Paper ML02

Sub-population Detection Using Graph-based Machine Learning

Sergey Glushakov, Intego Group, Maitland, FL, USA
Iryna Kotenko, Intego Group, Kharkiv, UKRAINE
Kostiantyn Drach, Jacobs University Bremen / Intego Group, Bremen, GERMANY

ABSTRACT

Exploratory analysis is used for investigating clinical data by building statistical models, defining end-points, and determining significant covariates. The expected outcome is the identification of a sub-population of patients most responsive to treatment under the study. A graph-based approach to visualize complex relationships in clinical datasets can be an effective solution for sub-population detection. In this approach each node on the graph corresponds to a single patient, while similar patients are connected with an edge. As datasets may include a large number of participants, visual exploration on the graph may be challenging or even misleading. This paper describes machine-learning algorithms applied to automatic detection of sub-populations of similar patients using a graph-based community search. The computational experiment was performed on a clinical study with 1,041 participants. A novel approach to Topological Data Analysis was used to extract graphs from the dataset to further perform a community search using several algorithms.

INTRODUCTION

The variety of important problems in clinical trials can be represented and studied using graphs. In this paper we rely on graphs as a fundamental approach to structuring and analyzing data in a clinical study. We explore the tendencies of nodes in a graph to form highly interlinked communities that can lead to the discovery of useful information. Community detection relies on detecting subgroups of densely connected nodes, with many edges connecting nodes of the same community and comparatively few edges connecting nodes of different communities. Such communities can be considered as fairly independent areas of a graph and help identify and exploit relevant relationships in the dataset.

In clinical trials, exploratory analysis is used for investigating clinical data by building statistical models, defining end-points, and determining significant covariates that describe sub-populations of the dataset. The expected outcome of the analysis is the identification of a sub-population of patients most responsive to treatment under the study. In this paper we introduce a graph-based approach to visualize complex relationships in clinical datasets which, combined with sophisticated algorithms used for a community search, can become an effective solution for sub-population detection.

Topological Data Analysis is a novel approach to building a visual representation of a clinical dataset. This analysis allows the extraction of comprehensive graphs from the dataset to provide a compressed graphical representation of a multidimensional set of interrelated clinical outcomes. In practice, this graph consists of nodes corresponding to patients participating in the clinical study and edges connecting patients that share similarities.

Following graph extraction from the clinical dataset, the researcher then undertakes a visual exploration with the purpose of discovering sub-populations within the data. For example, isolated components of the graph or highly interlinked groups of nodes that form communities may indicate meaningful relationships in the dataset. As datasets may include a large number of participants, visual inspection and further discovery of sub-populations on the graph may be challenging or even misleading. This paper describes novel machine-learning algorithms applied to the automatic detection of sub-populations using a graph-based community search, such as the Girvan-Newman algorithm, network modularity, random-walk algorithm, and the clique percolation method. We give an emphasis on the clique percolation method, and on our modification to it, as the most efficient in our applications to analyzing of clinical data.

The computational experiment was conducted for the dataset obtained from a clinical study publicly available for the educational and research purposes by the Childhood Asthma Management Program (CAMP), a clinical trial carried out in children having asthma with the total of 1,041 participants. With such a large number of participants, visual exploration of the graph in which every node corresponds to a single participant can be very challenging and may even lead to a wrong interpretation of the formation of sub-populations. Thus, modern graph-based machine learning algorithms were used for large-scale community detection to enhance the analysis of the graph built using Topological Data Analysis. Detected communities were then further statistically analyzed using SAS® to find predictors and outcomes responsible for the formation of discovered sub-populations.

1. EXPLORATORY ANALYSIS OF CLINICAL DATA

1.1 OVERVIEW

Exploratory data analysis is a statistical approach to performing the analysis of a dataset to provide a summary of its main features. This analysis often involves visual methods, with the key goal of discovering additional insights beyond formal statistical modeling or hypothesis testing. The pioneer of exploratory data analysis was John W. Tukey [1], who encouraged statisticians to explore data to formulate hypotheses that may lead to new experiments. To be clear, the goal of exploratory analysis is not the substitution of standard statistical analysis that confirms previously developed hypotheses. Quite the opposite, the key goals are:

- Identifying a hypothesis that may lead to the causes of anomalies discovered in a dataset
- Evaluating assumptions of the statistical methods used for the analysis
- Assisting in the selection of statistical methods and techniques

Data visualization is an extremely important part of exploratory analysis. It helps the researcher to understand the what, why and how of the problem to be analyzed. This is actually the first step the researcher usually performs while approaching the problem statement in a new dataset. Exploratory analysis enables:

- The breaking down of the problem statement into smaller pieces, the analysis of which can facilitate a better understanding of the dataset
- The revealing of insights that may assist the researcher in making key decisions
- Utilizing visualization as the key part of the analysis

In clinical trials, exploratory analysis is used to investigate clinical data by building statistical models, defining endpoints, and determining significant covariates that describe the sub-populations of the dataset. The expected outcome of the analysis is the identification of the sub-population of patients who are most responsive to the treatment under study. In this paper, we introduce a graph-based approach to visualize complex relationships in clinical datasets, which, combined with sophisticated algorithms used for a community search, can become an effective solution for sub-population detection.

1.2 CONFIRMATORY VS. EXPLORATORY ANALYSIS

Confirmatory data analysis and exploratory data analysis are two statistical approaches widely used in clinical data research. Confirmatory analysis utilizes traditional statistical tools to confirm/refute a hypothesis generated by the researcher. The hypothesis is usually generated as the goal of the clinical study and is formulated while developing a research protocol.

On the other hand, the results of exploratory analysis may be used for generating new hypotheses. Further these hypotheses will be confirmed or refuted by using standard statistical methods. Exploratory analysis is used to investigate data and discover valuable information, such as:

- Hidden data patterns
- Dependencies
- Anomalies and other features

In contrast, confirmatory analysis usually provides predetermined approaches for proving the hypotheses depending on which data types are used, and it also recommends methods for comparing groups of data. Meanwhile, exploratory analysis is often directed towards the following:

- Exploring data
- Looking at data from different perspectives
- Determining dependencies
- Understanding how data may behave
- Summarizing the main characteristics of the data
- Determining predictors that may have an influence on the outcome
- Possibly generating hypotheses

Exploratory analysis may be directed towards understanding what to make of the data, how to present and manipulate the data, and deciding which questions to ask and which areas to explore in the course of the analysis. In exploratory analysis, the researcher may focus his/her efforts on determining the structure of the data, dealing with missing data and anomalies, finding patterns, determining significant parameters, making assumptions and checking them, selecting a model and generating and checking a hypothesis in relation to the selected model, and selecting the most applicable model in addition to a number of other exploration techniques.

1.3 MAIN STEPS OF THE ANALYSIS

The exploratory analysis of clinical datasets may include different steps; however, to summarize the key concepts, all of the approaches involve the following steps (see Figure 1):

1) Data collection
2) Data cleaning
3) Data pre-processing
4) Models and algorithms
5) Data visualization
6) New hypothesis generation
7) Results confirmation

During the data collection step, data are gathered in a predetermined systematic way, e.g., via patient charts, doctor observation notes, studies by scientific institutions, clinical data, and so forth. Then, the data cleaning step includes identifying errors in the data, correcting data, handling missing values, checking data types, and other data cleaning approaches to ensure that the data are useable. Finally, the pre-processing of the data may be performed by transforming the collected and cleaned data into a predetermined format.

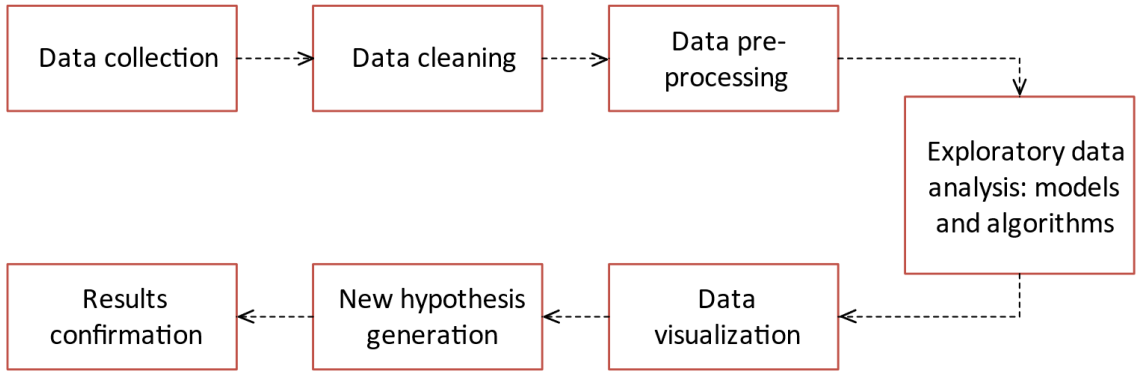


Figure 1. Key steps of exploratory data analysis

As a result, the dataset obtained in the previous steps is then processed to models and algorithms, which may utilize a number of tools and techniques to transform the data in the way the researcher requires. The most popular techniques are graphical techniques, dimensionality reduction, and quantitative techniques. In this paper, we use one of the most advanced techniques, which is based on a number of algorithms from a new and fast-evolving field of mathematics – Topological Data Analysis (see Section 2).

The result of the previous steps brings the researcher to the data visualization. This is an extremely important element in exploratory data analysis because it helps to link findings and form conclusions based on the results of all the previous steps of the analysis. This then leads the researcher to the generation of a new hypothesis based on discovered insights that were not available at the beginning of the analysis. The final step requires the confirmation of the discovered conclusions by performing statistical analysis using standard tools and methods.

1.4 POPULAR TECHNIQUES

The tools useful for exploratory analysis include, among others, graphical techniques, dimensionality reduction, and quantitative techniques. Graphical techniques may include building histograms, box plots, run-sequence plots, Pareto charts, scatter graphs, and a number of other types of diagrams. Graphical techniques also include the application of projection methods, such as grand tour, guided tour and manual tour, and creation of interactive versions of diagrams.

The dimensionality reduction is usually based on such techniques as multidimensional scaling, multilinear principal component analysis, and principal component analysis. The applicable quantitative techniques may include such procedures as median polish, ordination, and trimean. In the classical approach, exploratory analysis often includes the creation of regression models (linear, non-linear, multiple, etc.).

The development of machine learning has resulted in the widespread use of cluster analysis. Clustering is grouping objects (e.g., individuals) into subgroups (or clusters) so that the objects in the same cluster have more features in common with each other than with those in other clusters. For example, a researcher may use cluster analysis to analyze the diagnostic questionnaires of multiple patients and identify subgroups of patients who have similar symptoms. Then, the researcher may investigate the identified subgroups of patients individually using other statistical approaches to find those common characteristics of the patients that may be associated with the symptoms.

Typical clustering approaches include, for example, k-means clustering, hierarchical clustering, and graph-based clustering. In k-means clustering, the data are divided into clusters (subgroups) based on the distances between each data point and the center location of each cluster. Graph-based clustering includes inspecting the data represented in the form of a graph to identify a subset of nodes in which every two nodes are connected by an edge and considering the identified subset of nodes as a cluster.

Hierarchical clustering is based on the concept of proximity and builds a model based on distance connectivity by grouping objects into clusters based on their distance. Determining a distance function (also referred to as a distance) is one of the techniques used to determine the similarity of objects. The distance function is a tool that is used to measure the similarity between data points that belong to an outcomes dataset. Upon computing a distance for a dataset, the dataset may be represented in the form of a metric space that can be studied using geometric and topological methods. In addition to choosing the distance function, the researcher needs to define the notion of what constitutes a cluster and how to identify that cluster. The clustering algorithms for a dataset need to be chosen experimentally depending on the type of data.

