specific DILD patterns to develop and be observed and caught in the early stages [120]. One of the challenges in identifying this category of pathologies is that even with the right tools and a comprehensive medical record, they have an unpredictable temporal evolution, depending on the patient characteristics. One other influential factor is the type of predominantly affected tissue and its progression slope: inflammation or fibrosis.

In addition to classic imaging techniques, medical specialists use other functional lung investigations to assist in DILD diagnosis such as Pulmonary Function Tests (PFT). Recent research [121] implies that HRCT imaging could have a role to play in evaluating the diffusion capacity of the lungs for CO2 (DLco)[122].

There are still other indicators such as the modified ILD-GAP (Gender, Age, Physiology, ILD subtype) score which can be used in enhancing or supporting DILD diagnosis [123], [124], however, they are usually not considered an early detection mechanism, but rather a way of categorizing pathologies or creating a mortality prediction model [125].

5.2. Computer-aided diagnosis

The field of computer-aided diagnosis is still a developing one. While some of the existing techniques are based on artificial intelligence, neural networks, or machine learning [29]–[31] there are gaps that have not been covered yet, and areas needing improvement. One of the weak points these software applications have is the fact that they fail to represent the dynamics of illness development. These tools function by evaluating HRCTs as a one-time operation and their outcome refers strictly to the current evaluated image. Overmore, some of them cannot offer a proper conclusion unless input data is accompanied by pulmonary function tests, for instance, in the case of CALIPER.

Although the technique entailed by some of these tools involves analyzing image samples of various dimensions (15 x 15 x 15 px 3D cubes for Caliper, 11 x 11 px for Zrimec [95]) none of them manage to use this type of analysis in such a way that they offer an early DILD diagnosis and classification based on deterioration rate and affected lung volume.

This uncharted territory offers a great opportunity for approaches such as pattern matching and complex networks [111].

The two main points that this approach aims to solve are the following:

1. Quantitatively evaluating DILD illness evolution with the help of a complex networks analytical model
2. Contributing in early detection of DILD pathologies

5.3. HRCT lot selection

The total number of analyzed HRCTs is 96, split into two categories as follows:

1. Number of normal lung patients was 31

2. Number of DILD-affected patients was 65, each of them having a series of 2, 3, or even 4 HRCTs taken at different points in time

All data used for this research study has had patient written approval to be used for research purposes. The patient lot contained various people with roughly the same age and sex profile, and for all of them, there were additional function test results available, gathered from previous medical consults such as PFTs (forced vital capacity (FVC), spirometry, DLco).

All HRCTs study candidates had already been analyzed and categorized by three specialists and were selected based on multiple preestablished DILD-related criteria:

1. They all presented typical signs of the following interstitial lung diseases: sarcoidosis (S), idiopathic pulmonary fibrosis (IPF), organizing pneumonitis (OP), non-specific interstitial pneumonia (NSIP), hypersensitivity pneumonitis (HP). These pathologies present various combinations of the three basic lesions: Consolidation (C), Ground Glass Opacities (GGO), and Emphysema. These have also been defined within the previously described CN algorithm. The respective HU bands known in radiology have already been established in Chapter 4.
2. Preprocessing of the HRCT set was available: an experienced radiologist with a vast experience in imagistic diagnosis indicated the lung areas which the algorithm should focus on. Each of these pulmonary sections presented typical expressions of the above-mentioned pathologies: NSIP, S, OP, HP, and IPF.

HRCTs were all stored in the DICOM format and provided by the National Fibrosis Center database.

5.4. Pathological alterations specificities

As with the previous experiment, the selected HU bands are the same under scrutiny here: Consolidation (and reticulation) (C), Ground Glass Opacities (GGO), and Emphysema (E). The HU bands corresponding to these types of textures are the previously established intervals: Emphysema [-1024,-977], GGO [-977,-703], and Consolidation [-100, 5)[27, p. 19], [114, p. 19], [115].
Each of these layers is described in the literature according to their specificities[98], [126].

Emphysema (E) has a round or even polygonal aspect[127], [128], presenting a low attenuation on the HRCT. On the grayscale, this translates into dark shades of gray or even black.

Ground glass opacities (GGO) presents itself as a „foggy" area, less easy to isolate or visually separate from the neighboring elements, yet unlike the Emphysema lesion type, this one has an increased opacity [129]. In terms of pixel shades, this layer is represented by brighter shades, and lighter grays.

Finally, the consolidation layer (C) which is the most dense type of lesion, and as a result, it visually looks more compact on an HRCT. This is a consequence of the thickening of the intra and interlobular septa of the secondary pulmonary lobule which is due to interstitium injuries [130]. Pixel colors are even brighter than GGO.

In real life, in the case of DILD pathologies, there is almost never a case where one single lesion of these is found alone. Usually, there is a combination of them, in different proportions, overlapping, creating patterns. These patterns might sometimes look like a textbook definition of such DILD, yet other times they might be more elusive[131]–[137].

This is where algorithms such as the one presented in this thesis could provide major support to the diagnosis process, by separating these lesion layers and helping doctors better differentiate between them.

5.5. Complex Networks approach and data processing

As with the experiment presented in Chapter 5 the proposed complex networks algorithm is run against all HRCTs in both data sets (normal and pathological).

The following steps are performed, as extensively described in Chapter 4:

1. They are sampled to 65 x 65 px sections of the relevant lung area
2. All pixels converted to their HU equivalend
3. Layered into the three basic lung lesions (E, GGO, C) thus obtaining three separate images with the isolated lesion type
4. Convert images into complex networks, similar to [104], [105]
5. Analyze specificities according to network metrics

5.6. Relevant network metrics

The goal of this study was to identify the progression of DILD pathologies and as such, the complex networks algorithm and the resulting network should reflect the biological lung specificities in such a way as to describe an evolving process in terms of shape, dimension, but also as density (which could also be translated into interconnectedness)[138].

As a result, the most relevant network metrics to take into account are:

- Maximum degree - the maximum degree a single node can have. For a single vertex, this would be the total sum of in-degree and out-degree in a directed network, however, in an undirected network (the current case) there is no differentiation between them.

$$deg_{max} = deg(V_x) \big| \{ V_x \in G(V,E) \wedge \nexists V_y \big| deg(V_y) > deg(V_x) \} \tag{5.1}$$

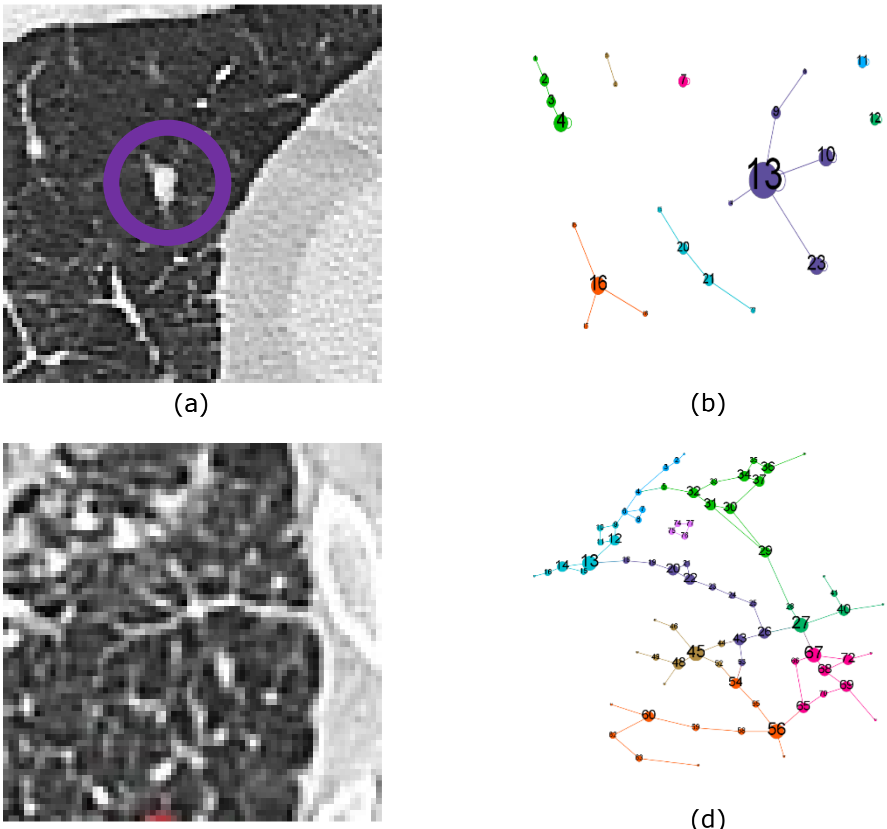
- Total degree count: the sum of connections in the network

$$T_{deg} = \sum deg(V_x) \text{ where } V_x \in G(V,E) \tag{5.2}$$

- Average degree count: average number of connections per node.

$$avg_{deg} = 2 * \frac{T_{deg}}{N_V} \tag{5.3}$$

where $N_V = count(V)$.

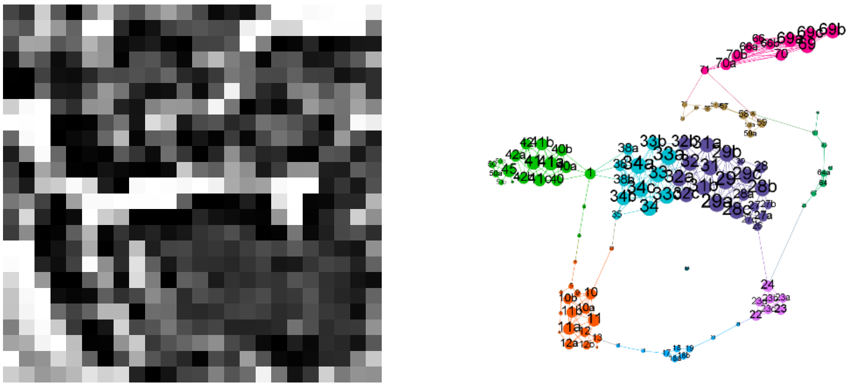


(a)

(b)

(c)

(d)



(e) (f)

Figure 5.1 Comparative illustrations of biological lung samples and their associated CN measurements[139] (a) CT section with a micronodule in the center (b) CN depicting micronodule CT (c) CT section with sarcoidosis(perilymphatic micronodules) (d) CN depicting sarcoidosis CT (e) CT section with honeycombing cysts (f) CN depicting honeycombing CT

Figure 5.1 above shows different types of CT samples with affected lungs and their equivalent complex networks counterparts. In figure 5.1 b), d) and f) vertex size is proportional to its degree (number of total connections towards other nodes) and each one is assigned a numeric ID for better identification. Cluster size and positioning within the network is meant to reflect the original pixel positioning within the DICOM image. Vertex colorization is only meant as a visual differentiating factor between communities and is not correlated with the original pixel grayscale shade. Edges share the same characteristics and width, except for the color, given that they are not weighted. The HRCT slice scale in Fig. 5.1 (a), (c), and (e) are not identical as the main purpose of this side-by-side comparison was to show the resulting complex networks.

