Anlauf *et al. Applied Network Science* (2025) 10:54
https://doi.org/10.1007/s41109-025-00743-6

Applied Network Science

RESEARCH Open Access

SimNetX: tinkering with patient similarity networks to understand biomedical data

Tomas Anlauf[1*], Kristyna Kubikova[1], Eliska Ochodkova[1], Eva Kriegova[2] and Milos Kudelka[1]

*Correspondence:
Tomas Anlauf
tomas.anlauf@vsb.cz
[1]Department of Computer Science, VSB - Technical University of Ostrava, 17. listopadu, 708 00 Ostrava, Poruba, Czech Republic
[2]Department of Immunology, Palacky University Olomouc and University Hospital Olomouc, Krizkovskeho, 779 00 Olomouc, Czech Republic

Abstract
Analyzing complex biomedical data often requires statistical and machine learning expertise, creating barriers for clinicians, laboratory scientists, and other non-technical users. Patient similarity networks (PSNs) offer an intuitive way to explore patient relationships and patterns, making data interpretation more accessible. However, constructing and analyzing PSNs typically involves multiple software tools or programming skills, limiting their usability for those without technical expertise. In this article, we introduce an approach that enables non-technical users to analyze biomedical data through PSNs without requiring programming knowledge. By integrating key functionalities-such as transforming vector-based data into networks, interactively exploring patient relationships, and applying statistical insights-this approach bridges the gap between complex data analysis and domain experts. To facilitate this, we provide a tool designed to implement these methods in an intuitive, interactive environment. We demonstrate its practical application using two well-known datasets as well as two real-world biomedical datasets, showing how non-experts can generate hypotheses and extract meaningful insights through visual exploration and built-in simple statistical analysis.

Keywords Patient similarity network, Biomedical data, Data visualization

Introduction

In the healthcare sector, a vast volume of patient-related data is collected on a daily basis. These datasets are often intricate and pose considerable difficulties during analysis. To extract precise and clinically relevant insights from such diverse sources—including patient-reported outcomes and genomic information—it is essential to employ advanced statistical techniques along with robust computational tools. One such valuable approach is multivariate analysis, which has gained prominence for its capacity to handle and interpret data involving numerous variables at once. These datasets typically encompass clinical metrics, lifestyle details, genetic profiles, and imaging data. This analytical method is especially effective in uncovering variables that influence disease susceptibility, progression, and therapeutic response.

In this article, we introduce an approach to analyzing biomedical data that leverages patient similarity network (PSN) exploration, making complex data more accessible

to non-technical users. To facilitate this, we provide SimNetX (SIMilarity NETwork eXplorer), a tool designed to support the interactive construction and analysis of PSNs (Mikulkova et al. 2021; Pai and Bader 2018). In a PSN, patient records (vector data) are represented as nodes, with edges indicating similarities between them. Patients are considered similar when their similarity is high enough, with network connections formed based on a threshold that can be set manually or determined by a network construction algorithm. PSNs offer a valuable way to visually explore patient relationships, revealing potential subgroups with shared clinical features or genetic markers. However, there is still no widely accepted methodology for fully utilizing PSNs or systematically determining the types of insights they can provide.

The intricate nature of multivariate patient data analysis in biomedical research frequently necessitates collaboration among various experts and the integration of multiple systems. This paper introduces a clinician-friendly tool designed to streamline the analysis process, eliminating the need for in-depth technical expertise. Its development was informed by firsthand experience with the complex, multi-stage procedures commonly encountered in biomedical research environments. By reducing dependence on specialized data professionals, the tool enhances workflow efficiency and supports clearer data interpretation within clinical practice. It enables users to explore datasets interactively through similarity networks, offering a foundation for formulating preliminary hypotheses. Although the tool is primarily intended for clinical users, this article also provides a technical overview of the methods used to construct and analyze these networks, serving as an introduction for researchers with a background in machine learning. A comprehensive explanation of the underlying algorithms is beyond the scope of this publication and is therefore only briefly mentioned.

To our knowledge, no existing system is capable of converting vector data into a network structure while allowing for instant exploration and analysis. As a result, we were unable to directly compare our tool with a similar system, as none appears to integrate both network construction and analysis on a single platform. The tool we present has already proven its worth in several cases, for example, Janca et al. (2023); Mikulkova et al. (2021); Sova et al. (2022).

This article is an extension of the Complex Networks and their Applications conference paper titled "SimNetX: Interactive support for biomedical data analysis using patient similarity networks" Anlauf et al. (2024). It delves deeper into the methodology behind the tool and provides a more detailed description of the analysis workflow. We conducted analysis experiments using the tool on four datasets, including datasets with real patient data, rather than proving the concept only on well-known datasets.