In general, the cluster analysis includes exploring data to find one or more parameters that may be considered as factors that influence the outcomes. Further steps include building a statistical model based on these parameters and evaluating a quality score of the statistical model to determine how 'truly' the statistical model describes the data. A significance level is then determined for the parameters found in the data to understand the level of influence of these parameters on the statistical model. Upon determining the significant parameters, also referred to as significant covariates, the researcher may consider the revealed significant covariates when developing recommendations on performing a clinical study. For example, the researcher may recommend limiting the input population based on age if age has been determined to be a significant covariate.

Although exploratory analysis is widely used in clinical trials, the researcher may face a number of difficulties when performing an exploratory analysis. Firstly, it may be difficult to adequately determine which statistical model is more suitable for a specific study. For this reason, in some cases the researcher may need to select several statistical models and evaluate a quality score for each of them to find the statistical model with an acceptable quality score.

Secondly, the statistical models may be resource- and time-consuming. For example, if the statistical model has a large number of observations and a researcher wants to check a large number of predictors to determine which of them may be significant, the amount of time and memory resources required to perform all the calculations based on this model can be extremely large. Although in some cases the researcher may select a sub-population for use in the model, the selected data may appear to be biased and the model built based on this sub-population may not adequately illustrate the original population.

Thirdly, the understanding of how the data that the researcher intends to use for building a statistical model look in an n-dimensional space may give a clue as to which statistical model to select. However, the visualization of these data may not be a trivial task for the researcher.

Fourthly, when building a statistical model in SAS®, the researcher needs to have both a programming and a statistical background to understand how to modulate the data and which type of model is better to use as well as to clearly see whether a component of the model is a random component or a fixed constituent in addition to solving other tasks.

In this paper, we start the analysis with the visualization of the data without the need to first build a model or understand how the data should be described. The visualization of the data allows us to find specific geometrical features of the data and use the geometry of the data to find parameters that may be significant in the model. This analysis may simplify further exploratory analysis for the researcher. In particular, the researcher may consider the geometry of the data when selecting a statistical model that describes the data and when selecting which parameters may be included in the building of the model or which parameters may be definitely omitted due to their low importance to the model.

2. TOPOLOGICAL DATA ANALYSIS

Topological data analysis is a novel approach to building a visual representation of a clinical dataset. This analysis allows the extraction of comprehensive graphs from the dataset to provide a compressed graphical representation of a multidimensional set of interrelated clinical outcomes. In practice, this graph consists of nodes corresponding to patients participating in the clinical study and edges connecting patients that share similarities. In this section, we look closely at the concept of the geometric properties of a dataset to understand how graphs can be extracted from clinical datasets to further use modern machine learning algorithms for the automatic detection of subgroups of related patients while performing exploratory data analysis.

2.1. TOPOLOGY AND DATA MINING

Topology is a field of mathematics that deals with the properties of objects that remain invariant under continuous deformation. Imagine a surface that is made of very thin and elastic material. You can bend, stretch or crumple the surface in any way you like; however, you cannot tear it or glue any parts of it together. As you deform the surface, it will change in many ways, but some properties will remain the same. The idea that underpins topology is that some geometric properties depend not on the exact shape of an object but rather on how its parts are combined.

As a simple example, consider geometric figures on the plane representing the numerical digits 0, 1, 2, …, 9. For a topologist, various representations of the digit 0 are equivalent since they can all be transformed into each other in a continuous way without cutting or gluing (Figure 2 a-d). It is possible to change the size, thickness, or slope of the digit 0 through continuous deformation; however, one property remains invariant: The object separates the plane into two regions, namely the interior and the exterior. At the same time, 0 is not topologically equivalent to 1 or 8: 1 does not encircle a region and 8 contains two holes (Figure 2 e). The topological classification of the digits results in the following five classes:

$$\{0\}, \{1, 2, 3, 5, 7\}, \{4\}, \{6, 9\}, \{8\}.$$

The digits in any of the classes are topologically identical, but no two digits that are taken from distinct classes are equivalent from the topological point of view.

The number of holes in a geometric object is a basic topological property. Another significant property is connectedness. Intuitively, an object is connected if it consists of a single piece. For example, the curve representing 0 is connected; if one removes any two points from it, it will become disconnected. Pieces of a disconnected object that are, themselves, connected are referred to as connected components. In the mathematical study of topology, all of these intuitive concepts are examined on a rigorous basis and are generalized to higher dimensions.

a) b) c) d) e)

Figure 2. Different representations of the digit 0 (a-d) are topologically equivalent.
All of them share a common topological property: They divide the plane into an interior region and an exterior region. The digit 8 (e) is not equivalent to 0 since it encloses two internal regions.

Topology deals with abstract mathematical entities, such as curves and surfaces, that consist of an infinite number of points. In practice, however, all datasets are necessarily finite. Recently, a new field has emerged at the crossroads of topology and data science. Topological data analysis (TDA) aims to extract topological data, i.e., qualitative information, from finite sets of data points. It involves exploring datasets (viewed as finite clouds of points in a multidimensional space) at multiple scales or resolutions, from fine- to coarse-grained. Given a complex dataset, TDA can be used to extrapolate the underlying topology and build a compressed yet comprehensive topological summary of the dataset. TDA exploits a variety of methods and algorithms stemming from computational topology and geometry, statistics, and data mining. For detailed expositions of the mathematical theories that underpin TDA together with some applications in biology, see [2-4] and the references therein.

2.2 ROBUST GEOMETRIC PROPERTIES OF DATASETS

For illustrative purposes, a simple two-dimensional dataset was constructed whereby the data points were arranged in a “zero-like” shape. We applied our proprietary patent-pending TDA algorithm to this dataset to build a graph in which every node corresponds to a single data point.

In order to show the robustness of the topological approach, some data points from the dataset were intentionally omitted at random, and additional graphs were built for the modified datasets whereby 50% and 90% of the original data points were missing (see Figure 2).

The graphs show certain geometrical stability even in the case of 90% missingness. The shape of the graphs built on the remaining data points is structurally similar to the shape of the graph corresponding to the complete dataset. Therefore, in this example, graphs representing a relatively small portion of the data still have a similar shape to the graph representing a complete dataset.

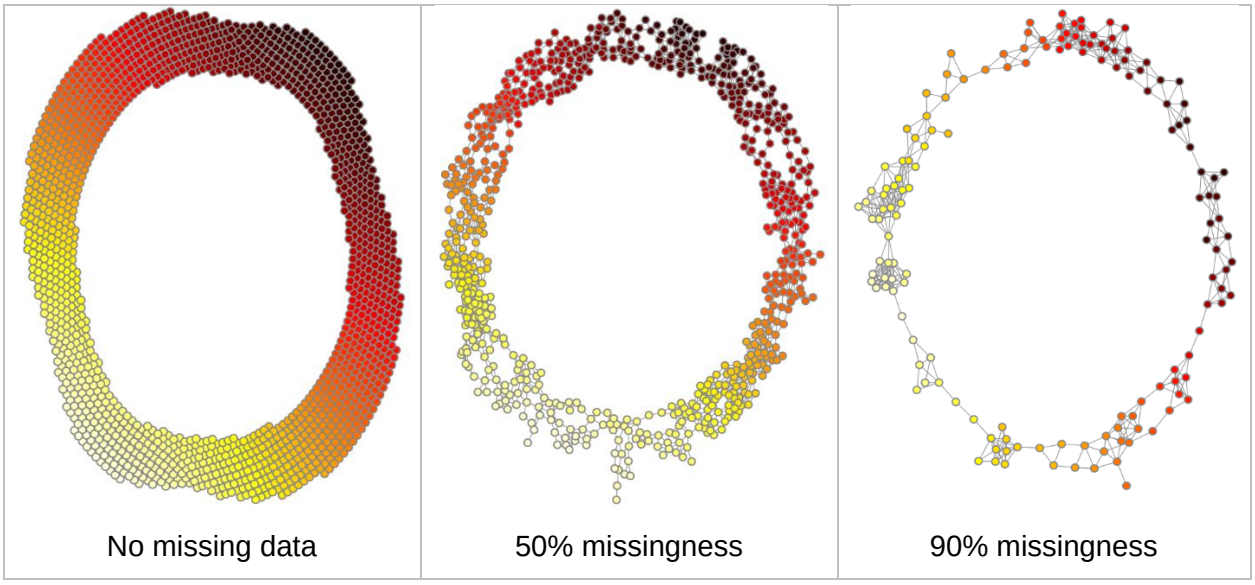


Figure 3. Graphs representing a dataset with varying proportions of randomly missing data.
Graphs produced by the TDA algorithm for a complete dataset (left panel) and datasets where 50% and 90% of the data points are missing at random (middle and right panels, respectively). This example illustrates that even with 90% of the data missing, the cyclic shape of the dataset is preserved in the corresponding graph.

2.3 UNDERSTANDING COMPLEX DATA USING TOPOLOGY

Topology was originally developed to distinguish between the qualitative properties of geometric objects. It can be used in conjunction with the usual data-analytic tools for the following tasks:

1) **Characterization and classification.** Topological features succinctly express qualitative characteristics. In particular, the number of connected components of an object is of importance for classification.
2) **Integration and simplification.** Topology is focused on global properties. From the topological perspective, a straight line and a circle are locally indistinguishable; however, they are not equivalent if they are considered as a whole. Topology offers a toolbox by which local information about an object can be integrated into a global summary. Thus, topology can provide the researcher with a natural "big-picture" view of complex, multidimensional data.
3) **Features extraction.** Topological properties are stable. The number of components or holes is likely to persist under small perturbations or measurement errors. This is essential in data mining applications because real data are always noisy.

In the context of clinical research, the dataset under study is typically a table of outcomes in a particular clinical trial. The table rows correspond to the individual participants in the clinical trial, and the columns contain information on specific outcome measures of interest, such as lab tests, vitals, questionnaires, etc. Given a table of clinical outcomes, two types of parameters are required to generate a graph using TDA. The first of these is a projection, a function that is used to stratify patients into subpopulations. The second is a distance function that measures the proximity between patients. The distance function makes it possible to split each subpopulation into clusters of related patients with similar outcomes.

To be considered for further analysis, a graph extracted from the dataset using TDA algorithms should meet certain requirements. Namely, it should:

- accurately represent the original dataset;
- eliminate the features of the dataset that are not relevant to the purpose of the study;
- reduce the complexity of the features that are shown on the data map;
- be insensitive to small noise, such as errors of measurement.

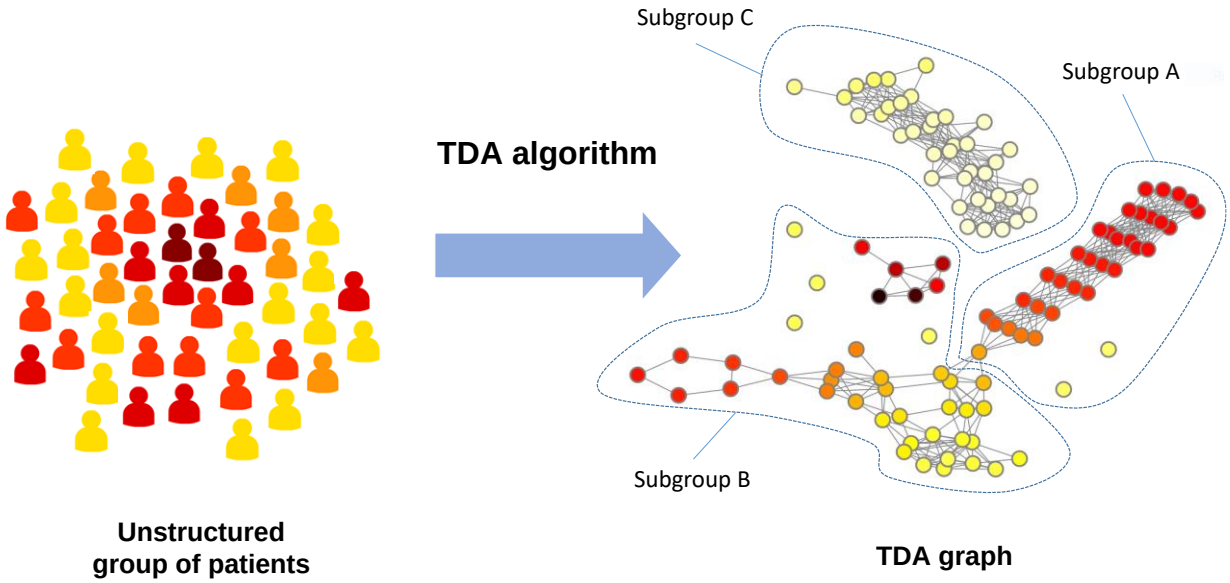


Figure 4. Discovery of multivariate patterns in clinical trial outcomes.
A graph represents groups of patients structured according to the similarity of clinical outcomes.