These three network measurements mentioned above have been presented in Fig 5.1. a) shows a micronodule (circled in purple) that gets converted into a group (cluster) of nodes within its CN equivalent (b) (the purple group with the highest degree).

In terms of total degree count (the second chosen metric) Figure 5.1 c) and e) present roughly the same number of total links yet the average degree metric shows a whole different story: while the average degree for the sarcoidosis sample is $avg_{deg}S\approx2$ (also reflected by what one can observe with the naked eye – a multitude of sparse nodes or small clusters with few connections), in the case of honeycombing cysts, $avg_{deg}hc=5.8$ which translates into a smaller number of vertices yet a higher density of connections per node.

As an overall conclusion for the three chosen metrics, each of them reflects an aspect of the analyzed lung sample: total count represents the cumulated damage on the whole patch, average count suggests the confinement of these lesions while the maximum degree shows the highest intensity of a lesion. As a result, the two aspects under scrutiny (interconnectedness and size) are rightfully reflected by the above metrics, following the proposed paradigm of HU layering for the three HU bands.

The approach proposed in this experiment (progression assessment) involved assessing a lot of patients which had an array of successive scans performed over a longer period of time. The lung sections (pertaining to the same patient) selected for analysis were taken from approximately the same area of the lung (for anatomical continuity). With the same approach as described in Chapter 4, every sample was split into the three HU bands (E, GGO, C) and compared to its subsequent ones.

Progression is measured as the variation of a parameter within a certain period of time, and this leads to the engineering formula for speed. Nevertheless, in order for this speed to be a valid measurement across all individuals in a lot, it needs to be defined as a relative speed, rather than an absolute speed. This relative speed formula is defined in equation (5.4):

$$v = \begin{cases} \frac{(s-s_0)}{s_0 \times t}, for\ s_0\ != 0 \\ \frac{s}{t} \qquad\quad , otherwise \end{cases} \tag{5.4}$$

where *s* is the metric under analysis and *s_0* is the equivalent point in the reference sample used for normalization. *t* is a measurement unit expressed in years, given that normally, DILD patients come in for control visits every year[140]. Calculating its value involves a fairly trivial formula, where the difference in days between the two HRCT dates is calculated as a delta, and then normalized by dividing it by the number of days in a year (considered the default as being 365 days). In the following equation, *t_0* represents the oldest HRCT taken for a patient, while *t_1* is the HRCT currently being evaluated.

$$t = DAY\ (DATE\ (t1) - DATE(t0))/365 \tag{5.5}$$

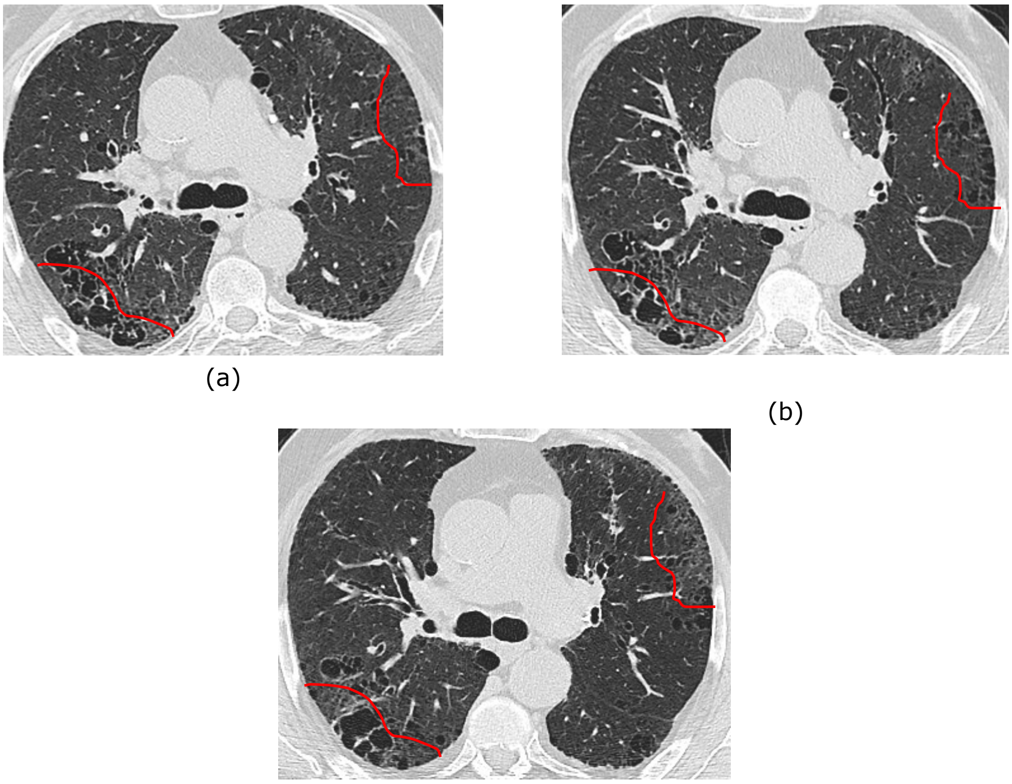
The default number of days in a year could just as well have been limited to 360, by taking the example of typical financial calculations, however, the most important aspect here is the normalization type consistency. For the purpose of this experiment, equation 5.5 above was used for normalization.

5.7. Results

5.7.1 Case studies

As an example of the above-proposed approach, two different patients with different pathologies have been selected and processed. For each of them, multiple samples from their HRCT were marked and compared, displaying the imagistic progression of their illnesses.

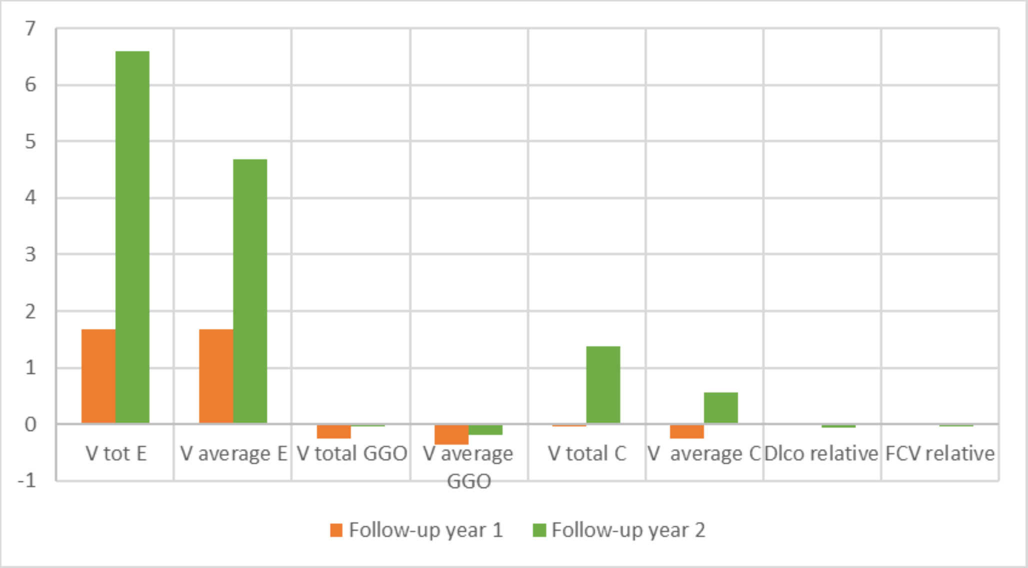
Figure 5.2 shows a typical patient suffering from UIP + emphysema (CPFE phenotype).



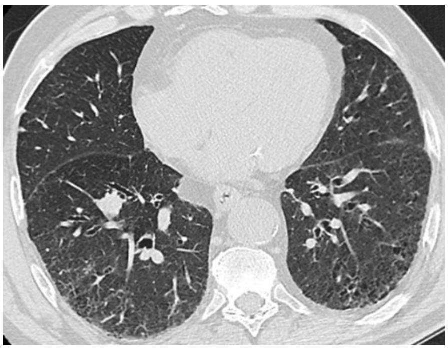
(a)

(b)

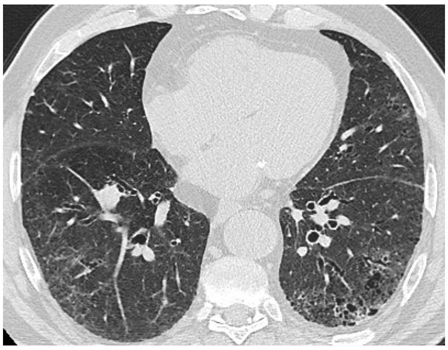
(c)


7
6
5
4
3
2
1
0
-1
V tot E
V average E
V total GGO
V average GGO
V total C
V average C
Dlco relative
FCV relative
🟧 Follow-up year 1
🟩 Follow-up year 2

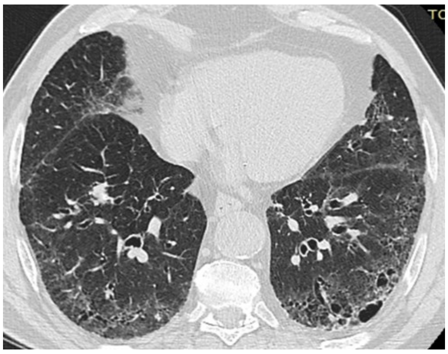
(d)