Our article is structured to provide a comprehensive understanding of our tool designed for patient similarity network (PSN) analysis. We begin by reviewing existing systems that incorporate PSN analysis, highlighting their capabilities, limitations, and how our approach differs. Next, we explore the motivation that led to the development of our tool. The following section delves into the theoretical foundations of the methods employed in our tool, ensuring a clear understanding of the underlying computational techniques. We then present a detailed examination of the tool itself, outlining its key functions, user interface, and workflow. Finally, we demonstrate the practical application of the tool through a series of experiments, showcasing how it can be used to analyze a

dataset and generate initial hypotheses, thereby illustrating its effectiveness in biomedical research.

For consistency, throughout this article, we will refer to dataset columns as attributes, numeric attributes as features, categorical attributes as labels, and communities as clusters.

Related work

Patient similarity networks (PSNs) are gaining traction in clinical research and precision medicine, offering a powerful approach to modeling and predicting patient outcomes, phenotypes, and disease risk. In these networks, patients are represented as nodes, with weighted edges denoting degrees of similarity. PSNs effectively combine diverse data sources, such as omics, clinical records, laboratory results, and imaging data, to provide a holistic perspective on patient relationships and improve the interpretability of machine learning models (Gliozzo et al. 2025). A key advantage of PSNs is their ability to handle incomplete multimodal datasets, where some data types may be absent for certain individuals. Learning strategies based on message passing have proven successful in integrating such incomplete data, as demonstrated using breast cancer data from The Cancer Genome Atlas (TCGA) (Gliozzo et al. 2023). Beyond profiling, PSNs have applications in therapeutic decision-making, including drug recommendations. For example, the DAPSNet framework utilizes PSNs to represent various medical codes and incorporates drug-related knowledge from patients in similar clinical states, emulating clinician reasoning and enhancing predictions of effective drug combinations (Wu et al. 2023). PSNs also support similarity-based patient retrieval, which is crucial for tailoring treatments. Techniques based on heterogeneous information networks (HINs) help overcome the limitations of sparse and high-dimensional data representations by linking patients with diseases and medications, while also capturing temporal aspects (Huang et al. 2021).

Recent developments have further refined PSNs through hierarchical clustering and contrastive learning techniques. For example, a study titled "Explainable hierarchical clustering for patient subtyping and risk prediction" presents a pipeline that hierarchically subtypes patients in an explainable manner, leading to improved outcome predictions for many identified subtypes (Werner et al. 2023). In addition, contrastive learning techniques have been applied to calculate patient similarity in large electronic health record datasets, using graph-based similarity analysis to extract clinical characteristics and aggregate information from similar patients (Liu et al. 2024).

The effective use of PSNs in clinical settings also depends on visualization tools that can help researchers and clinicians explore patient data. While advancements in algorithmic approaches have significantly improved PSNs, effective visualization remains a critical challenge in clinical settings. Tools such as MetaRelSubNetVis enable group-wise comparison of PSNs, enabling interactive exploration of patient-specific attributes (Auer et al. 2022). This is in line with Van den Elzen and van Wijk's approach to network exploration, emphasizing the importance of transitions from detailed views to broader overviews through selections and aggregations, which are critical for making sense of multivariate network data (Elzen and Van Wijk 2014). Such functionalities are particularly valuable in clinical settings where effective communication and data sharing are crucial.

To further enhance the visualization and interpretability of PSNs, recent tools such as CHDmap and CompositeView have been developed. CHDmap provides an interactive visualization of patient similarities based on echocardiographic indicators, specific diagnoses, and surgical features, helping clinicians predict outcomes after congenital heart surgery (Li et al. 2024). While focused on a specific patient cohort, this tool enables real-time navigation through patient similarity networks, supporting more informed clinical decision-making. CompositeView, on the other hand, facilitates the integration and exploration of heterogeneous network-based data, allowing researchers to interactively analyze complex datasets and uncover underlying patterns (Allegri et al. 2022). Both tools provide valuable information for clinical practice, but are primarily designed for specific domains or types of data.

Although patient similarity networks (PSNs) offer considerable potential, they are often hindered by the complexity of their network structures and the challenges in interpreting similarity metrics. Many existing classification models fail to fully capitalize on the advantages that PSNs offer, resulting in networks that are overly complicated and not used to their full capacity. Nonetheless, recent innovations such as StellarPath, a method that hierarchically integrates omics data and applies graph convolutional neural networks, demonstrate the potential to improve the effectiveness of PSNs by generating informative features and enhancing classification outcomes (Giudice et al. 2024).