The core idea of data mining using TDA relies on the visual discovery of subgroups of related patients in a graph (see Figure 4) that retains the relevant information about the dataset in a compact and efficient manner. For the clinical dataset, the following criteria have to be met to perform the analysis:

- **Each node represents a patient** – a graph extracted from a clinical dataset is actually a graphical representation of the dataset in which each node represents an individual.
- **Similar nodes are connected** – two nodes representing similar patients (in terms of a predefined set of clinical outcomes) are connected with an edge.
- **Coloring focused on specific outcomes** – the color of the nodes helps to highlight emerging patterns in the data and to identify subgroups of patients related to the distribution of a variable of interest.
- **Visual discovery of subgroups** – clusters or "communities" of nodes on a graph reflect a segmentation of patients that may indicate robust patterns within the data.

To construct a visual representation of clinical trial data, the dataset in CDISC format is pre-processed using specific algorithms that deal with data-specific issues, such as proper scaling of numerical variables, conversion of categorical variables, and others. At the initial stage, a primary dataset needs to be determined whereby each row in a data table corresponds to a unique patient or volunteer who participated in the clinical study, while the columns represent either observational variables (outcomes), such as safety and efficacy biomarkers, or predictors, such as demographic attributes, medical history, interventions, etc. The resulting dataset is further processed by a TDA algorithm to construct a visualization of the observational variables represented by a graph.

TDA can deal with a variety of numerical and categorical outcomes:

- **Interrelated biomarkers** – e.g. patients' vital signs or basic metabolic panel results on a specific day of study.
- **Series of repeated measurements** – e.g. weekly hemoglobin levels during chemotherapy in oncological patients.
- **Questionnaire data** – binary, nominal or ordinal responses to the items of a questionnaire, aggregate scores, etc.

After a graph is constructed based on the selected outcomes, the researcher then visually explores it for the purpose of discovering interesting subgroups within the data. For example, the isolated components of a data map or highly interlinked groups of nodes that form communities may indicate meaningful relationships within the dataset.

2.4. PREDICTORS AND OUTCOMES

A very common situation in statistics occurs when the distribution of an outcome (or response variable) is related to one or several predictors (or explanatory variables). A standard approach through which researchers study the relationship between the predictor and the outcome is the application of a suitable statistical model. The model selection depends on the data types of the predictor and outcome (quantitative, binary, categorical, etc.) and often involves additional assumptions concerning the distribution of the outcome. For example, linear regression is often used when both the predictor and the outcome are quantitative (e.g., BMI and blood pressure); Fisher's exact test or the χ^2 test can be applied when both variables are binary or categorical (e.g., gender and ECOG score); and logistic regression can be a suitable model for evaluating the relationship between a quantitative predictor and a binary outcome. The application of such approaches can be problematic in the context of complex settings that have multivariate outcomes, i.e., when many related outcomes are recorded for the same individuals.

TDA is naturally designed to assist researchers in dealing with multivariate heterogeneous outcomes in such a manner that it is possible to study several related outcomes of different types (quantitative, ordinal, categorical) together. An incomplete list of multivariate outcomes includes a series of repeated evaluations of a given response variable over time; simultaneous evaluations of different, but potentially correlated, biomarkers (e.g., levels of serum creatinine, blood urea nitrogen, and neutrophil gelatinase-associated lipocalin as a means of evaluating kidney function); and questionnaire data to assess patient's general health or quality of life, etc.

TDA takes a panel of personalized outcomes of a clinical trial as its input. More specifically, the outcomes table is a synthetic dataset that consists of row vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, with each vector corresponding to a single

participant. Here, x_i denotes the i-th outcome reading for the participant labeled $\boldsymbol{x}$. Outcomes are either calculated or directly extracted from the original "raw" datasets that were collected during the course of the clinical trial and are stored in the CDISC SDTM or ADaM format.

From the clinical research perspective, an outcome is an evaluation of some aspect of a participant's health that results in a recorded datum. There is more than one way of classifying clinical trial outcomes (see Table 1). Depending on the research goal, it is useful to differentiate between outcomes linked to biomarkers and clinical outcome assessments (COA) (see [5]). A biomarker is a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes or pharmacologic responses to a therapeutic intervention [6]. A COA is any assessment that may be influenced by human choices, judgment or motivation, and it may provide either direct or indirect evidence of the benefits associated with a given treatment. In contrast to biomarkers, which are determined using automated processes or algorithms, COAs depend on a participant's or clinician's implementation, interpretation, and reporting of the data.

Table 1. Classifications of clinical trial outcomes

Clinical Trial Goal	Specialty	CDISC Domain	Data Type	Variable Type
Safety Efficacy Effectiveness Quality of life	Allergy/immunology Cardiology Endocrinology Gastroenterology Hematology/oncology …	AE EG LB QS VS …	Cross-sectional Longitudinal Aggregate	Quantitative Categorical Ordinal Interval

It is important to note that specific research objectives require customized configurations of outcomes panels. In this paper, we consider several different outcomes panels derived from the same clinical trial dataset to study various aspects of the disease.

3. COMMUNITY SEARCH ALGORITHMS

3.1 INTRODUCTION TO THE PROBLEM

The variety of significant problems in clinical trials can be represented and studied using graphs. In this paper, we rely on graphs as a fundamental approach to structure and analyze data in a clinical study. We explore the tendencies of nodes in a graph to form highly interlinked communities that can lead to the discovery of useful information. Community detection relies on detecting subgroups of densely connected nodes, with many edges connecting nodes of the same community and comparatively few edges connecting nodes of different communities. Such communities can be considered to represent relatively independent areas of a graph and help identify and exploit relevant relationships in the dataset.

By constructing a graph that represents the clinical dataset, the researcher can undertake a visual exploration with the purpose of discovering sub-populations within the data. For example, isolated components of the graph or highly interlinked groups of nodes may indicate meaningful relationships in the dataset. As datasets can span large study populations, visual inspection, and further discovery of sub-populations within the graph can be challenging or even misleading. In this section, we describe some known machine-learning algorithms applied to the automatic detection of sub-populations using a graph-based community search (the Girvan-Newman algorithm, network modularity, random-walk algorithm, and the clique percolation method). We give an emphasis on the clique percolation method, and on our modification to it, as the most efficient in our applications to analyzing of clinical data.

3.2 BASIC CONCEPTS

The modern approach to data science frequently employs graphs to enhance understanding of complex systems. The key feature of a graph is a community structure, which relates to the way the nodes are organized in communities. Specifically, many edges connect nodes within the same community (or cluster), while comparably few edges connect nodes between different communities [7]. These clusters or communities can be considered to represent independent structures within the graph, and the detection of those independent communities is one of the key goals in the analysis of large graphs that represent complex relationships within datasets.

Graphs can be analyzed using global, local, and intermediate-scale approaches. The identification of intermediate-scale structures within the graph enables the discovery of features that cannot be identified at either the local level of vertices (or nodes) or the global level of general graph statistics [8].

In graphs that represent real-world systems, the distribution of edges over subgroups of vertices is usually non-uniform. This reflects possible presence of some hidden structure and patterns in the graph, and hence in the real-world data from which the graph was created. Specifically, some groups of vertices may have high concentrations of edges, while the concentrations of edges between these groups of vertices may be low. This structure takes the form of an intermediate-scale graph structure known as a community structure [9], or a cluster structure, where a group of densely connected vertices is referred to as a community. Figure 5 illustrates an example of a community structure within a graph that contains three clusters of vertices with dense internal connections and comparably fewer connections between clusters.

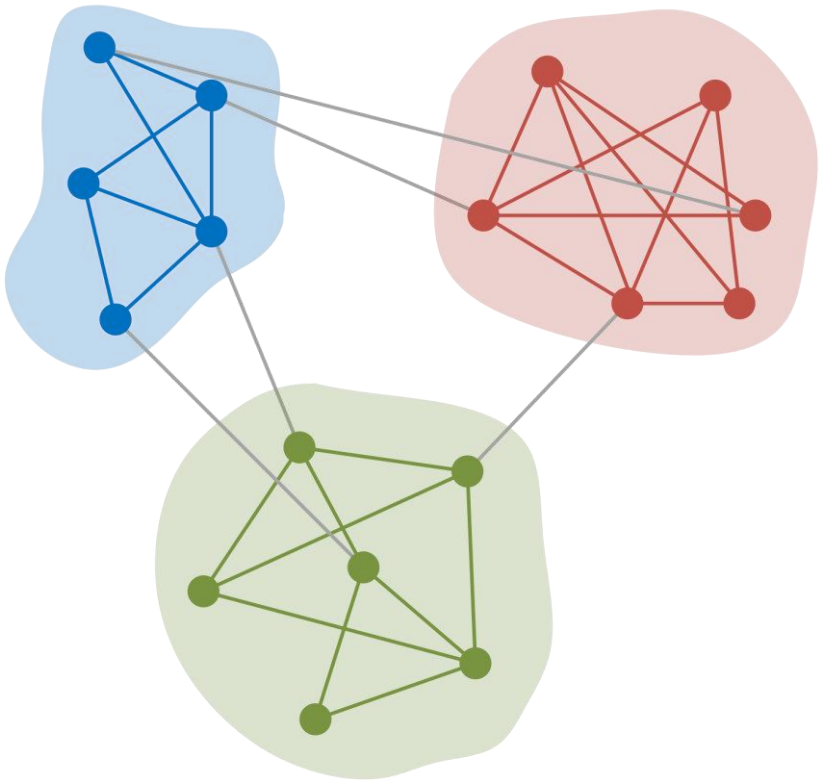


Figure 5. A schematic representation of a simple graph that has a community structure.
The graph contains three communities of densely connected vertices that have a much lower density of connections (gray edges) between them.

Communities, or clusters, are groups of vertices within a graph that are likely to share common properties and/or play similar roles within the graph. In view of this, where possible, the aim of community detection is to identify communities within the graph and their hierarchical organization by using the information that is contained within the graph topology alone [7]. Identifying communities according to the topological properties of the graph only allows classification of nodes according to their structural position on the graph. Thus, nodes with a central position in their communities share the largest number of edges with the other vertices within the community, which may indicate the important role they play in the stability of the community. On the other hand, vertices that are located at the boundaries between communities may play an important role as mediators leading the relationships and exchange between different communities.

The problem of graph clustering, intuitive at first sight, is actually not well defined. Though numerous attempts have been made to analyze real-world systems based on the community structure in multiple disciplines and practical applications, graph theory does not define the problem of graph clustering and no universally accepted definitions for a community or partitioning into communities have arisen. Therefore, the concepts of a community and partitioning into communities require some extent of arbitrariness from a researcher based on the specific problem under consideration [7].

Detecting communities within a graph (especially large ones) can be computationally difficult if the number of communities within the graph is unknown and the size and density of the communities are unequal. However, several algorithms have been developed and used for community search with varying degrees of success in recent years. The following sections will review some of the most notable algorithms that have been proposed for community search.