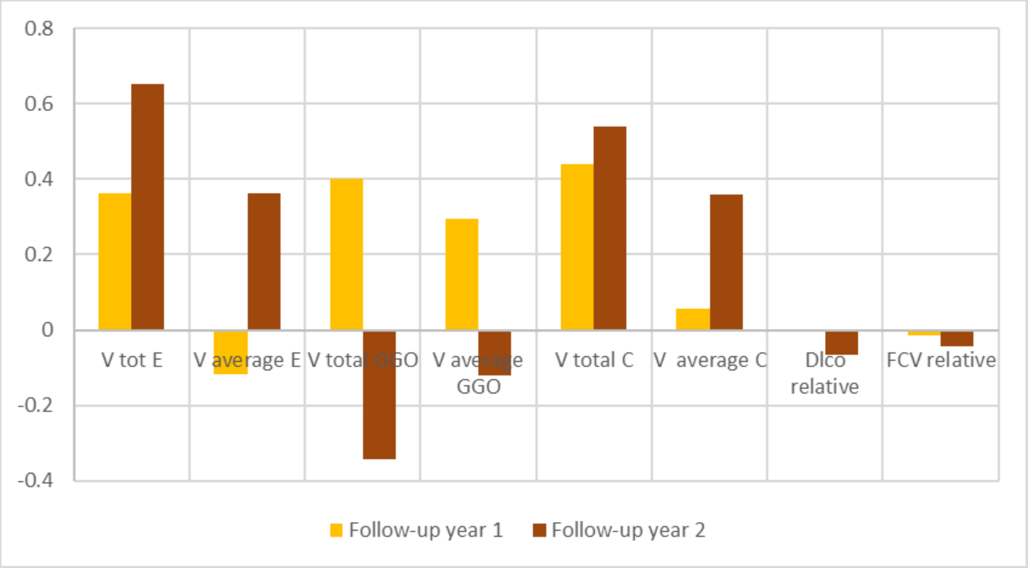
(e)



(f)



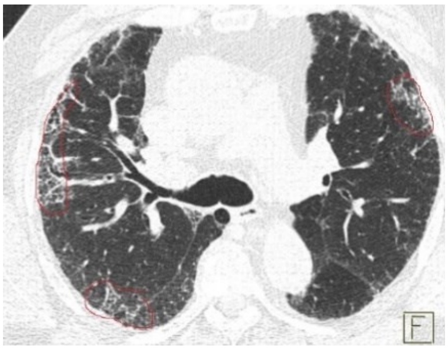
(g)



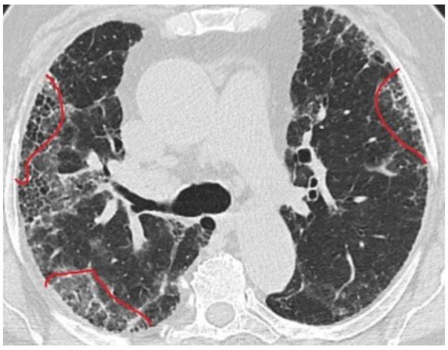
(h)

Figure 5.2 Case study for lung HRCT (axial), UIP+E pattern(CPFE) patient progression for three consecutive years (a) t0 year - Superior lung region (b) t1 year - same lung region in the following year (c) t2 year – same region in the second year (d) Relative speed variations on the superior lung slice (E, GGO and C HU layers) (e) t0 year – Basal lung area (f) t1 year – same basal lung area in the following year (g) t2 year – same lung area in the second year (h) Relative speed variations for basal lung sections (E, GGO and C HU layers) [139]

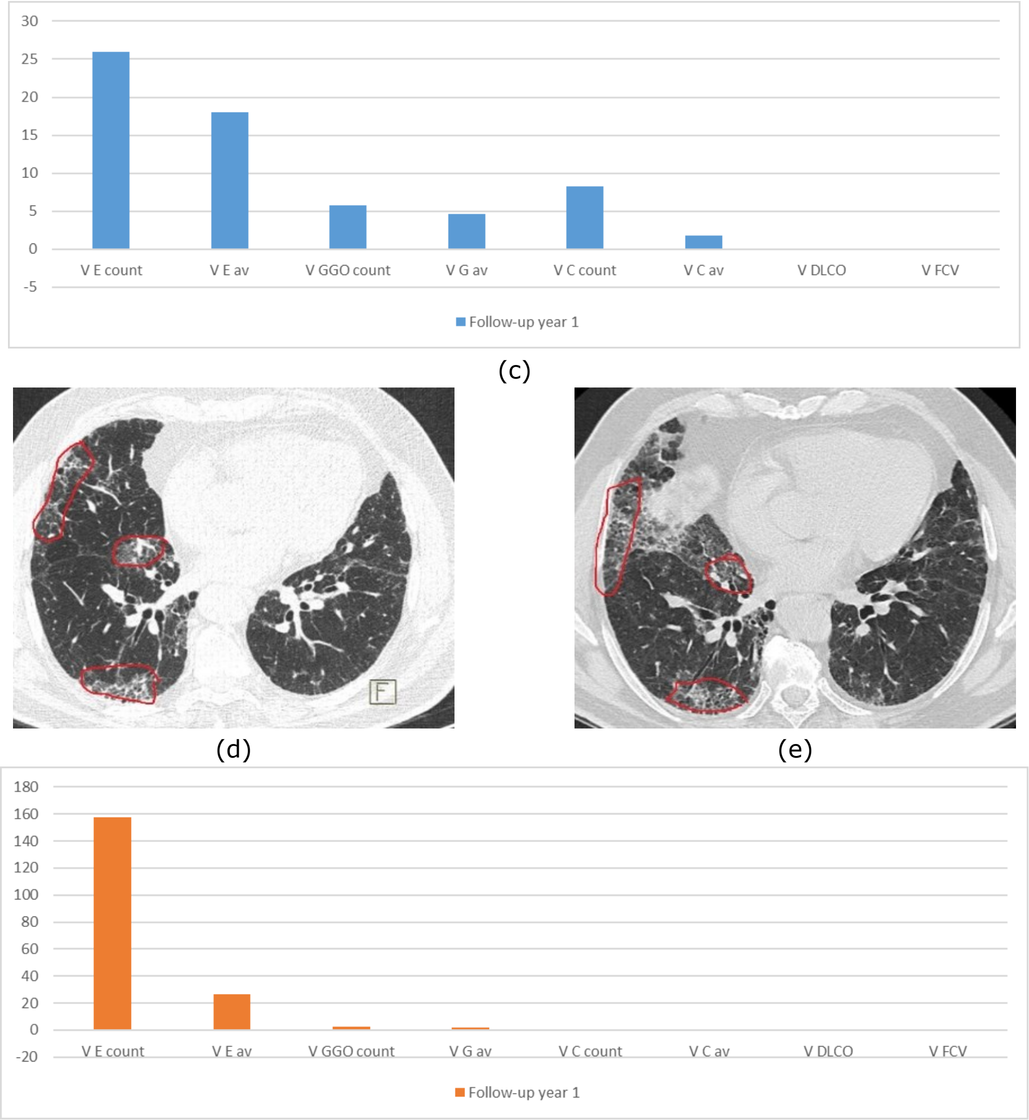
The following represents another patient with a classic NSIP pattern dynamic.



(a)



(b)



(c)

(d)

(e)

(f)

Figure 5.3 Case study for an NSIP+E patient progression (a) t_0 year - superior lung area in initial t0 year (b) t_1 year - superior lung area in the following year (c) Relative speed variations on the superior lung slice (E, GGO, C) (d) t_0 year - basal lung area in first year (e) t_1 year - basal lung region axial HRCT slice in the following year (f) Relative speed variations on the basal lung slice (E, GGO, C) [139]

5.7.2 Progression speed

The patient HRCT sets were processed and analyzed according to the previously described approach. A t-test was calculated considering the relative speed parameter (for every one of the HU layers) compared to the DLco relative variation. The t-test evaluated all of the HRCTs: normal and DILD. This type of analysis could also be performed against the maximum degree metric, however, when talking about progression, dealing with peak values does not render the proper result.

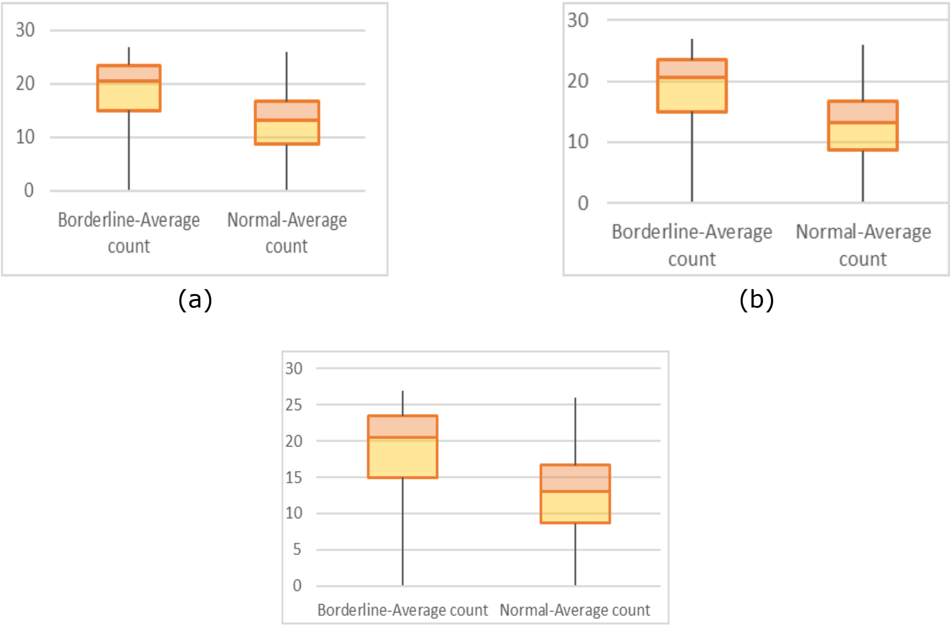
HU Layer	Total count / DLco	Avg count VS DLco	Parameters
E	1.81144865	***2.297734923***	t Stat
	0.038529988	***0.013194925***	P(T≤t) one-tail
	2.016692199	***2.015367574***	T Critical two-tail
GGO	-1.334981884	-1.82528253	t Stat
	0.092702764	0.035714932	P(T≤t) one-tail
	1.987934206	1.987934206	T Critical two-tail
C	-1.334981884	-1.82528253	t Stat
	0.093421672	0.035996812	P(T≤t) one-tail
	1.999623585	1.992543495	t Critical two-tail

Table 5.1 T-test calculus for the relative speed of progression in HU bands against DLco [139]

Looking at the results, Table 5.1 shows that the null hypothesis is rejected by the E band and the values corresponding to Average count / DLco [2.297734923; 0.013194925; 2.015367574].