In addition, several other state-of-the-art GNN-based methods have been proposed. MOGONET (Wang et al. 2021) employs modality-specific GNNs with a cross-omics attention mechanism, while MoGCN (Li et al. 2022) combines GCNs with a co-training strategy to exploit both labeled and unlabeled samples. SUPREME (Kesimoglu and Bozdag 2022) integrates patient similarity graphs across omics layers into a unified GNN framework, MOGAT (Tanvir et al. 2024) applies attention mechanisms to capture inter- and intra-omics relationships, and MORE (Wang et al. 2024) introduces a multi-level relational GNN to model cross-omics dependencies. Together, these approaches highlight the growing role of GNN-based models in integrating multi-omics data, uncovering complex patient similarities, and enhancing predictive performance.

Furthermore, the incorporation of deep learning methods, including BERT, convolutional neural networks (CNNs), and LSTM-based autoencoders, has helped overcome issues related to data heterogeneity and high dimensionality. These advances not only improve classification accuracy, but also maintain temporal data characteristics (Navaz et al. 2022). The netDx framework provides another strong example of PSN utility, outperforming conventional machine learning models in predicting cancer survival and uncovering relevant biological pathways (Pai et al. 2019).

To provide a clearer comparison, Table 1 summarizes the capabilities of SimNetX (described later), MetaRelSubNetVis, CHDmap, and CompositeView. *Network construction* captures whether a tool can transform raw tabular data directly into a network representation. *Similarity functions* reflect the ability to define patient similarities using different mathematical functions, such as correlations or distance metrics. *Group-wise comparison* indicates whether users can compare predefined groups of patients within the same network visualization. *Composite scores* concern the ability to calculate and visualize composite scores that summarize multiple variables into a single measure. *Multivariate data* shows whether the tools can integrate diverse multivariate attributes, such as clinical, genomic, or imaging data. *Statistical analysis* highlights support for statistical

Table 1 Feature comparison across SimNetX, MetaRelSubNetVis, CHDmap, and CompositeView

Feature	SimNetX	MetaRelSubNetVis	CHDmap	CompositeView
Network construction	✓	✗	✓	✗
Similarity functions	✓	✗	✓	✗
Group-wise comparison	✗	✓	✗	✗
Composite scores	✗	✗	✗	✓
Multivariate data	✓	✓	✓	✓
Statistical analysis	✓	✗	✓	✗
Layout controls	✓	✓	✗	✓
Exportable results	✓	✗	✗	✓
Reproducibility	✓	✓	✗	✓
Availability	Web app	Web app	Research tool	Open-source app

analyses, for example, cluster evaluation or predictive validation. *Layout controls* represent the ability to interactively adjust network layouts, helping users explore structural patterns. *Exportable results* cover whether results, such as networks or metrics, can be exported for reuse. *Reproducibility* reflects the capacity to preserve and share complete interactive visualization states for reproducibility. *Availability* describes the type of availability of each tool, such as a prototype, web-based system, research-only, or open-source application.

What differentiates the SimNetX tool is its ability to construct networks directly from vector data while maintaining a strong focus on interactive and comprehensive visualization, ensuring that users can immediately see how their input configurations impact the network and statistical outputs in real time. Beyond visualization, our tool supports a wide range of data manipulation techniques. Users can apply various filtration methods to refine the network and perform feature transformations such as normalization, scaling, standardization, and conversion from log-normal to normal distributions, functionalities often restricted or unavailable in other tools. To further support data exploration, the tool includes histogram generation for visualizing feature distributions, making it a well-rounded solution for detailed analysis and visualization of PSNs.

SimNetX as an observation-based tool

Several years of collaboration between clinicians, laboratory personnel, and computer scientists in the preparation of research articles related to biomedical data analysis have highlighted the interesting aspects of this collaboration. The key aspects, which can be summarized in several observations, were the main motivation for developing the SimNetX tool.

Observation 0: class-cluster relationship

In addition to common machine learning tasks such as classification, which are not addressed in this paper, the combination of supervised and unsupervised methods has emerged as a key task corresponding with multivariate analysis. As a result, in our case, it is about finding the relationship between automatically found groups (clusters) of patients with similar characteristics and classes describing, e.g., symptoms, disease stage, treatment success, etc. A related expectation is a comprehensible, explainable, and interpretable form of results.