3.3 GIRVAN-NEWMAN ALGORITHM

The Girvan-Newman algorithm [9] attempts to identify the edges that are located “between” some pairs of vertices in the graph. In the algorithm, the distance between all pairs of vertices is calculated; i.e., the shortest edge-based path. Such paths define the edge betweenness characteristic of edges. The edge betweenness characteristic of an edge is the number of shortest paths between pairs of vertices that run along the edge.

The method of community detection using the Girvan-Newman algorithm is based on calculating the edge betweenness characteristic for all edges in the graph. The method includes steps of removing the edge having the highest edge betweenness and recalculating the edge betweenness for all edges affected by the removal. The steps are repeated until no edges remain. The edges that have the highest edge betweenness characteristic are the most “loaded” and, hence, are considered to lie the most “between” communities. The removal of the revealed edges from the graph results in the vertices falling into communities. The removal of edges that have the further highest values of edge betweenness characteristic separates further communities within the graph.

Let’s briefly review the main steps involved in the algorithm (see Figure 6):

- STEP 1. Calculate the betweenness of all existing edges in the graph
- STEP 2. Remove the edge with the highest betweenness
- STEP 3. Recalculate the betweenness of all edges affected by the removal of the edge with the highest betweenness
- STEP 4. Repeat STEP 2 and STEP 3 until no edges remain

The Girvan-Newman algorithm has been widely applied to a variety of graphs, e.g., graphs of human and animal social networks, metabolic graphs, gene graphs, graphs representing collaborations between scientists and musicians, and so forth. However, this algorithm is computationally intensive and takes $O(m^2n)$ times on a graph with m edges and n vertices. In view of the large amount of time required to perform the calculations, the use of the algorithm is limited to graphs that contain less than a few thousand vertices. Furthermore, the algorithm does not show how many edges need to be removed for the most optimal community detection [9].

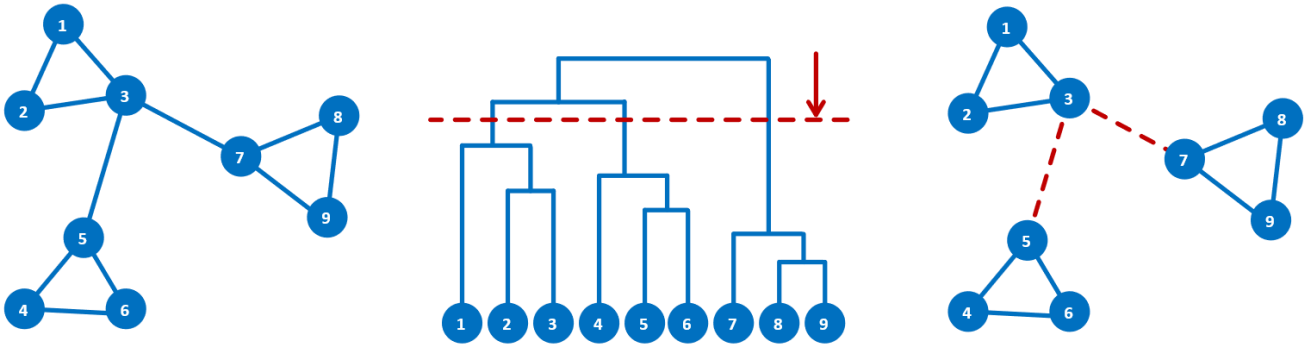


Figure 6. A hierarchical decomposition of the graph.
As we move down the dendrogram, we see the partitioning of communities.

3.4 MODULARITY-BASED ALGORITHMS

The modularity is a function that measures the quality of the community partitions within a graph. It measures the strength of the division of the graph into clusters or communities. Upon clustering the graph into communities, we can use the modularity score to assess the quality of the clustering performed (see Figure 7).

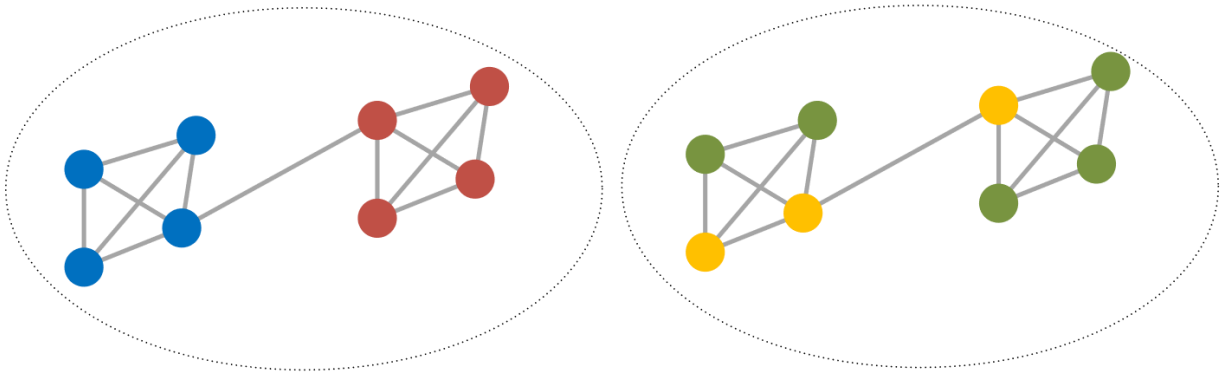


Figure 7. Modularity can be used to measure the 'quality' of a partition within the graph

The modularity shows that the partitioning into communities is "good" when there are many edges within the communities and few edges between the communities. This is based on the assumption that high modularity values correspond to "good" partitions.

For example, let's look at one of the approaches to the modularity-based algorithm. According to this algorithm, the graph is first partitioned into a finite number of arbitrary partitions. Then, the modularity of each arbitrary partition is calculated. The partitioning that exhibits the highest value of modularity is deemed to be of the highest quality and, hence, is selected as the most optimal partitioning.

Although the modularity-based algorithm is widely used, it is impossible to exhaustively optimize the value of modularity in view of the large number of ways in which a graph may be partitioned. Some algorithms, such as greedy optimization, simulated annealing, and extremal optimization, provide good approximations of maximum values of modularity in a reasonable time [10]. However, in most cases, modularity-based algorithms have a "resolution" preference, which means they tend to prefer clusters of a particular size for a given graph. Therefore, they suffer a resolution limit and cannot be effectively used to detect small communities.

3.5 RANDOM WALK ALGORITHM

The random walk algorithm provides random paths between vertices in a graph (see Figure 8). It operates on the assumption that a random walker is placed on an arbitrary vertex in the graph and starts walking randomly from one vertex to another. According to the random walk algorithm, if the graph has a community structure, many of the random walker's paths will be on edges within the community due to the high density of edges. On the contrary, a walker will have fewer walks on edges lying between communities. The distance between vertices is defined by using some information about the paths of a random walker. Based on the distances, vertices that are located close to each other and, hence, form a community, are determined

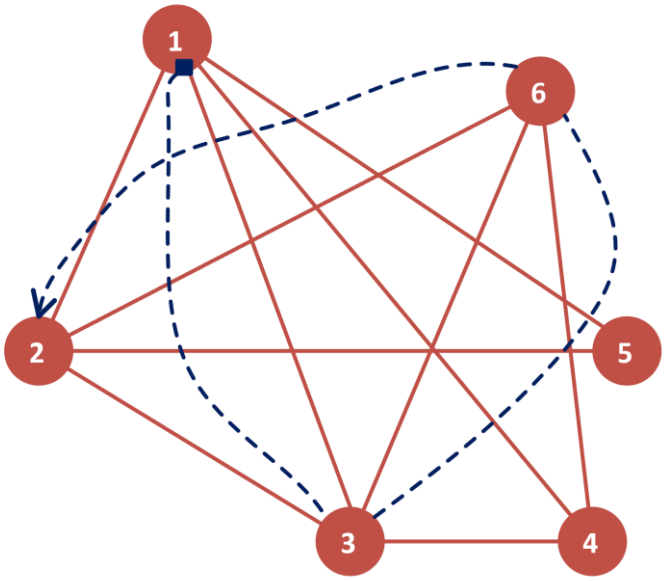


Figure 8. Random walk is an algorithm that provides random paths in a graph.
We start at one node, choose a neighbor to navigate to at random and then repeat the process keeping the resulting path in a list.

Random walks can also be useful to find communities. If a graph has a strong community structure, a random walker spends a long time inside a community due to the high density of internal edges and the consequent number of paths that could be followed. Various guises of the random walk algorithm are described in numerous publications (see [11], [12], [13]). However, calculations of distances between the paths the random walker walks from each vertex of the graph may demand huge computing resources. To avoid extensive calculations, the number of vertices the random walker needs to walk from each vertex is usually limited to a preselected value. In view of this, the random walk algorithm may result in the non-optimal detection of communities.

3.6 COMMUNITY DETECTION IN R AND PYTHON

The open-source package *igraph* is available (https://igraph.org/) in both R and Python. This package was specifically designed for the purposes of creating and manipulating graphs and analyzing networks. The key advantage of *igraph* is that it can effectively handle large networks.

This package incorporates functions for finding communities in graphs and predominantly employs methods that are based on the algorithms described above, including the *cluster_edge_betweenness* function used within the Girvan-Newman algorithm, the *cluster_fast_greedy* function employed within the modularity-based algorithm, and the *cluster_walktrap* function that is employed in the random walk algorithm.

Furthermore, the *Networkx* package is available in Python (https://networkx.github.io/). This package takes the form of a library of functions that can be used to study graphs and networks and specifically target large real-world graphs with millions of nodes and edges. *Networkx* incorporates some functions that are lacking in *igraph*. In particular, *Networkx* uses the clique percolation method, which is used for solving the problem stated in this paper, as described in detail below.

3.7 PERCOLATION THEORY – CLIQUE PERCOLATION METHOD

The community detection methods described above are beneficial for finding communities in graphs that have non-overlapping communities, i.e., communities where a vertex belongs only to one community but not to two or more communities. However, most of the graphs that represent real-world systems incorporate overlapping or nested communities.

One of the most popular approaches for finding overlapping communities is the Clique Percolation Method [14]. This method operates on the assumption that the internal edges within a community form *k*-cliques (in view of the high density of the edges), and the edges that lie between the communities are not likely to form cliques [7,14]. In graph theory, a *k*-clique is a subgraph of the original graph isomorphic to a complete graph with *k* vertices (see Figure 9). A complete graph is a graph in which every pair of vertices is connected by a unique edge.

The use of this method is based on the assumption that if a clique can “move” in the graph, it will get trapped inside the community and will not manage to pass in between two communities due to a lack of connecting paths. In this method, a community is defined as a maximal connected subgraph of the original graph so that each vertex in this graph belongs to some *k*-clique which lies entirely in the subgraph. The classical Clique Percolation Method receives a value of *k* as an input and produces the list of all possible communities (as described above for the given value of *k*) as an output.