5.7.3 Early detection hypothesis validation

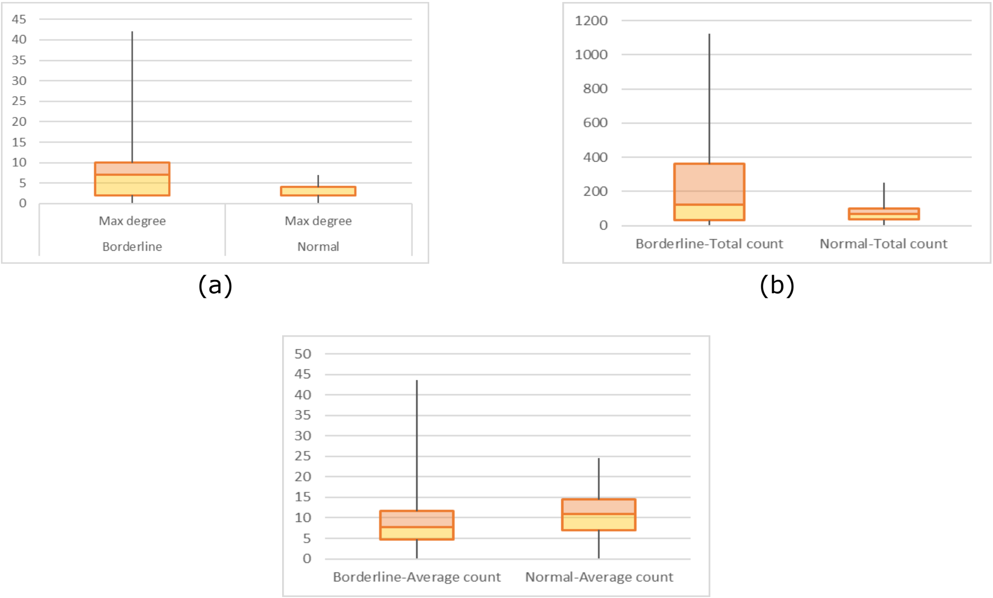
To assess whether early detection is possible with the described method, the patients were divided into two sets of HRCTs: the normal ones, and the cases with early signs of DILD having some fairly decent functional characteristics (0-3 GAP-ILD points, DLco between 70-85%). DLco values were represented as an interval centered around the 80% standard limit in order to be able to include the early stages of affected alveolar-capillary membranes. The evaluated metrics (maximum degree, average count, total count) are shown in Fig 5.6 and then a t-test is performed in Table 5.2.



(a) (b)

(c)

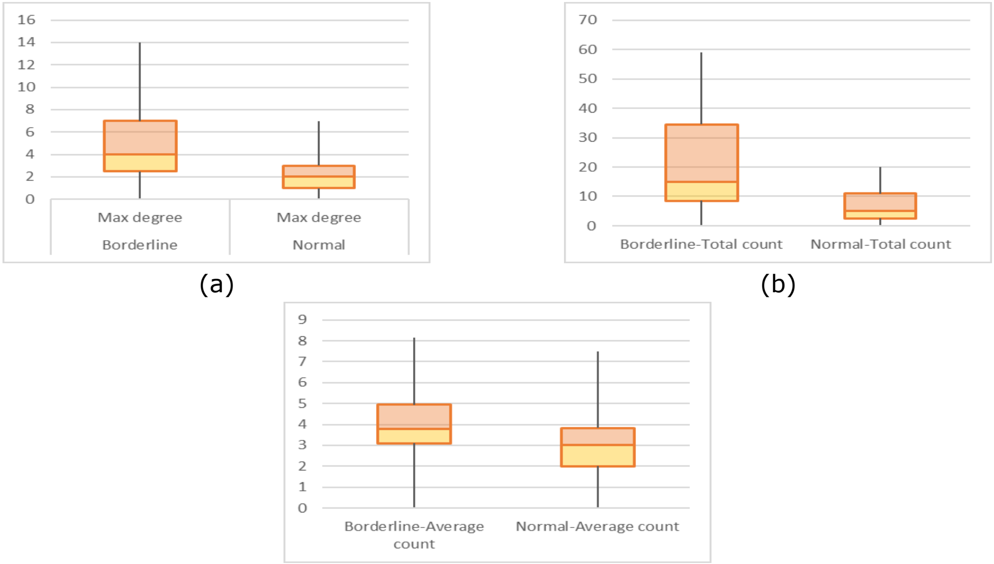
Figure 5.4 Network metrics on Borderline normal / Normal – Emphysema layer (a) Max degree (b) Total count (c) Avg count [139]



(a) (b)

(c)

Figure 5.5 Network metrics on Borderline normal / Normal - GGO layer (a) Max degree (b) Total count (c) Avg count [139]



(a) (b)

(c)

Figure 5.6 Network metrics on Borderline normal / Normal - Consolidation layer (a) Max degree (b) Total count (c) Avg count [139]

Table 5.2 below presents the t-test calculations for the three HU layers and the aforementioned metrics, highlighting the series rejecting the null hypothesis.

HU Layer	Max Degree	Total count	Avg Count	Parameters
E	−0.357327012	−0.33960631	−1.194455411	t Stat
	0.361362738	0.367964892	0.119667428	P(T ≤ t) one-tail
	2.02107539	2.02107539	2.02107539	t Critical two-tail
GGO	***2.362901118***	***2.496174465***	***2.132901092***	t Stat
	0.016568972	***0.012345754***	***0.023097162***	P(T ≤ t) one-tail
	2.144786688	***2.131449546***	***2.093024054***	t Critical two-tail
C	***2.787128882***	***2.910253494***	1.723111496	t Stat
	0.006593367	***0.005384188***	0.048371727	P(T ≤ t) one-tail
	2.119905299	***2.131449546***	2.055529439	t Critical two-tail

Table 5.2 Statistical t-test results for borderline and normal lungs [139]

5.8. Discussion

The analysis of a patient suffering from UIP and Emphysema is presented in Figure 5.2. In this context, there are two regions included (superior and basal), being considered a classical imagistic progression of such pathology.

In terms of relative variation speed on each HU band, the complex networks proposed model offers these values. The calculated speed is considered to be typical to a selected area and displays a relative variation in characteristics within a time frame. This type of measurement is meant to underline rapid changes in lung tissue consistency and is not an absolute value. Another advantage of this approach is its granularity, given that the complex networks algorithm analyses affected lung tissue with an area of around 3 mm [111], which is quite small for the human eye to be able to detect, considering that a medical specialist looks at the whole slice and all of the overlapping layers as a whole, and not separately.

Analyzing the relative speed variation with regards to the patient in Fig. 5.2 there are certain visible aspects reflected through this metric. Although on the Emphysema layer both the superior and basal affected areas evolve in year 1 and year 2, their progression speeds are definitely different. The inferior (basal) area is almost 10 times slower in deteriorating than the one in the upper part of the lung, where the cumulated emphysema and honeycombing areas are also better expressed.

In terms of Consolidation, the density of this layer appears to have increased both in the upper part and the basal one. This type of progression can be categorized as a usual pathological process of lesion progression.

There are also some small GGO variations highlighted by the proposed method, more precisely in the lower plane (Fig 5.2 e,f,g,h). This type of granularity in detecting small-scale changes has proved to be more efficient than what medical specialists could detect when looking at the same HRCT slices. Clinical patient data from the following year (year 1) suggests that some symptoms had slightly worsened, unlike year 2 however. This explains the subtle variation in the calculated relative speed and validates the complex networks model capability for early detection. The results from the following year, however, suggest that the functional status is almost unchanged, suggesting the idea that the subtle change detected by the complex network model was indeed a premature one.

For the second patient with a NSIP case, presented in Fig 5.3, the relative speed variation in the case of Emphysema shows a definite increase on the total count axis, while the average degree shows only a medium increase. The GGO layer shows a small up variation in t_1 compared to the initial evaluation while the Consolidation layer also has an up variation reflected by the multilayer cysts and their walls. Functional parameters show no significant variation, which also comes to show that the complex networks method has detected a premature change.

In terms of the whole group of patients (both normal and borderline), Table 5.1 validates the testing of hypothesis 1, and more precisely, that the proposed algorithm has the capability of characterizing DILD progression in an accurate and quantitative manner. This theory is proved to be true due to the fact that comparison results between DLco and complex networks measurements display valid resemblance. The sole outlier is the Emphysema layer (column Average count versus DLco) which has been highlighted accordingly in the table. The complex networks metrics have shown to be similar to the biological terms for the Emphysema layer, however, the average intensity proved to be more suitable to describe it than its equivalent functional parameter variance.

Given the results shown in Figure 5.5 and Table 5.2, it can be concluded that this direction requires a more in-depth analysis. No early detection has been proven on the Emphysema layer due to having no statistical difference between the early diagnosis and the normal one. In terms of GGO and Consolidation, however, the statistical results show that on these layers there are notable differences and the proposed model can detect changes. In the case of GGO, the null hypothesis is rejected. As for the Consolidation layer, the maximum degree and total count measurements successfully allow for early DILD detection, but not on account of the average degree. To sum up, the complex networks method has been proven effective in the case of well-contoured consolidation lesions, but cannot yet detect diffuse early consolidations in their premature state. In conclusion, with regards to the second hypothesis, the complex networks model is effective in detecting early changes on the GGO band, partially true on the Consolidation band and false on the Emphysema band.

6. Conclusions and personal contributions

6.1. Conclusions to the Hybrid 3D Network Layout Visualization Algorithm

The Hybrid layout proposed in Chapter 3 aims to become a viable and appealing choice when talking about 2D and 3D graph layout solutions. This type of algorithm has proved that it can successfully fill the gap or the missing piece in network software which was until now lacking in terms of network structure, from a visual point of view. This approach offers a special type of view (2D and 3D) into the network elements, and also considers the user's point of interest (centering the user's cluster of interest as a 3D entity). It also enhances the network view with a gridded aspect, which gives a more practical approach to graph representation, offering clear quantifiable visual queues (graded *xOyz* axes). 3D graph interaction and manipulation are also possible with this solution, giving users the possibility to turn the resulting image on all axes.

The additional third dimension (*Oz* axis) offers more space for a better node distribution across the entire 3D canvas, as well as providing a layering of elements, giving a different significance to each of them.

Further improvements aim at improving generation time, especially in terms of edge plotting, as well as reducing edge crossings, thus enhancing graph readability.

6.2. Conclusions to Approach 2 – A novel method for Computer Tomography image interpretation

A novel complex networks-based method that transforms and interprets HRCTs has been developed and tested. This approach analyzes medical data in a three dimension manner, involving mathematical function fitting. The overview and algorithm development sections in Chapter 4 describe the algorithm stages in a detailed manner.

There is a solid argumentation regarding the analyzed sample size (65 x 65 px) and a comparison with existing field tools justifies taking a step further and enlarging the interest area. Sample dimensions are also consistent with the anatomical details (secondary pulmonary lobule).

Vertex connectivity and the associated radial distance selection are supported by an extensive experiment regarding pixel similarity criteria corroborated with complex networks attachment principles, as well as evaluating the role of network density and clusterization in the process of medical diagnosis.