Observation 1: preparing the dataset

Real biomedical data recorded by clinicians and supplemented in laboratories are often clouded by inaccuracies (equipment errors, human factors), incomplete and heterogeneous (aggregated from different sources). However, clinicians and laboratory staff like to take the first steps to prepare and perform simple statistical analysis of the kit themselves using various user-friendly tools. These are mainly univariate statistical analyses, selection of useful labels and features, finding frequencies of occurrence, means, medians and quartiles, confidence intervals, significance, etc. Sometimes, their pre-analysis requires multiple steps and consultations.

Observation 2: configuration of datasets

After pre-analysis and preparation of the set for deeper analysis, multiple experiments with different settings are expected. These include reducing the dataset, i.e., excluding some patients and features from the analysis, using one of the labels to represent classes, and different types of normalization of selected features.

Observation 3: patient data as a network

The patient similarity network has become an absolutely crucial tool. This is mainly due to the possibility of straightforward visualization, which supports a clear interpretation of the analysis results. In addition, it is relatively easy to learn to 'read' the network and perceive the relatively complex relationships between classes and clusters through simple visualization.

Observation 4: augmented network visualization

The power of network visualization, especially when combined with network coloring according to different aspects, e.g., expressing feature values by gradient or network node sizes, coloring nodes, etc., was demonstrated in collaboration with clinicians and laboratory staff.

Observation 5: visual statistics

Visual statistical tools are strongly preferred for team communication and, generally, for the joint detailed examination of analytical results. The most understandable ones were barplots, boxplots, silhouettes (to evaluate the quality of clusters), distributions of feature values, and boxplots for individual features in clusters.

Observation 6: preservation of experiments

The ability to keep experiments in progress is a very important expectation. One dataset is often associated with multiple repeated experiments that require frequent revisitation and modification of their settings.

PSN visualization and analysis

Figure 1 provides an overview of a systematic approach to analyzing patient similarity networks. It illustrates the steps of transforming vector-based patient data into a network representation, allowing users to explore patient similarities visually. The figure shows how data can be interactively modified, including row and column removal, as well as column normalization, before regenerating an updated network. Various

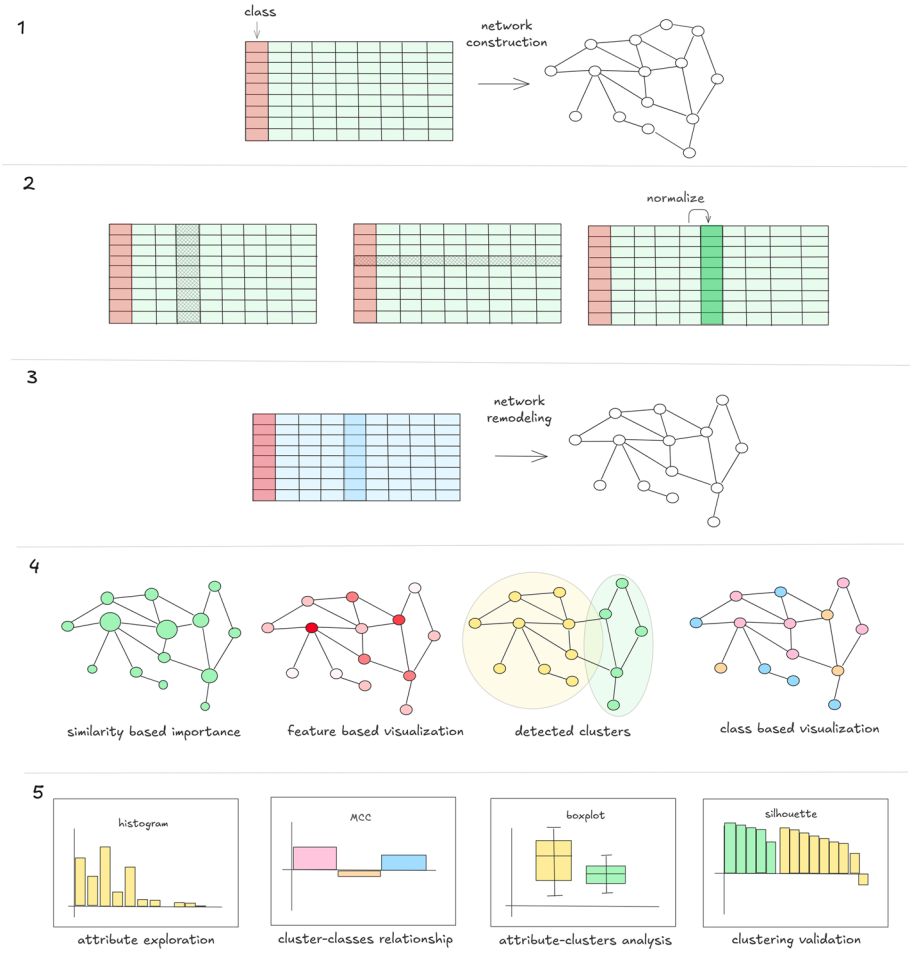

Fig. 1 Overview diagram

visualization options are also demonstrated, such as adjusting node size based on degree (reflecting similarity-based importance), applying color gradients for feature-based representation, distinguishing detected clusters using different colors, and coloring nodes based on classes.