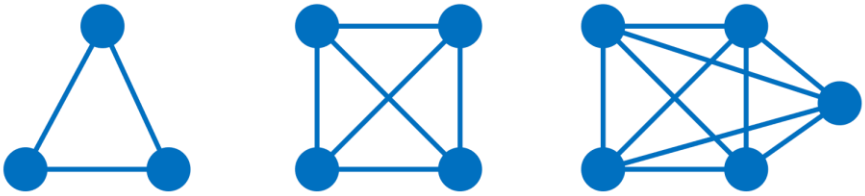
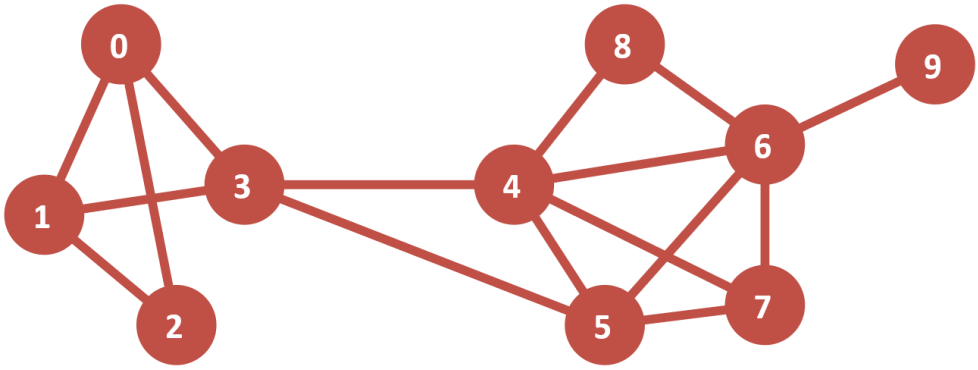
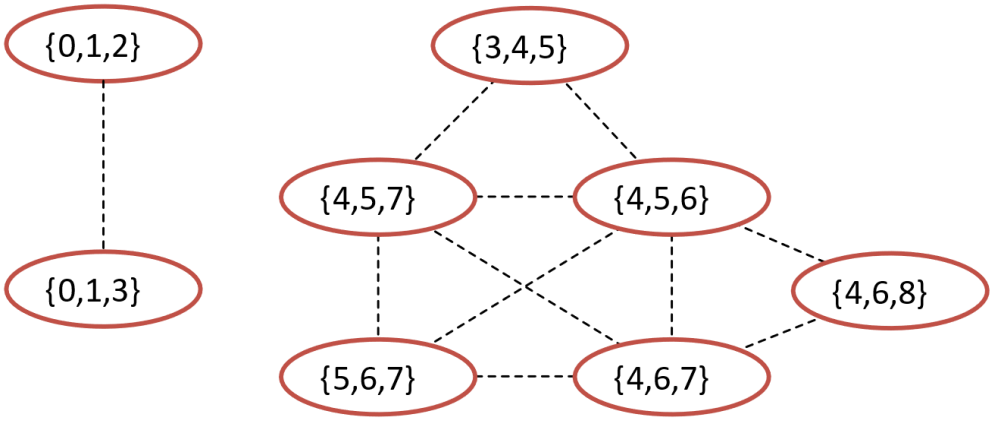


Figure 9. *k*-clique is a complete graph with *k* vertices

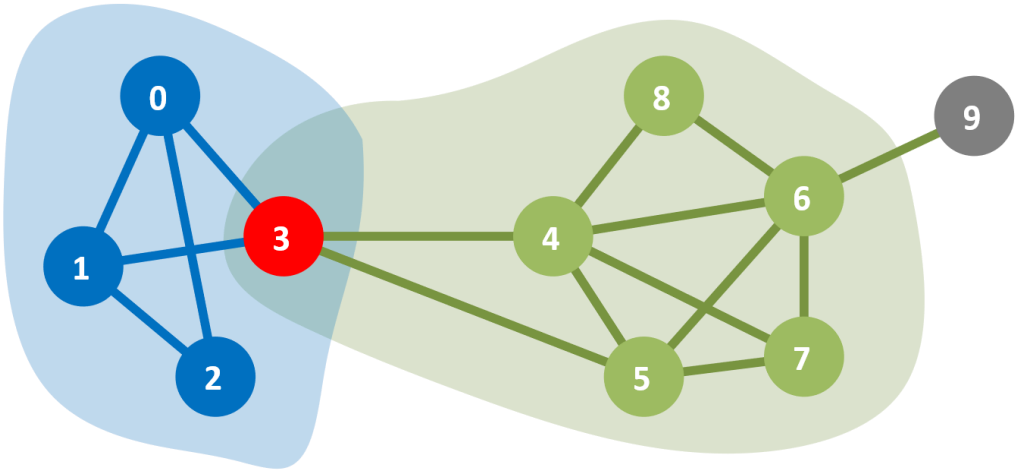
A particular property of the Clique Percolation Method is that some vertices may belong to several communities as several *k*-cliques may pass through these vertices, and some vertices may not belong to any community as they cannot be reached by any *k*-clique (see Figure 10). Furthermore, this algorithm is computationally intensive because the detection of maximal cliques requires processing time that runs exponentially to the size of the graph.



a) Original graph



b) Clique-graph built for *k* = 3



c) Illustration of *3*-clique communities

Figure 10. Example of overlapping community detection by Clique Percolation Method on a simple graph

However, as shown by the practical applications of this algorithm to real-world systems, this method works reasonably fast due to a limited number of cliques in real-world-based graphs. For example, the analysis of graphs that incorporate 105 vertices can be achieved in a reasonably short period of time [7].

Figure 10 presents an example of the Clique Percolation Method. Where a simple graph is in place with nodes numbered from 0 to 9 (see Figure 10 a), the algorithm first extracts all cliques of size *k*. In this example, *k* = *3*. This entails that all combinations of nodes forming *3*-cliques should be extracted from the original graph:

Cliques for *k* =*3* :
{0,1,2}, {0,1,3}, {3,4,5}, {4,5,7}, {4,5,6}, {5,6,7}, {4,6,7}, {4,6,8}

Using these combinations of *3*-cliques, the algorithm then builds a clique graph in which every node is represented by a *3*-clique from the list and two nodes are connected by the edge if they share two common nodes of the original graph (see Figure 10 b). The connected groups on the clique graph represent communities, while actual members of the community are obtained by extracting the nodes of the individual cliques that form the connected group:

***3*-clique communities:**
Blue Community: {0,1,2,3}
Green Community: {3,4,5,6,7,8}

In this example, the algorithm discovered two overlapping communities (see Figure 10 c), which share a common node (node 3). Here, node 9 is excluded from any of the communities because, with *k* = *3*, it does not belong to any of the two discovered communities. In other words, node 9 is connected by just one edge with node 6; as such, it cannot be reached by any of *3*-cliques.

For the purpose of the experiment performed specifically for this paper, we customized the Clique Percolation Method. Among many areas of improvement we worked on, the most important one was finding the optimal value of *k* while taking into account the specifics of the dataset that was the subject of the analysis. Another extremely important aspect of the algorithm modification related to the productivity optimization in light of the significant amount of computing power required to make community search algorithms work.

4. COMPUTATIONAL EXPERIMENT

4.1 CLINICAL STUDY OVERVIEW

The proprietary algorithm for topological data analysis (TDA) was applied to the publicly available dataset provided by the Childhood Asthma Management Program (CAMP) for educational purposes following a clinical trial involving children with asthma. The aim of the CAMP study was to investigate the long-term effects of three treatments: budesonide, nedocromil, and a placebo, on a pulmonary function as measured by normalized forced expiratory volume (FEV) over a 5-6.5 year period [15]. In this multicenter, masked, placebo-controlled, randomized trial, 1,041 participants were randomly assigned to one of three treatment groups, with 311 children in the budesonide group, 312 children in the nedocromil group, and 418 children in the placebo group. The initial clinical trial lasted about 22 months (between December 1993 and September 1995), and this was followed by the CAMP Continuation Study follow-up, which involved 941 participants over 4.5 years, with further extension of the follow-up through the second and third continuation studies. The study was primarily concerned with lung function as measured by the FEV at 1 second (FEV_1).

4.2 DATASET DESCRIPTION

The dataset variables selected as predictors included a treatment group, age, gender, race, hemoglobin level, leucocyte level, age of the building in which the participant lived, presence of pets or wood stove at home, use of a dehumidifier, and whether parents or family members smoked at home. The treatment group predictors included the TX predictor and TG predictor. The TX predictor spanned the following treatment groups: budesonide (bud), nedocromil (ned), budesonide placebo (pbud), and nedocromil placebo (pned), while the TG predictor spanned the following treatment groups: budesonide (A=bud), nedocromil (B=ned), and placebo (C=plbo).

The main outcomes selected for the computational experiment in this paper included the relative value (the ratio of the predicated value to the measured value) of FEV_1 prior to the administration of a bronchodilator (pre-bronchodilator) (PREFEVPP), the relative value of the pre-bronchodilator forced vital capacity (FVC) (PREFVCPP), the relative value of FEV_1 after the administration of a bronchodilator (post-bronchodilator) (POSFEVPP), and the relative value of the post-bronchodilator FVC (POSFVCPP).

In the original dataset, there was a line of data per patient per visit. Different patients had a different number of visits, varying from 1 to 20. For the TDA algorithm to run properly, the original dataset needed to be transformed such that each row within the dataset represented one participant within the clinical trial while the columns represented specific outcomes. If we had transformed the original dataset straight, we would have generated a lot of empty cells (missing data) because some values were not recorded during the patents' visits. As such, to follow the exploratory analysis workflow, the original data needed to be pre-processed before the analysis was run (see Figure 1). A more in-depth review of the data transformation that was performed is presented below.

For the purpose of the experiment, we used the PREFVCPP and PREFEVPP outcomes and time at which the values were measured to build a scatter diagram for the outcomes per patient. Figure 11 contains a scatter diagram of the PREFVCPP over time for a first patient, Figure 12 shows a scatter diagram of the PREFVCPP over time for a second patient, and Figure 13 shows a scatter diagram of PREFVCPP over time for a third patient.