The proposed algorithm uses Hounsfield Unit intervals both for image layering and simultaneously as similarity criteria for potentially linked nodes, allowing for more granularity when observing lung injuries. These ranges are also dependent on the device and resolution of the machine used to perform the HRCTs.

The results section presents a full algorithm execution and its comparative outcome for two sample patients, a normal one (baseline sample) as well as a pathological one.

Furthermore, the discussion section justifies the coherence and correctness of the complex networks-based algorithm from a Systems Science standpoint, by entailing the metric of degree distribution as a central device for system representation. We also showcase clusterization as a network measurement which shows distinct discrepancies between the two studies HRCT lots: healthy (normal) and pathological patients. From a Medical Science viewpoint, the model is validated by its faithful and fine-grained representation of clinical data and this can prove to be crucial in the diagnosis process.

Finally, the comparisons with other present days tools underline the advantages of using the proposed method: offering a comprehensive measuring instrument for qualitative and quantitative analysis.

Among the drawbacks, we mention its inability to work as off-the-shelf software yet, as well as the particularly modest lot size used for testing it. Improvements regarding the aforementioned are to be addressed in future research, with a much larger training set, as well as user-friendly customizations (a graphical user interface) which would offer it a more appealing look.

In conclusion, the new complex networks algorithm has been shown to be extremely useful in the DILD diagnosis process, by transforming lung HRCTs into quantifiable and qualifiable structures.

6.3. Conclusions to Approach 3 - Enhancing Interstitial Lung disease diagnostic

This complex networks approach has been developed as a means of support for the process of diagnosing and managing DILDs. For this purpose, there were two hypotheses being evaluated: early detection and accurately evaluating disease progression, as these are the two main factors that medical specialists have been struggling with. Especially when it comes to IPF, for instance, existing techniques or technologies in the field have, so far, not been able to provide effective answers.

For the first of the proposed hypotheses, regarding progression, the proposed complex networks algorithm and overall approach have proven to be a success. Given its precision and 3 mm granular lesion detection, this approach has shown a very good connection with the clinical symptoms at such a level that could not be reached by the usual functional tests. It is worth noting here that the Emphysema layer constitutes an exception to this conclusion (average count measurement), however, this is easily surpassed by the other five metric axes.

As for the second hypothesis, regarding early detection, the best results have been for the GGO and Consolidation band. From a medical perspective, the GGO and

Consolidation layers and their expression are very important in detecting common DILD states. This is an essential capability demonstrated by an algorithm of this kind, and is very fitting to DILD, unlike other software in the field such as Caliper.

In terms of challenges that are still considered for improvement, this approach still takes a fair amount of time, which is directly correlated to the dimension of the analyzed window. Added to this is the time spent preprocessing the HRCT slices, but this can be overcome through the use of other CAD rather than manually.

Future improvements involve integrating this algorithm into a larger software solution and combining swifter ML segmentation and pattern recognition competencies with the steadier but more granular and precise complex networks in-depth analysis.

6.4. Personal Contributions

The aim of the current research paper has been to offer insight into biomedical patterns via two major complex networks approaches: an innovative visualization layout for complex networks as well as a Computer Aided Diagnosis algorithm based on complex networks.

The first part of this thesis deals with a hybrid 3D layout algorithm developed for visualizing complex networks (biological data) in a completely new manner: combining 2D and 3D dimensions to display and enhance data features that might otherwise be underrepresented with other available layout tools. Thus, regarding personal contributions, the following have been achieved:

- I conducted an analysis and evaluation of current state-of-the-art tools for complex networks visualization published in [4], [9], [13];
- I proposed and implemented a new force-directed complex networks layout algorithm in a 3D space published in [83], [84];
- I gathered and tested (performance-wise) and fine-tuned against multiple various-sized data sets pertaining to the biology domain;
- I implemented a new co-variance approach for vertex similarity and positioning within the network layout [83];
- I conducted a comparison between the current approach and other state-of-the-art tools[83], [84].

Considering the second and third parts of this thesis (Chapters 4 and 5) regarding the Novel Method for Computer Tomography image interpretation, the following contributions have been brought:

- I participated in the multidisciplinary approach of creating a complex networks method for modeling lung HRCTs published in [111], [139];
- I proposed and implemented the HRCT processing algorithm based on complex networks published in [111];

- I created and defined a model responsible for layering and analyzing DICOM images according to the three dimensions: Emphysema, GGO, and Consolidation[111];
- I processed and curated all data sets used for this approach, after their inclusion in the lot by the medical couterparts;
- I participated in defining a proper window size (crop dimensions of 65 x 65 pixels) for the analyzed data sets, so as to satisfy two major requirements: medical relevance (covering the basic lung unit – secondary pulmonary lobule) and delivering an adequate performance and throughput;
- I conducted an in-depth analysis of various radial distance dimensions and the impact or relevance of such values for the final complex networks model of lung tissue [111], [139];
- I defined a similarity metric (delta) based on HU bands (Emphysema, GGO, Consolidations) and validated it against medical data, together with a medical team of specialists [111], [139];
- I identified the mathematical functions fitting the complex networks degree distributions associated with normal lungs and affected lungs[111], [139];
- I conducted an analysis for assessing the accuracy of such mathematical functions in the case of normal lungs and diseased lungs, as well as the extent to which they reflect System Science as well as Medical Science[111], [139].
- I validated the proposed model from a Network Science perspective[111], [139].
- I analyzed the model for three different complex networks metrics: total count, average degree, and maximum degree[111], [139].
- I participated in defining a new measurement type and mathematical formula for fibrosis progression speed. Based on the classical notion of speed (defined as variation over time), the new relative variation speed formula was proposed[139].

6.5. Future research directions

The algorithms developed and proposed in this research paper are two different approaches to visualizing and identifying patterns in the realm of biomedical data. However complex they are, there still are multiple points where improvements could lead to better and more accurate results.

Regarding the 3D Hybrid visualization layout algorithm, performance improvements would involve: faster processing and rendering times, reduce edge overlapping, refining the force-directed formula for node placement, customizing it with new metrics which would enhance different network aspects, and developing a user-friendly graphical user interface (GUI).

Referring to the second part of the thesis and the complex networks HRCT processing algorithm, the following enhancements are planned to be developed: an automated system for lung pre-segmentation, a continuous automated assessment across the whole lung in all three dimensions, integrating an artificial intelligence classifier for identifying outlier lung regions, a user-friendly GUI to render it more appealing to non-technical users, using cloud technologies for storing and processing HRCTs – allowing for more processing resources and better performance.

Potential future developments also aim to accommodate the analysis of other categories of human tissue and associated ailments, given that this method is generic enough to be adapted to any type of entry data, as long as it is represented in the standardized DICOM format.

6.6. Publications

To this date, I have the following publications submitted, accepted and presented at international conferences, journals or books:

1. Ancușa Versavia, **Broască Laura**. (2015). "A Method to Pinpoint Undiscovered Links in Genetic and Protein Networks". Studies in health technology and informatics. 210. 771-5. 10.3233/978-1-61499-512-8-771.

2. **Broască Laura**, Ancușa Versavia, Ciocârlie Horia. (2016). "Bioinformatics Visualisation Tools: An Unbalanced Picture". Studies in health technology and informatics. 228. 760-4.

3. **Broască Laura**, Ancușa Versavia-Maria, Ciocârlie Horia (2017). "Social Media as Medical Validator". BRAIN. Broad Research in Artificial Intelligence and Neuroscience, 8(3), pp. 47-56.

4. **Broască Laura**, Ancușa Versavia, Ciocârlie Horia. (2019). "A Qualitative Analysis on Force Directed Network Visualization Tools in the Context of Large Complex Networks". 656-661. 10.1109/ICSTCC.2019.8885641.

5. **Broască Laura**, Ancușa Versavia, Ciocârlie Horia. (2020). "A 3D Surface Fitting Layout for Complex Networks Visualization". Studies in health technology and informatics. 272. 362-365. 10.3233/SHTI200570.

6. **Broască Laura**, Ancușa Versavia, Ciocârlie Horia. (2020). "Towards a Hybrid Layout for Complex Networks Visualization". 118-123. 10.1109/ICSTCC50638.2020.9259656.

7. Trușculescu Adriana, **Broască Laura**, Ancușa Versavia, Manolescu Diana, Tudorache Emanuela, Oancea Cristian. (2021). "Managing Interstitial Lung Diseases with Computer-Aided Visualization". In: Kumar Bhoi, A., Mallick, P.K., Narayana Mohanty, M., Albuquerque, V.H.C.d. (eds) "Hybrid Artificial Intelligence and IoT in Healthcare". Intelligent Systems Reference Library, vol 209. Springer, Singapore 10.1007/978-981-16-2972-3_12.

8. **Broască Laura**, Trușculescu Ana, Ancușa Versavia, Ciocârlie Horia, Oancea Cristian-Iulian, Stoicescu Emil, Manolescu Diana. (2022). "A Novel Method for Lung Image Processing Using Complex Networks". Tomography (Ann Arbor, Mich.). 8. 1928-1946. 10.3390/tomography8040162.

9. Trușculescu Ana, Manolescu Diana, **Broască Laura**, Ancușa Versavia, Ciocârlie Horia, Pescaru Camelia, Vaștag Emanuela, Oancea Cristian. (2022). "Enhancing Imagistic Interstitial Lung Disease Diagnosis by Using Complex Networks". Medicina. 58. 1288. 10.3390/medicina58091288.

10. Awarded 1st prize for best paper at the 27th Congress of the Romanian Society of Pneumology (Sinaia 2-6 Nov 2022) for the paper: Trușculescu Ana, Ancușa Versavia, **Broască Laura**, Manolescu Diana, Pescaru Camelia, Oancea Cristian. "Diffuse Interstitial Lung Diseases Computer-Aided Imaging Diagnosis, with the help of Complex Networks".

References

[1] R. Pastor-Satorras, C. Castellano, P. V. Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Rev. Mod. Phys.*, vol. 87, no. 3, pp. 925–979, Aug. 2015, doi: 10.1103/revmodphys.87.925.

[2] K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabási, "The human disease network," *PNAS*, vol. 104, no. 21, pp. 8685–8690, 2007.

[3] C. J. Stam and J. C. Reijneveld, "Graph theoretical analysis of complex networks in the brain," *Nonlinear Biomed. Phys.*, vol. 1, no. 1, p. 3, Jul. 2007, doi: 10.1186/1753-4631-1-3.