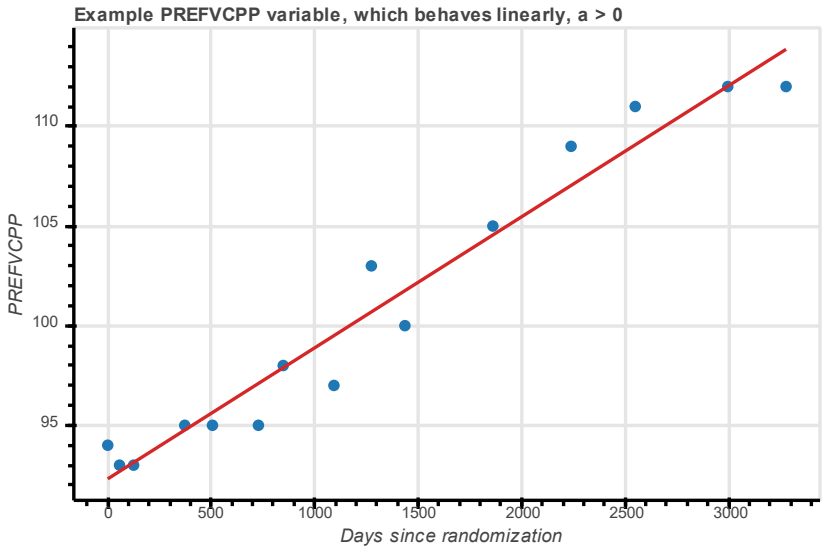


Figure 11. PREFVCPP in time for a first patient

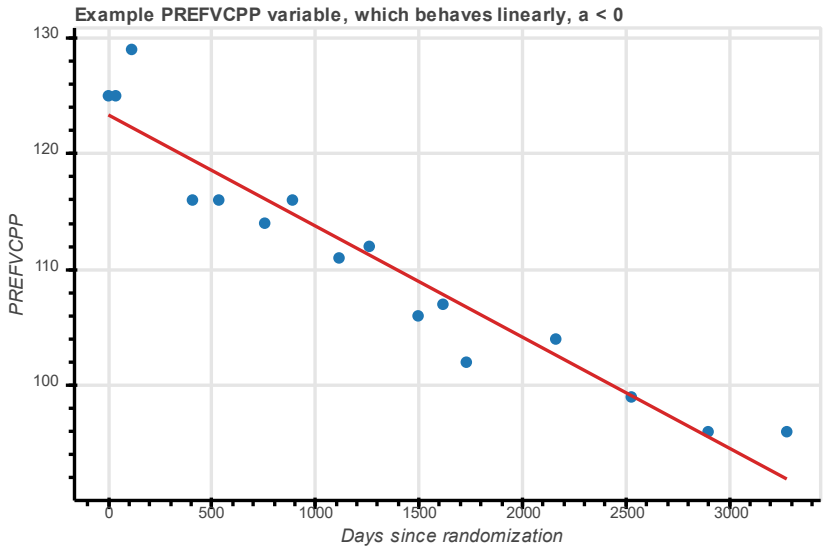


Figure 12. PREFVCPP in time for a second patient

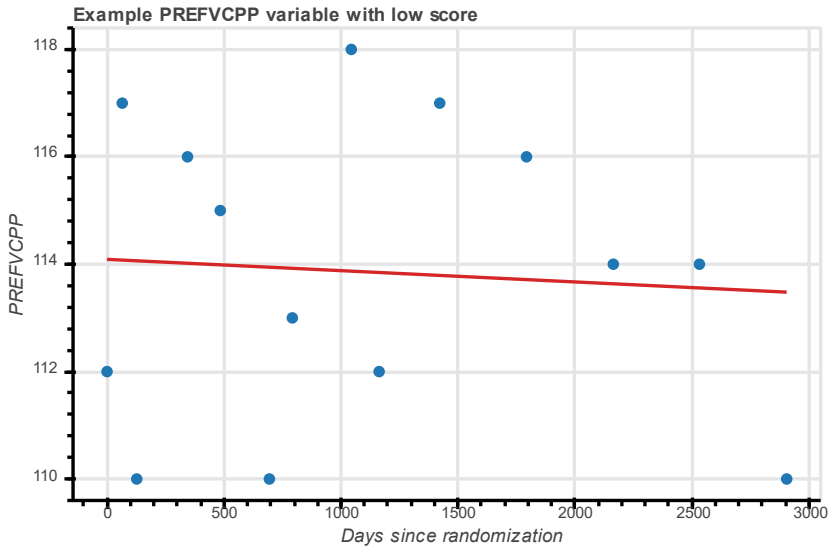


Figure 13. PREFVCPP in time for a third patient

As can be seen in Figure 11 and Figure 12, the PREFVCPP values exhibit a linear trend over time, i.e., lie approximately on a line. We applied a linear regression to model the relationship between the variable and time. The linear regression line obtained based on the linear regression provided approximated values of PREFVCPP over time at any point in time during the clinical trial, irrespective of the number of visits the patient had.

The linear regression line representing PREFVCPP over time can be described by the following linear regression formula:

$$\text{PREFVCPP} = \boldsymbol{a} \cdot \boldsymbol{t} + \boldsymbol{c}$$

where ***t*** is time, and ***a*** and ***c*** are regression coefficients.

The ***a*** coefficient represents an inclination of the linear regression line. ***a*** > 0 means that the value of the variable increases over time and ***a*** < 0 means that the value of the variable decreases over time. The increase in values indicates that the FEV and FVC of a patient rose during the clinical trial. The c coefficient is an approximated value of the variable at first visit; i.e., an approximated value of a measured parameter at the beginning of the treatment.

Further, the same approach for building scatter diagrams was implemented for the PREFEVPP outcome – the relative value (the ratio of the predicated value to the measured value) of FEV_1 prior to the administration of a bronchodilator (pre-bronchodilator) – to illustrate the dynamic over time for each patient.

Following the construction of the linear regression line, a quality score, referred herein to as the ***score***, of the linear regression model was calculated to determine the extent to which the data was coherent with the linear regression model. In other words, the ***score*** represents how accurately the linear regression model described the data. The score may range from 0 to 1. A high ***score*** means that the linear regression line lies close to the data and, hence, shows that the linear regression model is well fit to the data.

In Figure 11 and Figure 12, the ***score*** of the linear regression model is high, which means that the linear regression line approximates the data well. In Figure 13, the ***score*** of the linear regression model is low, which indicates that the linear regression model doesn't fit to the data.

4.3 EXPERIMENT WORKFLOW

After transforming the data using the linear regression, the PREFVCPP and PREFEVPP measurements for every patient can be described by three parameters: ***a***, ***c***, and ***score***. We used ***a***, ***c***, and ***score*** for PREFVCPP and PREFEVPP; as such, there were six values in total and six outcomes upon which a dataset could be constructed and used as the basis for the TDA algorithm.

We used proprietary patent-pending TDA algorithms to extract a graph from the constructed dataset in which each row represented a participant in the clinical trial. The six columns are associated with the outcomes ***a***, ***c***, and ***score*** for PREFVCPP and PREFEVPP. As a result, the computational platform generated a metric graph in which every node corresponded to one patient while two nodes representing similar patients (in terms of pre-defined outcomes) were connected with an edge (see Figure 14).

The graph generated from the clinical dataset clearly exhibits a Y-shape with three possible communities of related patients based on pre-defined outcomes. In this graph, we can plainly recognize the first community within the left branch of the graph, the second community within the right branch, and the third community at the stem of the Y-shape. As there are a significant number of nodes within the graph, it is difficult or almost impossible to identify sub-communities within the three large communities through the use of visual exploration alone; as such, there is a need to use special machine-learning algorithms for the purpose of the community search.

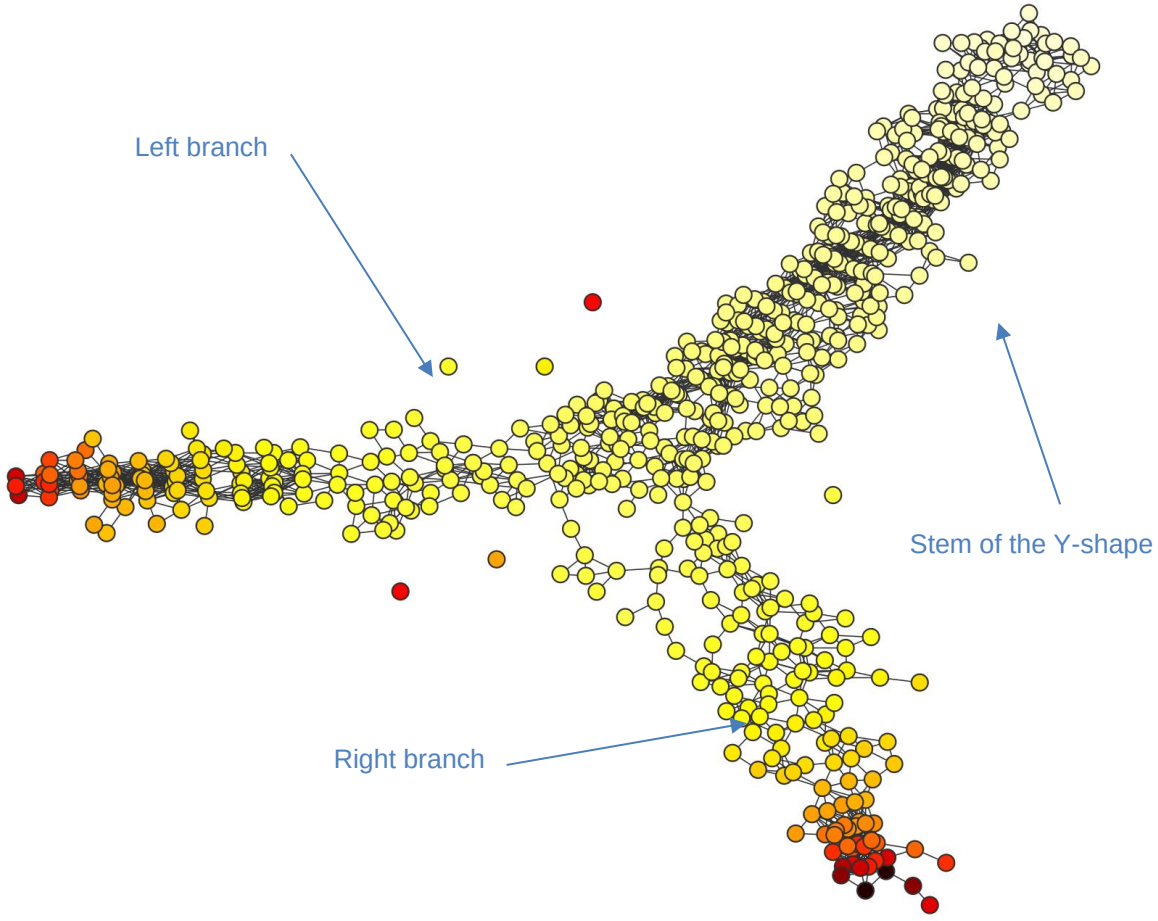


Figure 14. Y-shape graph generated by the TDA algorithm from the clinical dataset

As was previously discussed in Section 3, we used the Clique Percolation Method to perform an automatic search for communities. The key feature of a graph is community structure, which relates to the way the nodes are organized in communities based on the distribution of edges among nodes. The aim of community detection using the Clique Percolation Method is to identify communities within the graphs and their hierarchical organization by using the information that is contained within the graph topology alone.

The original Clique Percolation Method was modified to find the optimal value of *k*-clique. For our Y-shape graph, the algorithm identified the most optimal size of *k*-clique = 3. Running the algorithm further, we identified six communities based on the topology of the graph (see Figure 15). These six communities were used for further analysis.

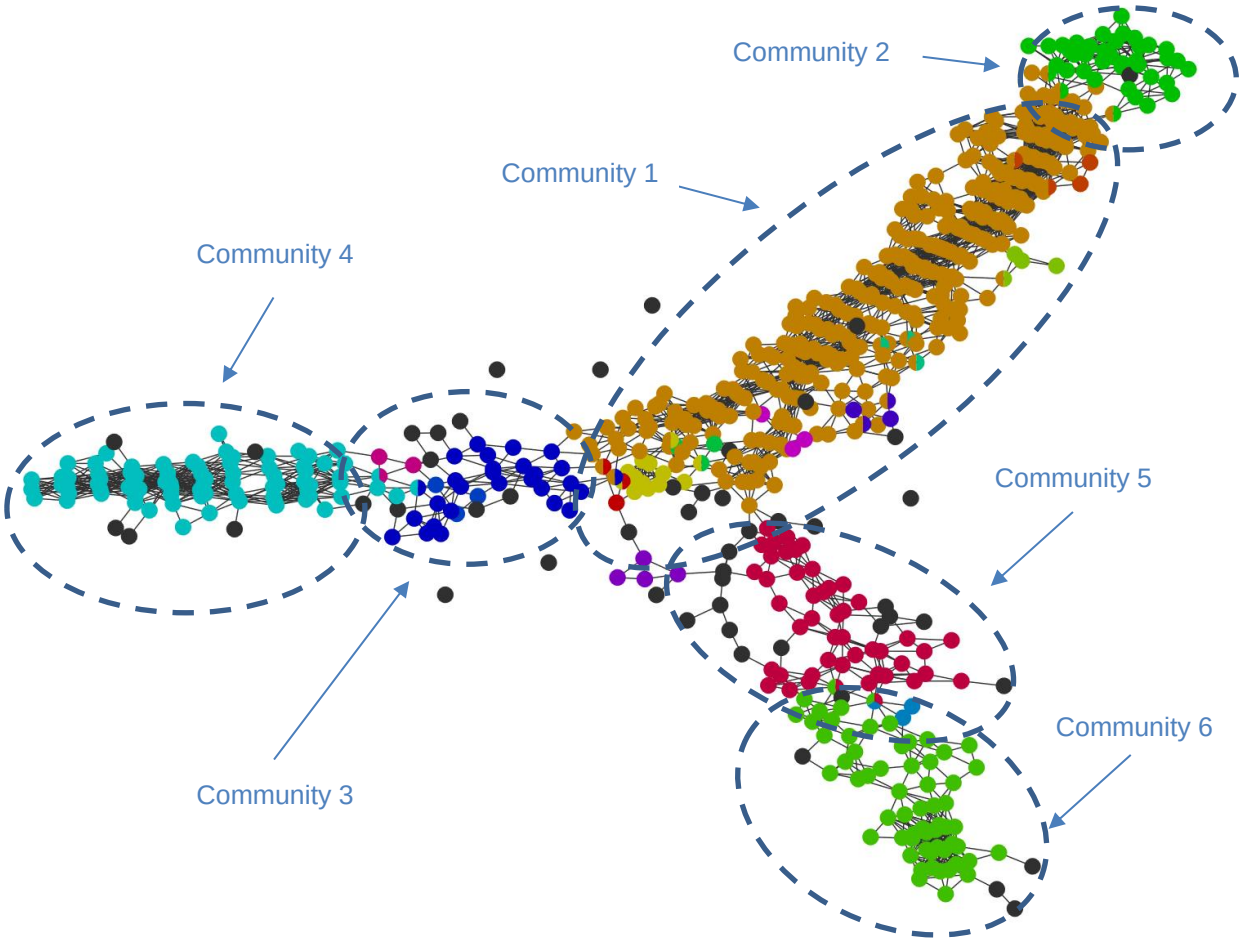


Figure 15. Communities 1-6 were identified by running the Clique Percolation Method with *k*-clique = 3.

4.4 RESULTS AND DISCUSSION

To discover the reasons why those six communities are different from each other additional statistical analysis and graph manipulation were performed. Thus, specifically, the communities were analyzed based on the ***score*** value by changing the color of the nodes on the graph. As a result, we found that Community 1 had a low ***score***, Community 2 had a low ***score*** but a slightly higher ***score*** than Community 1, Communities 3 and 4 had a high ***score*** with an increase in values from lower to higher, Community 5 had a mean score with a decrease in the values from higher to lower, and Community 6 had a high ***score*** with a decrease in the values from higher to lower.

Further analysis of the communities based on the ***a*** value revealed that Communities 3 and 4 (in the left branch) had ***a*** > 0; i.e., the values of the PREFEVPP and PREFVCPP parameters increased during the treatment in these communities. Communities 5 and 6 (in the right branch) had ***a*** < 0; i.e., the values of the PREFEVPP and PREFVCPP parameters decreased during the treatment of Communities 5 and 6.

Looking at the ***c*** value, Communities 5 and 6 (in the right branch) had a higher value of ***c*** in comparison to Communities 3 and 4 (in the left branch). This indicates that patients in Communities 5 and 6 had higher PREFEVPP and PREFVCPP values at the beginning of the treatment.

The analysis further included a comparison of two groups of communities with each other. In Figure 16, we combined Communities 3 and 4 (in the left branch) into one group (blue color) and Communities 5 and 6 (in the right branch) into another group (red color). The chi-squared test showed the statistical significance of the differences between the TX and TG predictors (p-value = 0.009) for these two groups. Figure 16 shows a comparison of these two groups based on the TX predictor. The comparison reveals that group marked red (Communities 5 and 6) included 44 patients (the majority) treated with budesonide and 22 patients treated with nedocromil, and the group marked blue (Communities 3 and 4) included 40 patients (the majority) treated with nedocromil and 27 patients treated with budesonide. Therefore, the comparison of these two groups shows that Communities 3 and 4 differ from Communities 5 and 6 in terms of the treatment group.

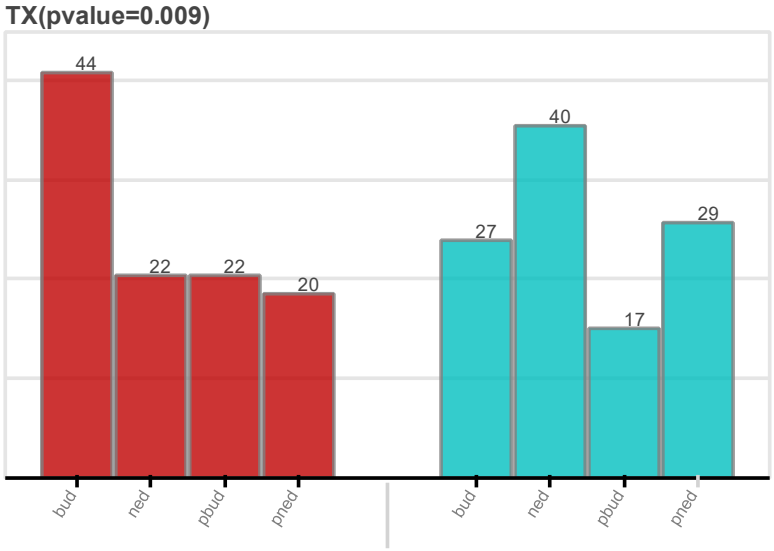


Figure 16. Comparison of two groups of patients based on TX predictor.
Red group of patients is related to the treatment groups for Communities 5 and 6 (the right branch), while the blue color shows the number of patients in treatment groups for Communities 3 and 4 (the left branch).

The next step involved the analysis of the time dependences of the mean values of PREFEVPP, and a 95% confidence limit was established for patients within each of the six communities, see Figures 17-22. The findings revealed that the treatment group appeared to be a significant parameter that may have an effect on the outcomes of the communities.

Thus, the multi-staged analysis generated several significant insights. First, the topological features of the graph based on the application of the TDA algorithm to the clinical dataset indicated that there were three separate groups of patients, as indicated by the different branches of the Y-shape. Second, an automatic search of the communities analyzed topology of the graph using the Clique Percolation Method and revealed that there were actually six communities, as evidenced by the number of edges that connected nodes on the graph. Third, further statistical analysis in combination with a visual exploration of the graph revealed that several patient communities were different from one another in terms of outcomes. Each community was analyzed to identify the parameters that may be considered as factors that influence the outcomes.

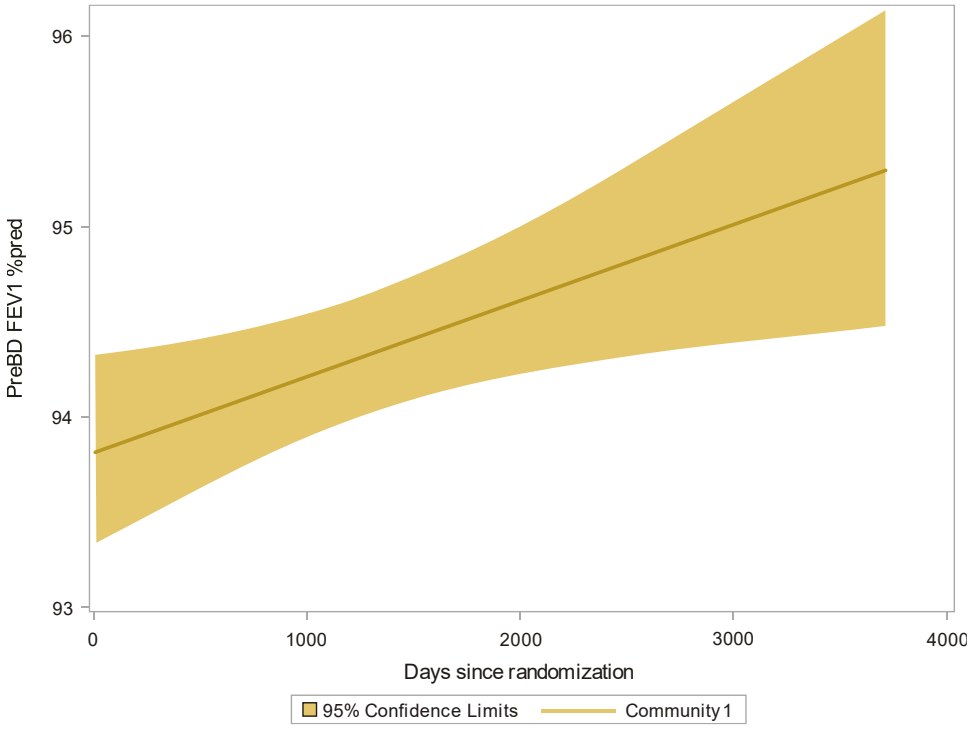


Figure 17. Time dependence of mean values of PREFEVPP for patients in Community 1

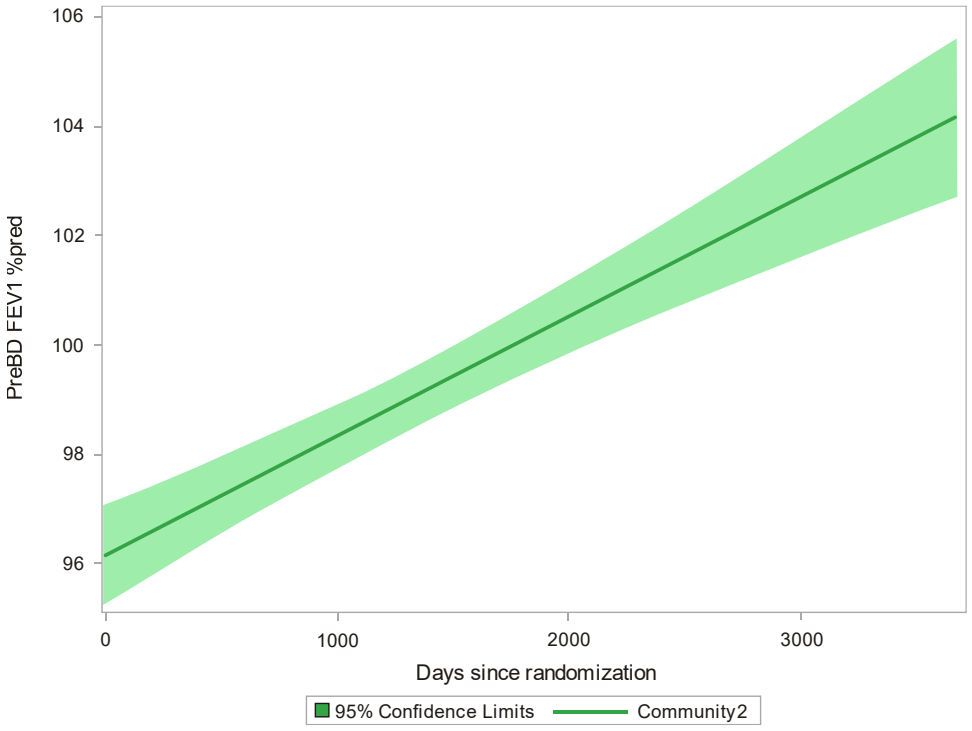


Figure 18. Time dependence of mean values of PREFEVPP for patients in Community 2

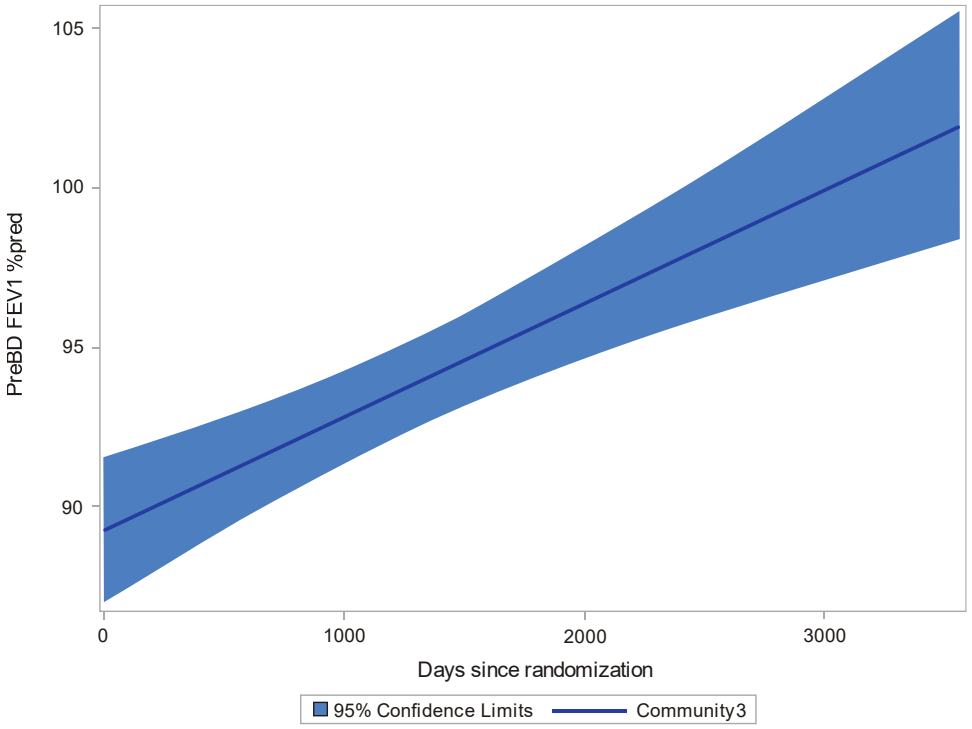


Figure 19. Time dependence of mean values of PREFEVPP for patients in Community 3