[4] V. M. Ancusa and L. Broasca, "A Method to Pinpoint Undiscovered Links in Genetic and Protein Networks," *Stud. Health Technol. Inform.*, vol. 210, pp. 771–775, 2015.

[5] V. Marx, "The big challenges of big data," *Nature*, vol. 498, no. 7453, pp. 255–260, Jun. 2013, doi: 10.1038/498255a.

[6] Y. Li and L. Chen, "Big biological data: challenges and opportunities.," *Genomics Proteomics Bioinformatics*, vol. 12, no. 5, pp. 187–189, Oct. 2014, doi: 10.1016/j.gpb.2014.10.001.

[7] A. S. da Mata, "Complex Networks: a Mini-review," *Braz. J. Phys.*, vol. 50, no. 5, pp. 658–672, Oct. 2020, doi: 10.1007/s13538-020-00772-9.

[8] M. Zanin *et al.*, "Combining complex networks and data mining: Why and how," *Comb. Complex Netw. Data Min. Why How*, vol. 635, pp. 1–44, May 2016, doi: 10.1016/j.physrep.2016.04.005.

[9] L. Broască, V. Ancuşa, and H. Ciocârlie, "Bioinformatics Visualisation Tools: An Unbalanced Picture," *Stud. Health Technol. Inform.*, vol. 228, pp. 760–764, 2016.

[10] C. V. R. Hütter, C. Sin, F. Müller, and J. Menche, "Network cartographs for interpretable visualizations," *Nat. Comput. Sci.*, vol. 2, no. 2, pp. 84–89, Feb. 2022, doi: 10.1038/s43588-022-00199-z.

[11] S. Heymann and B. L. Grand, "Visual Analysis of Complex Networks for Business Intelligence with Gephi," in *2013 17th International Conference on Information Visualisation*, 2013, pp. 307–312. doi: 10.1109/IV.2013.39.

[12] O.-H. Kwon, T. Crnovrsanin, and K.-L. Ma, "What Would a Graph Look Like in this Layout? A Machine Learning Approach to Large Graph Visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 478–488, Jan. 2018, doi: 10.1109/tvcg.2017.2743858.

[13] L. Broasca, V.-M. Ancusa, and H. Ciocarlie, "A Qualitative Analysis on Force Directed Network Visualization Tools in the Context of Large Complex Networks," in *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*, Oct. 2019, pp. 656–661. doi: 10.1109/ICSTCC.2019.8885641.

[14] A. Christe *et al.*, "Computer-Aided Diagnosis of Pulmonary Fibrosis Using Deep Learning and CT Images.," *Invest. Radiol.*, vol. 54, no. 10, pp. 627–632, Oct. 2019, doi: 10.1097/RLI.0000000000000574.

[15] J. Yanase and E. Triantaphyllou, "The seven key challenges for the future of computer-aided diagnosis in medicine.," *Int. J. Med. Inf.*, vol. 129, pp. 413–422, Sep. 2019, doi: 10.1016/j.ijmedinf.2019.06.017.

[16] N. Safari-Alighiarloo, M. Taghizadeh, M. Rezaei-Tavirani, B. Goliaei, and A. A. Peyvandi, "Protein-protein interaction networks (PPI) and complex diseases.," *Gastroenterol. Hepatol. Bed Bench*, vol. 7, no. 1, pp. 17–31, Winter 2014.

[17] L. Vulliard and J. Menche, "Complex Networks in Health and Disease," *Syst. Med.*, pp. 26–33, 2021, doi: 10.1016/B978-0-12-801238-3.11640-X.

[18] K. Zengler and L. S. Zaramela, "The social network of microorganisms — how auxotrophies shape complex communities," *Nat. Rev. Microbiol.*, vol. 16, no. 6, pp. 383–390, Jun. 2018, doi: 10.1038/s41579-018-0004-5.

[19] D. Otasek, J. H. Morris, J. Bouças, A. R. Pico, and B. Demchak, "Cytoscape Automation: empowering workflow-based network analysis," *Genome Biol.*, vol. 20, no. 1, p. 185, Sep. 2019, doi: 10.1186/s13059-019-1758-4.

[20] L. Chang, G. Zhou, O. Soufan, and J. Xia, "miRNet 2.0: network-based visual analytics for miRNA functional analysis and systems biology," *Nucleic Acids Res.*, vol. 48, no. W1, pp. W244–W251, Jun. 2020, doi: 10.1093/nar/gkaa467.

[21] R. Pizarro *et al.*, "Using Deep Learning Algorithms to Automatically Identify the Brain MRI Contrast: Implications for Managing Large Databases.," *Neuroinformatics*, vol. 17, no. 1, pp. 115–130, Jan. 2019, doi: 10.1007/s12021-018-9387-8.

[22] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Spec. Issue Deep Learn. Med. Phys.*, vol. 29, no. 2, pp. 102–127, May 2019, doi: 10.1016/j.zemedi.2018.11.002.

[23] J. N. Itri, R. R. Tappouni, R. O. McEachern, A. J. Pesch, and S. H. Patel, "Fundamentals of Diagnostic Error in Imaging," *RadioGraphics*, vol. 38, no. 6, Oct. 2018, doi: 10.1148/rg.2018180021.

[24] A. Trușculescu, L. Broască, V. M. Ancușa, D. Manolescu, E. Tudorache, and C. Oancea, "Managing Interstitial Lung Diseases with Computer-Aided Visualization," in *Hybrid Artificial Intelligence and IoT in Healthcare*, A. Kumar Bhoi, P. K. Mallick, M. Narayana Mohanty, and V. H. C. de Albuquerque, Eds. Singapore: Springer Singapore, 2021, pp. 245–271. doi: 10.1007/978-981-16-2972-3_12.

[25] J.-Z. Cheng *et al.*, "Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans," *Sci. Rep.*, vol. 6, no. 1, p. 24454, Apr. 2016, doi: 10.1038/srep24454.

[26] W. Shi *et al.*, "A deep learning-based quantitative computed tomography model for predicting the severity of COVID-19: a retrospective study of 196 patients.," *Ann. Transl. Med.*, vol. 9, no. 3, p. 216, Feb. 2021, doi: 10.21037/atm-20-2464.

[27] L. Li *et al.*, "Using Artificial Intelligence to Detect COVID-19 and Community-acquired Pneumonia Based on Pulmonary CT: Evaluation of the Diagnostic Accuracy," *Radiology*, vol. 296, no. 2, pp. E65–E71, Aug. 2020, doi: 10.1148/radiol.2020200905.

[28] D. Müller and F. Kramer, "MIScnn: a framework for medical image segmentation with convolutional neural networks and deep learning," *BMC Med. Imaging*, vol. 21, no. 1, p. 12, Jan. 2021, doi: 10.1186/s12880-020-00543-7.

[29] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ArXiv14091556 Cs*, Apr. 2015, Accessed: Feb. 06, 2022. [Online]. Available: http://arxiv.org/abs/1409.1556

[30] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec. 2014, pp. 844–848. doi: 10.1109/ICARCV.2014.7064414.

[31] S. L. F. Walsh, L. Calandriello, M. Silva, and N. Sverzellati, "Deep learning for classifying fibrotic lung disease on high-resolution computed tomography: a case-cohort study," *Lancet Respir. Med.*, vol. 6, no. 11, Art. no. 11, Nov. 2018, doi: 10.1016/S2213-2600(18)30286-8.

[32] H. Jeong, Tombor, R. Albert, Z. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651–654, 2000.

[33] A. Vinayagam *et al.*, "Controllability analysis of the directed human protein interaction network identifies disease genes and drug targets," *PNAS*, pp. 1–6.

[34] F. Ruzzenenti, D. Garlaschelli, and R. Basosi, "Complex Networks and Symmetry II: Reciprocity and Evolution of World Trade," *Symmetry*, vol. 2, pp. 1710–1744, 2010.

[35] V. Lapatas, M. Stefanidakis, R. C. Jimenez, Via, and M. V. Schneider, "Data integration in biological research: an overview," *J. Biol. Res.*, vol. 22, no. 1, 2015.

[36] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[37] R. Albert and A.-L. Barabási, "Topology of Evolving Networks: Local Events and Universality," *Phys. Rev. Lett.*, vol. 85, pp. 5234–5237, 2000.

[38] L. O. Prokhorenkova and E. Samosvat, "Global Clustering Coefficient in Scale-Free Networks," *Int. Workshop Algorithms Models Web-Graph*, 2014.

[39] M. E. Newman, "The mathematics of networks," *New Palgrave Encycl. Econ.*, vol. 2, pp. 1–12, 2008.

[40] "Modularity and community structure in networks," *Proc. Natl. Acad. Sci.*, vol. 103.23, pp. 8577–8582, 2006.

[41] R. V. D. Hofstad, *Random graphs and complex networks Vol 1*. Cambridge University Press, 2016.

[42] M. Chen, K. Kuzmin, and B. K. Szymanski, "Community detection via maximization of modularity and its variants," *IEEE Trans. Comput. Soc. Syst.*, vol. 1.1, pp. 46–65, 2014.

[43] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[44] L. C. Freeman, "Centrality in Social Networks: I Conceptual clarifications," Lausanne: Elsevier Sequoia, 1979, pp. 215–239.

[45] "A Graph-theoretic perspective on centrality," *Soc. Netw.*, vol. 28.4, pp. 466–484, 2006.

[46] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[47] A.-L. Barabási, "Network Science: The Barabasi-Albert Model." 2014. Accessed: Feb. 12, 2022. [Online]. Available: http://barabasi.com/f/622.pdf

[48] G. Basler, Z. Nikoloski, A. Larhlimi, A.-L. Barabási, and Y.-Y. Liu, "Control of Fluxes in Metabolic Networks," *Genome Res.*, vol. 7, no. 26, pp. 956–968, 2016.

[49] J. Gao, B. Barzel, and A.-L. Barabási, "Universal resilience patterns in complex networks," *Nature*, vol. 530, pp. 307–312, 2016.

[50] B. C. Coutinho *et al.*, "The Network Behind the Cosmic Web," *Cosmol. Nongalactic Astrophys.*, 2016.

[51] H. Shen, D. Wang, C. Song, and A.-L. Barabási, "Modeling and predicting popularity dynamics via reinforced poisson processes," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[52] P. R. Erdős, "On Random Graphs I.," *Publ. Math.*, vol. 6, pp. 290–297, 1959.

[53] "Graph Models." Apr. 30, 2015. Accessed: Mar. 12, 2022. [Online]. Available: https://cs.hse.ru/data/2015/04/30/1098190683/3._Graph_Models.pdf

[54] E. P. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, pp. 17–61, 1960.

[55] M. E. J. Newman, S. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Phys. Rev.*, vol. 64, 2001.

[56] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 393, pp. 440–442, 1998.

[57] A. Barrat, M. Barthelemy, and A. Vespignani, *Dynamical Process on Complex Networks*. Cambridge University Press, 2008.

[58] M. Penrose, *Random geometric graphs*. New York: Oxford University Press, 2003.

[59] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, p. P10008, Oct. 2008, doi: 10.1088/1742-5468/2008/10/p10008.

[60] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, Nov. 2002.

[61] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proc. Natl. Acad. Sci.*, vol. 101, no. 9, pp. 2658–2663, 2004, doi: 10.1073/pnas.0400054101.

[62] V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to Leiden: guaranteeing well-connected communities," *Sci. Rep.*, vol. 9, no. 1, p. 5233, Mar. 2019, doi: 10.1038/s41598-019-41695-z.

[63] H. Meyerhenke, P. Sanders, and C. Schulz, "Parallel Graph Partitioning for Complex Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 9, pp. 2625–2638, 2017, doi: 10.1109/TPDS.2017.2671868.

[64] A. Kirkley and M. E. J. Newman, "Representative community divisions of networks," *Commun. Phys.*, vol. 5, no. 1, p. 40, Feb. 2022, doi: 10.1038/s42005-022-00816-3.

[65] V. Faber, "Clustering and the continuous K-means algorithm," *Los Alamos Sci.*, vol. 22, Jan. 1994.

[66] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software," *PloS One*, vol. 9, p. e98679, Jun. 2014, doi: 10.1371/journal.pone.0098679.

[67] T. M. J. Fruchterman and E. M. Reingold, "Graph Drawing by Force-directed Placement," *SOFTWARE—PRACTICE Exp.*, vol. 21, no. 1, pp. 1129–1164, 1991.

[68] J. Barnes and P. Hut, "A hierarchical O(N log N) force-calculation algorithm," *Nature*, vol. 324, no. 6096, pp. 446–449, Dec. 1986, doi: 10.1038/324446a0.

[69] Y. Hu, "Efficient and High Quality Force-Directed Graph Drawing," *Math. J.*, vol. 10, pp. 37–71, Jan. 2005.

[70] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, 1989, doi: https://doi.org/10.1016/0020-0190(89)90102-6.

[71] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An Open Source Software for Exploring and Manipulating Networks," Mar. 2009. doi: 10.13140/2.1.1341.1520.

[72] N. Developers, "NetworkX." NetworkX, 2010. Accessed: Jul. 12, 2022. [Online]. Available: https://networkx.org/documentation/networkx-1.7/overview.html

[73] N. Developers, "Random Geometric Graph." NetworkX, 2010. Accessed: Jul. 12, 2022. [Online]. Available: https://networkx.github.io/documentation/networkx-1.7/examples/drawing/random_geometric_graph.html

[74] G. Csárdi and T. Nepusz, "iGraph." 2013. Accessed: Jul. 12, 2022. [Online]. Available: https://igraph.org/

[75] P. Shannon *et al.*, "Cytoscape: a software environment for integrated models of biomolecular interaction networks.," *Genome Res.*, vol. 13, no. 11, pp. 2498–2504, Nov. 2003, doi: 10.1101/gr.1239303.

[76] G. Pavlopoulos, D. Paez Espino, N. Kyrpides, and I. Iliopoulos, "Empirical Comparison of Visualization Tools for Larger-Scale Network Analysis," *Adv. Bioinforma.*, vol. 2017, Jul. 2017, doi: 10.1155/2017/1278932.

[77] "Cytoscape - Navigation and Layout." [Online]. Available: https://manual.cytoscape.org/en/stable/Navigation_and_Layout.html

[78] N. Henry, J.-D. Fekete, and M. J. McGuffin, "NodeTrix: a Hybrid Visualization of Social Networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1302–1309, 2007, doi: 10.1109/TVCG.2007.70582.

[79] "Network Analysis and Visualization with R and igraph." 2016. Accessed: Aug. 12, 2022. [Online]. Available: https://kateto.net/netscix2016.html

[80] M. Jünger and P. Mutzel, "2-Layer Straightline Crossing Minimization: Performance of Exact and Heuristic Algorithms," *J. Graph Algorithms Appl.*, vol. 1, Jan. 1997, doi: 10.1142/9789812777638_0001.

[81] X. Y. Li and M. F. Stallmann, "New bounds on the barycenter heuristic for bipartite graph drawing," *Inf. Process. Lett.*, vol. 82, no. 6, pp. 293–298, 2002, doi: https://doi.org/10.1016/S0020-0190(01)00297-6.

[82] V. Latora, V. Nicosia, and G. Russo, *Complex Networks Principles, Methods and Applications*. 2017. Accessed: Nov. 27, 2022. [Online]. Available: https://www.complex-networks.net/datasets.html

[83] L. Broască, V.-M. Ancușa, and H. Ciocârlie, "Towards a Hybrid Layout for Complex Networks Visualization," in *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*, 2020, pp. 118–123. doi: 10.1109/ICSTCC50638.2020.9259656.

[84] L. Broasca, V. Ancusa, and H. Ciocarlie, "A 3D Surface Fitting Layout for Complex Networks Visualization," *Stud. Health Technol. Inform.*, vol. 272, pp. 362–365, Jun. 2020, doi: 10.3233/SHTI200570.

[85] M. Molina-Molina *et al.*, "Importance of early diagnosis and treatment in idiopathic pulmonary fibrosis," *Expert Rev. Respir. Med.*, vol. 12, no. 7, Art. no. 7, Jul. 2018, doi: 10.1080/17476348.2018.1472580.

[86] M. Kolb *et al.*, "Nintedanib in patients with idiopathic pulmonary fibrosis and preserved lung volume," *Thorax*, vol. 72, no. 4, Art. no. 4, Apr. 2017, doi: 10.1136/thoraxjnl-2016-208710.

[87] N. Sverzellati, "Highlights of HRCT imaging in IPF," *Respir. Res.*, vol. 14 Suppl 1, p. S3, 2013, doi: 10.1186/1465-9921-14-S1-S3.

[88] K. C. Meyer, "Diagnosis and management of interstitial lung disease," *Transl. Respir. Med.*, vol. 2, p. 4, Feb. 2014, doi: 10.1186/2213-0802-2-4.

[89] D. Manolescu, L. Davidescu, D. Traila, C. Oancea, and V. Tudorache, "The reliability of lung ultrasound in assessment of idiopathic pulmonary fibrosis," *Clin. Interv. Aging*, vol. 13, pp. 437–449, 2018, doi: 10.2147/CIA.S156615.

[90] R. M. du Bois, "An earlier and more confident diagnosis of idiopathic pulmonary fibrosis," *Eur. Respir. Rev. Off. J. Eur. Respir. Soc.*, vol. 21, no. 124, Art. no. 124, Jun. 2012, doi: 10.1183/09059180.00000812.

[91] M. Inomata *et al.*, "Clinical impact of the radiological indeterminate for usual interstitial pneumonia pattern on the diagnosis of idiopathic pulmonary fibrosis," *Respir. Investig.*, vol. 59, no. 1, Art. no. 1, Jan. 2021, doi: 10.1016/j.resinv.2020.07.001.

[92] S. L. F. Walsh, L. Calandriello, N. Sverzellati, A. U. Wells, D. M. Hansell, and UIP Observer Consort, "Interobserver agreement for the ATS/ERS/JRS/ALAT criteria for a UIP pattern on CT," *Thorax*, vol. 71, no. 1, Art. no. 1, Jan. 2016, doi: 10.1136/thoraxjnl-2015-207252.

[93] S. L. F. Walsh *et al.*, "Multicentre evaluation of multidisciplinary team meeting agreement on diagnosis in diffuse parenchymal lung disease: a case-cohort study," *Lancet Respir. Med.*, vol. 4, no. 7, Art. no. 7, Jul. 2016, doi: 10.1016/S2213-2600(16)30033-9.

[94] A. A. Trusculescu, D. Manolescu, E. Tudorache, and C. Oancea, "Deep learning in interstitial lung disease-how long until daily practice," *Eur. Radiol.*, vol. 30, no. 11, Art. no. 11, Nov. 2020, doi: 10.1007/s00330-020-06986-4.

[95] Q. Li, W. Cai, and D. D. Feng, "Lung image patch classification with automatic feature learning," *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Int. Conf.*, vol. 2013, pp. 6079–6082, 2013, doi: 10.1109/EMBC.2013.6610939.

[96] M. S. Crews *et al.*, "Automated CT Analysis of Major Forms of Interstitial Lung Disease," *J. Clin. Med.*, vol. 9, no. 11, Art. no. 11, Nov. 2020, doi: 10.3390/jcm9113776.

[97] S. L. F. Walsh and M. Kolb, "Radiological diagnosis of interstitial lung disease: is it all about pattern recognition?," *Eur. Respir. J.*, vol. 52, no. 2, Art. no. 2, Aug. 2018, doi: 10.1183/13993003.01321-2018.

[98] "Signs and Patterns of Lung Disease - Chest Radiology: The Essentials, 2nd Edition," Feb. 06, 2022. https://doctorlib.info/medical/chest/2.html (accessed Feb. 06, 2022).

[99] B. Di Muzio, T. Fahrenhorst-Jones, and A. Murphy, "HRCT chest (protocol)", doi: 10.53347/rID-68126.

[100] S. Hobbs, J. H. Chung, J. Leb, K. Kaproth-Joslin, and D. A. Lynch, "Practical Imaging Interpretation in Patients Suspected of Having Idiopathic Pulmonary Fibrosis: Official Recommendations from the Radiology Working Group of the Pulmonary Fibrosis Foundation," *Radiol. Cardiothorac. Imaging*, vol. 3, no. 1, p. e200279, Feb. 2021, doi: 10.1148/ryct.2021200279.

[101] B. J. Bartholmai *et al.*, "Quantitative CT Imaging of Interstitial Lung Diseases," *J. Thorac. Imaging*, vol. 28, no. 5, p. 10.1097/RTI.0b013e3182a21969, Sep. 2013, doi: 10.1097/RTI.0b013e3182a21969.

[102] T. Zrimec and S. Busayarat, "Computer-aided Analysis and Interpretation of HRCT Images of the Lung," 2011. doi: 10.5772/14507.

[103] A. Depeursinge, T. Zrimec, S. Busayarat, and H. Müller, "3D Lung Image Retrieval Using Localized Features," *SPIE Med. Imaging 2011 Lake Buena Vista Orlando Fla. U. S.*, Oct. 2011, doi: 10.1117/12.877943.