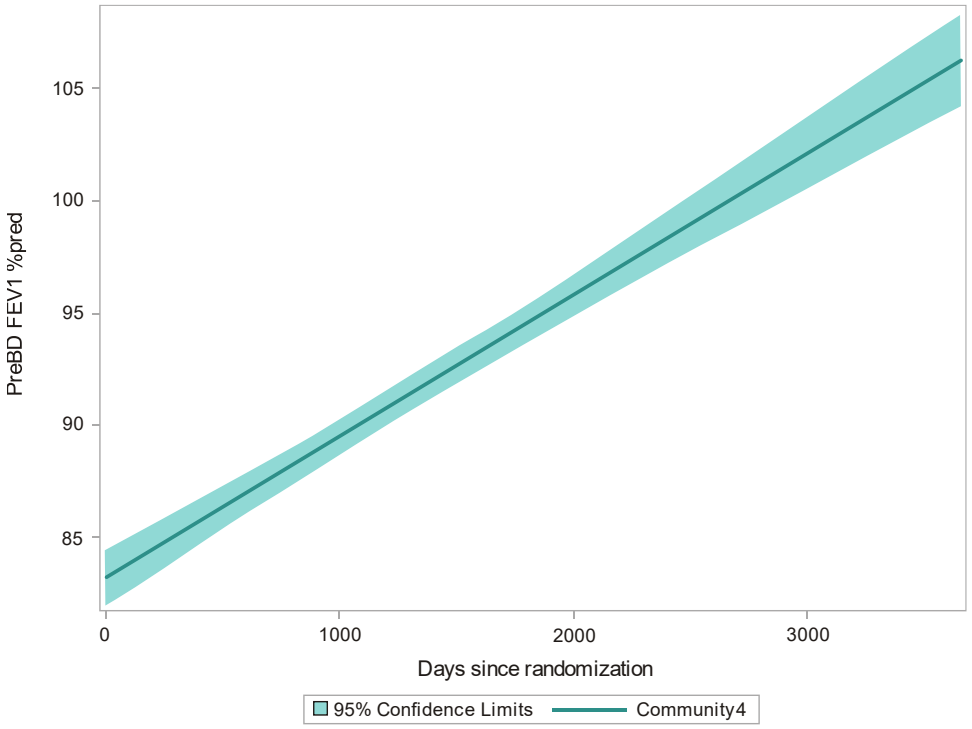


Figure 20. Time dependence of mean values of PREFEVPP for patients in Community 4

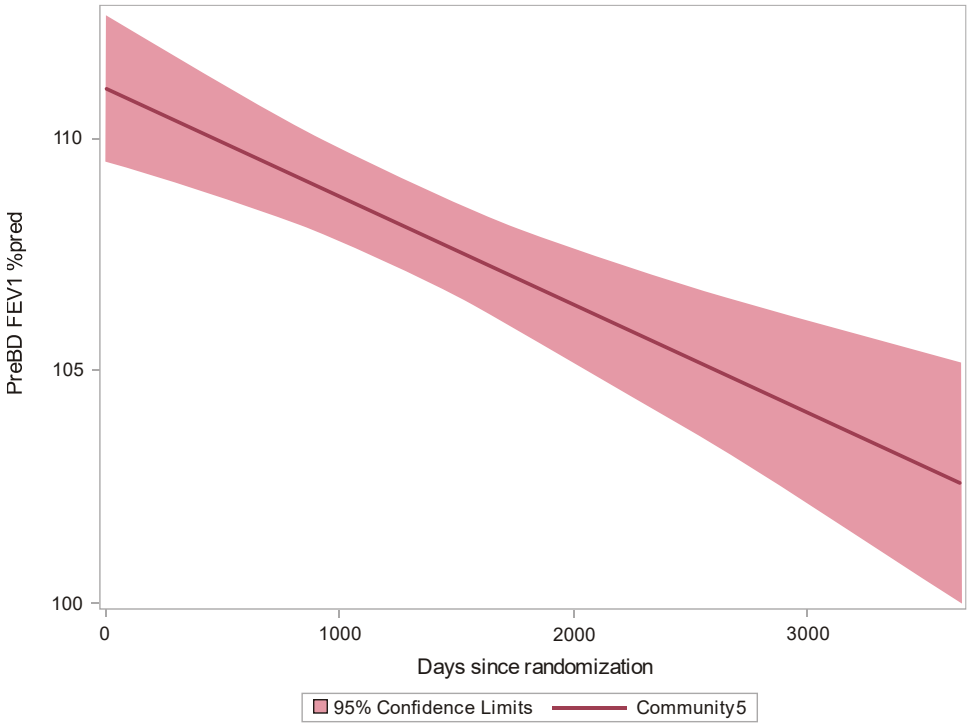


Figure 21. Time dependence of mean values of PREFEVPP for patients in Community 5

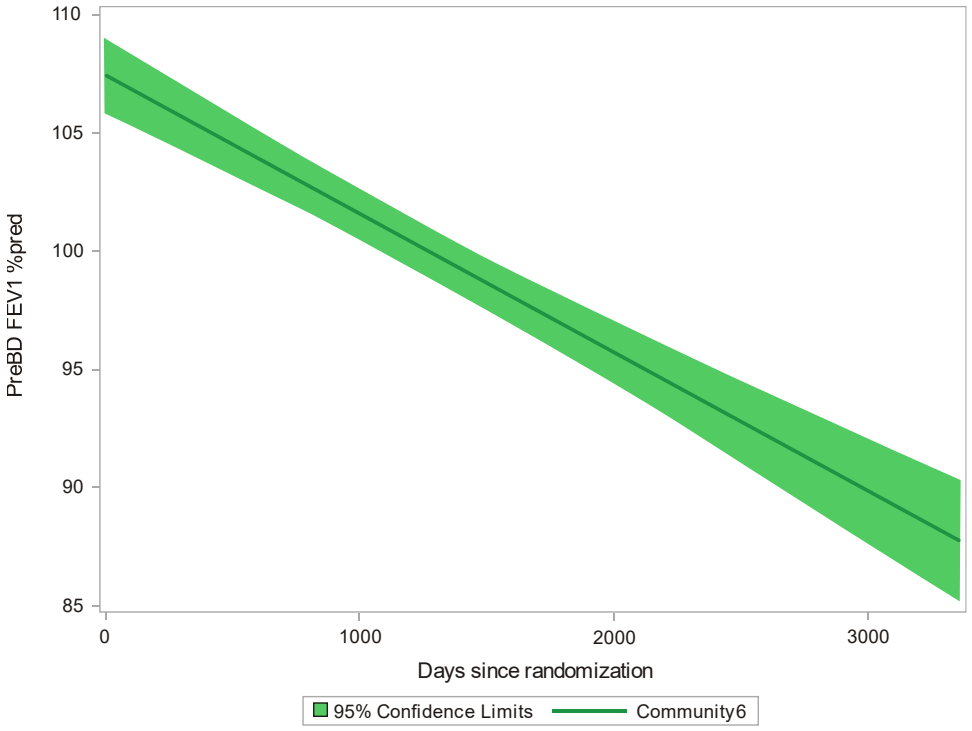


Figure 22. Time dependence of mean values of PREFEVPP for patients in Community 6

5. CONCLUSION

In this paper authors rely on graphs as a fundamental approach to structuring and analyzing data in a clinical study. As datasets may include a large number of participants, visual exploration on the graph may be challenging or even misleading. We described machine-learning algorithms applied to automatic detection of sub-populations of similar patients using a graph-based community search, specifically the Clique Percolation Method. The multi-staged experiment generated several significant insights. First, the topological features of the constructed graph indicated that there were three separate groups of patients, as indicated by the different branches of the Y-shape. Second, an automatic search of the communities analyzed topology of the graph and revealed that there were actually six communities, as evidenced by the number of edges that connected nodes on the graph. Third, further statistical analysis in combination with a visual exploration of the graph revealed that several patient communities were different from one another in terms of outcomes.

REFERENCES

[1] Tukey, John W. (1977). Exploratory Data Analysis. Pearson. ISBN 978-0201076165.
[2] Edelsbrunner, Herbert; Harer, John (2010). Computational Topology: An Introduction. American Mathematical Soc. ISBN 9780821849255.
[3] Zomorodian, Afra (2005). Topology for Computing. Cambridge University Press. ISBN: 9780511546945.
[4] Carlsson, Gunnar (2009). "Topology and data". Bulletin of the American Mathematical Society. 46(2): 255-308.
[5] Qualification Process for Drug Development Tools (2014). U.S. Department of Health and Human Services. Food and Drug Administration Center for Drug Evaluation and Research (CDER). p. 1-35.
[6] Biomarkers Definitions Working Group (2001). Clinical Pharmacology and Therapeutics, 69, p. 89-95.
[7] Fortunato, Santo. (2009). Community Detection in Graphs. Physics Reports. 486.
[8] Rombach, M. P., Porter, M. A., Fowler, J. H., & Mucha, P. J. (2014). Core-periphery structure in networks. SIAM Journal on Applied mathematics, 74(1), 167-190.
[9] Girvan M. and Newman M.E.L. Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA 99, 7821-7826 (2002)
[10] Clauset A, Newman M.E.L., Moore C. Finding community structure in very large networks. Phys. Rev. E, 70:066111, 2004.
[11] Zhou, H., 2003a, Phys. Rev. E 67(6), 061901.
[12] Zhou, H., and R. Lipowsky, 2004, Lect. Notes Comp. Sci. 3038, 1062.
[13] Latapy, M., and P. Pons, 2005, Lect. Notes Comp. Sci. 3733, 284.
[14] Palla, Gergely & Derényi, Imre & Farkas, Illés & Vicsek, Tamás. (2005). Uncovering the overlapping community structure of complex networks in nature and society. Nature. 435. 814-818.
[15]. More information about the clinical study can be found here: https://www.ncbi.nlm.nih.gov/pubmed/10027502

ACKNOWLEDGMENTS

We would like to acknowledge Victoria Shevtsova (Intego Group, Ukraine), Bogdan Chornomaz (Kharkiv National University, Ukraine / Vanderbilt University, United States), Yan Rybalko (Kharkiv National University, Ukraine) and Lyudmyla Polyakova (Kharkiv National University, Ukraine) for being core members of the research team and for the significant contribution they made to the development of the mathematical foundation of the TDA methodology. Without you this research would not have been possible.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the group of authors at:

Contact: Sergey Glushakov
Company: Intego Group
Address: 555 Winderley Place, Ste. 129, Maitland, FL 32751
Work Phone: 407.641.4730
Email: sergey.glushakov@intego-group.com
Web: www.intego-group.com