[104] G. V. L. de Lima, T. R. Castilho, P. H. Bugatti, P. T. M. Saito, and F. M. Lopes, "A Complex Network-Based Approach to the Analysis and Classification of Images," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Cham, 2015, pp. 322–330. doi: 10.1007/978-3-319-25751-8_39.

[105] Y. Mourchid, M. E. Hassouni, and H. Cherifi, "A General Framework for Complex Network-Based Image Segmentation," *Multimed. Tools Appl.*, vol. 78, no. 14, pp. 20191–20216, Jul. 2019, doi: 10.1007/s11042-019-7304-2.

[106] "The DICOM Standard." National Electrical Manufacturers Association. Accessed: Sep. 12, 2022. [Online]. Available: https://www.dicomstandard.org/current

[107] D. R. Varma, "Managing DICOM images: Tips and tricks for the radiologist.," *Indian J. Radiol. Imaging*, vol. 22, no. 1, pp. 4–13, Jan. 2012, doi: 10.4103/0971-3026.95396.

[108] D. R. Varma, "Free DICOM browsers," *Indian J. Radiol. Imaging*, vol. 18, no. 1, pp. 12–16, 2008.

[109] W. R. Webb, "Thin-section CT of the secondary pulmonary lobule: anatomy and the image–the 2004 Fleischner lecture.," *Radiology*, vol. 239 2, pp. 322–38, 2006.

[110] M. C. Liszewski, P. Ciet, and E. Y. Lee, "Lung and Pleura," in *Pediatric Body MRI: A Comprehensive, Multidisciplinary Guide*, E. Y. Lee, M. C. Liszewski, M. S. Gee, P. Daltro, and R. Restrepo, Eds. Cham: Springer International Publishing, 2020, pp. 1–28. doi: 10.1007/978-3-030-31989-2_1.

[111] L. Broască *et al.*, "A Novel Method for Lung Image Processing Using Complex Networks," *Tomogr. Ann Arbor Mich*, vol. 8, no. 4, pp. 1928–1946, Jul. 2022, doi: 10.3390/tomography8040162.

[112] M. Hiramatsu *et al.*, "Pulmonary ground-glass opacity (GGO) lesions-large size and a history of lung cancer are risk factors for growth," *J. Thorac. Oncol. Off. Publ. Int. Assoc. Study Lung Cancer*, vol. 3, no. 11, pp. 1245–1250, Nov. 2008, doi: 10.1097/JTO.0b013e318189f526.

[113] T. DenOtter and J. Schubert, "Hounsfield Unit," in *StatPearls [Internet]*, Treasure Island (FL): StatPearls Publishing, 2022.

[114] M. P. Belfiore *et al.*, "Artificial intelligence to codify lung CT in Covid-19 patients," *Radiol. Med. (Torino)*, vol. 125, no. 5, pp. 500–504, May 2020, doi: 10.1007/s11547-020-01195-x.

[115] R. Grassi *et al.*, "COVID-19 pneumonia: computer-aided quantification of healthy lung parenchyma, emphysema, ground glass and consolidation on chest computed tomography (CT)," *Radiol. Med. (Torino)*, vol. 126, no. 4, pp. 553–560, Apr. 2021, doi: 10.1007/s11547-020-01305-9.

[116] "DICOM Standard Browser - Pixel Value Transformation Sequence Attribute." Innolitics LLC. [Online]. Available: https://dicom.innolitics.com/ciods/enhanced-mr-image/enhanced-mr-image-multi-frame-functional-groups/52009229/00289145

[117] K. M. Smith, "Explaining the emergence of complex networks through log-normal fitness in a Euclidean node similarity space," *Sci. Rep.*, vol. 11, no. 1, Art. no. 1, Jan. 2021, doi: 10.1038/s41598-021-81547-3.

[118] M. Adler, A. Mayo, and U. Alon, "Logarithmic and Power Law Input-Output Relations in Sensory Systems with Fold-Change Detection," *PLOS Comput. Biol.*, vol. 10, no. 8, p. e1003781, Aug. 2014, doi: 10.1371/journal.pcbi.1003781.

[119] Y. Shang, "Degree distribution dynamics for disease spreading with individual awareness," *J. Syst. Sci. Complex.*, vol. 28, no. 1, pp. 96–104, Feb. 2015, doi: 10.1007/s11424-014-2186-x.

[120] H. Hatabu, G. M. Hunninghake, and D. A. Lynch, "Interstitial Lung Abnormality: Recognition and Perspectives," *Radiology*, vol. 291, no. 1, Art. no. 1, Apr. 2019, doi: 10.1148/radiol.2018181684.

[121] E. G. Hieba, E. E. Shaimaa, S. S. Dina, and A. O. Noha, "Diffusion lung capacity for carbon monoxide correlates with HRCT findings in patients with diffuse parenchymal lung disease," *Egypt. J. Bronchol.*, vol. 14, no. 1, p. 39, Nov. 2020, doi: 10.1186/s43168-020-00042-x.

[122] V. Cristian Oancea Ovidiu Fira-Mlădinescu Voicu Tudorache, "Tratat de Pneumologie pentru medici rezidenti.," in *Capitolul 3 .Metode de investigatie imagistica a patologiei pulmonare*, pp. 42–59.

[123] S. H. Lee *et al.*, "Comparison of CPI and GAP models in patients with idiopathic pulmonary fibrosis: a nationwide cohort study," *Sci. Rep.*, vol. 8, no. 1, p. 4784, Mar. 2018, doi: 10.1038/s41598-018-23073-3.

[124] C. Hyldgaard, O. Hilberg, and E. Bendstrup, "Validation of GAP score in Danish patients diagnosed with idiopathic pulmonary fibrosis," *Eur. Respir. J.*, vol. 42,

no. Suppl 57, Sep. 2013, Accessed: Jun. 15, 2022. [Online]. Available: https://erj.ersjournals.com/content/42/Suppl_57/P2367

[125] C. J. Ryerson *et al.*, "Predicting Survival Across Chronic Interstitial Lung Disease: The ILD-GAP Model," *CHEST*, vol. 145, no. 4, pp. 723–728, Apr. 2014, doi: 10.1378/chest.13-1474.

[126] "The Radiology Assistant : Basic Interpretation," Feb. 06, 2022. https://radiologyassistant.nl/chest/hrct/basic-interpretation (accessed Feb. 06, 2022).

[127] M. Takahashi *et al.*, "Imaging of pulmonary emphysema: A pictorial review," *Int. J. Chron. Obstruct. Pulmon. Dis.*, vol. 3, no. 2, Art. no. 2, Jun. 2008.

[128] D. C. Caltabiano *et al.*, "Cystic pattern in lung diseases: a simplified HRCT guide based on free-hand drawings," *ECR 2017 EPOS*, Mar. 01, 2017. https://epos.myesr.org/poster/esr/ecr2017/C-2141 (accessed Feb. 13, 2022).

[129] D. M. Hansell, A. A. Bankier, H. MacMahon, T. C. McLoud, N. L. Müller, and J. Remy, "Fleischner Society: glossary of terms for thoracic imaging," *Radiology*, vol. 246, no. 3, Art. no. 3, Mar. 2008, doi: 10.1148/radiol.2462070712.

[130] Collins Jannette and Stern Eric J, "Alveolar Lung Disease - Chest Radiology: The Essentials, 2nd Edition," Feb. 06, 2022. https://doctorlib.info/medical/chest/4.html (accessed Feb. 06, 2022).

[131] P. P. Teixeira e Silva Torres1, M. Fouad Rabahi2, M. A. do Carmo Moreira2, D. Luiz Escuissato3, G. de Souza Portes Meirelles4, and E. Marchiori5, "Importance of chest HRCT in the diagnostic evaluation of fibrosing interstitial lung diseases," *J. Bras. Pneumol.*, p. e20200096, Jun. 2021, doi: 10.36416/1806-3756/e20200096.

[132] G. Dalpiaz and A. Cancellieri, "Alveolar Pattern," *Atlas Diffuse Lung Dis.*, pp. 145–162, Dec. 2016, doi: 10.1007/978-3-319-42752-2_9.

[133] C. A. Ridge, A. A. Bankier, and R. L. Eisenberg, "Mosaic attenuation," *AJR Am. J. Roentgenol.*, vol. 197, no. 6, Art. no. 6, Dec. 2011, doi: 10.2214/AJR.11.7067.

[134] F. Gaillard, "Head cheese sign (lungs) | Radiology Reference Article | Radiopaedia.org," *Radiopaedia*, Feb. 12, 2022. https://radiopaedia.org/articles/head-cheese-sign-lungs (accessed Feb. 12, 2022).

[135] S. E. Rossi, J. J. Erasmus, M. Volpacchio, T. Franquet, T. Castiglioni, and H. P. McAdams, "'Crazy-Paving' Pattern at Thin-Section CT of the Lungs: Radiologic-Pathologic Overview," *RadioGraphics*, vol. 23, no. 6, Art. no. 6, Nov. 2003, doi: 10.1148/rg.236035101.

[136] N. Gupta, R. Vassallo, K. A. Wikenheiser-Brokamp, and F. X. McCormack, "Diffuse Cystic Lung Disease. Part II," *Am. J. Respir. Crit. Care Med.*, vol. 192, no. 1, Art. no. 1, Jul. 2015, doi: 10.1164/rccm.201411-2096CI.

[137] S. R. Desai, H. Prosch, and J. R. Galvin, "Plain Film and HRCT Diagnosis of Interstitial Lung Disease," in *Diseases of the Chest, Breast, Heart and Vessels 2019-2022: Diagnostic and Interventional Imaging*, J. Hodler, R. A. Kubik-Huch, and G. K. von Schulthess, Eds. Cham (CH): Springer, 2019. Accessed: Feb. 06, 2022. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK553872/

[138] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Adv. Phys.*, vol. 56, no. 1, pp. 167–242, Jan. 2007, doi: 10.1080/00018730601170527.

[139] A. A. Trușculescu *et al.*, "Enhancing Imagistic Interstitial Lung Disease Diagnosis by Using Complex Networks.," *Med. Kaunas Lith.*, vol. 58, no. 9, Sep. 2022, doi: 10.3390/medicina58091288.

[140] A. M. Nambiar, C. M. Walker, and J. A. Sparks, “Monitoring and management of fibrosing interstitial lung diseases: a narrative review for practicing clinicians,” *Ther. Adv. Respir. Dis.*, vol. 15, p. 17534666211039772, Sep. 2021, doi: 10.1177/17534666211039771.