Additionally, visual statistics plays a crucial role, offering insights through histograms for feature exploration, the Matthews correlation coefficient (MCC) to assess class-cluster relationships, boxplots to examine feature-cluster associations, and the silhouette coefficient for clustering validation. While the figure highlights selected functionalities, further details on these capabilities and their usage will be described later in this article. . By integrating these capabilities, the SimNetX tool bridges the gaps in current visualization approaches, providing a more comprehensive and interactive framework for PSN analysis.

Theoretical background

Multivariate network analysis encompasses a variety of methods used to examine and understand the structure, behavior, and dynamics of networks where multiple types of relationships or attributes are present. These methods, grounded in Graph Theory, provide tools for analyzing the relationships between entities represented as nodes and their connections as edges, while also accounting for additional dimensions of data such as

attributes or multiple interaction types. By applying different analytical techniques, multivariate network analysis aims to uncover patterns, identify key components, and evaluate the overall structure of a network.

In this context, several key methods are applied in our multivariate network analysis tool, each offering distinct insights into network structure and behavior. These include methods for data transformation; techniques for constructing networks from vector data; network layout algorithms; cluster detection, and basic statistical measures for evaluating detected clusters. These methods will be further explored to illustrate their contribution to the analysis of the PSN network.

Data normalization

Data transformation techniques are essential in data preprocessing to ensure that datasets are suitable for analysis, modeling, and machine learning algorithms. These techniques help improve the accuracy and efficiency of the models by adjusting the scale, distribution, or format of the data. Key transformation methods, which we focus on, include max normalization (Eq. 1) and min-max normalization (Eq. 2), which rescales the data to a fixed range (e.g. [0,1]); standardization (Eq. 3), which adjusts the data to have a mean of 0 and a standard deviation of 1; and changing the distribution, such as applying a logarithmic transformation to convert a log-normal distribution into a normal one (Eq. 4). These methods improve comparability and stabilize variance.

Let X be the vector of data, x the original data point, x' the new transformed value, X_{min} the minimum value in X, X_{max} the maximum value in X, μ the mean of X, and σ the standard deviation of X, then to calculate the maximum normalization is defined in Eq. (1), the min-max normalization in Eq. (2), the standardization in Eq. (3) and the log-normal to normal distribution conversion in Eq. (4).

$$x' = \frac{x}{|X_{max}|} \tag{1}$$

$$x' = \frac{x - X_{min}}{X_{max} - X_{min}} \tag{2}$$

$$x' = \frac{x - \mu}{\sigma} \tag{3}$$

$$x' = ln(|x| + 1) \tag{4}$$

Network construction

A set of vector data O is not always accompanied by a corresponding network that describes the relationships between individual objects. Therefore, if the data set does not contain this information, the network must be created on the basis of the calculated measure between each object. The resulting network should preferably be constructed in such a way that clusters, outliers, nearest neighbors, and other properties are preserved (Ochodkova et al. 2017).

We utilized two algorithms that solve the problem of converting vector data to the network. But first, commonly used similarities will be described as they are an integral part of the algorithms.

Similarities

Similarity is a function ρ: $O \times O \rightarrow \mathbb{R}_+$. The similarity value ranges from 0 to 1. For each definition, let $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$ be n–dimension vectors of objects from *O*.

Gaussian kernel is a nonlinear function of Euclidean distance using variance. The similarity is defined in Equation 5.

$$s(\vec{x}, \vec{y}) = e^{-\frac{\|\vec{x}-\vec{y}\|^2}{2\sigma^2}} \tag{5}$$

where $\vec{x}$ and $\vec{y}$ are vectors and σ is a standard deviation of the normal distribution. A value of σ controls the spread of the function, where smaller values make the kernel more sensitive to local differences, and larger values result in smoother, more generalized similarity measures. Optimal σ depends on the dataset.

Cosine similarity (Han et al. 2011) measures the similarity between two vectors of an inner product space. Measured by the cosine of the angle between two vectors, it determines whether two vectors point in roughly the same direction. A cosine value of 0 means that the two vectors are at 90-degree angle to each other (orthogonal) and do not have a match. The closer the cosine value to 1, the smaller the angle, and the greater the match between the vectors. It is often used to measure document similarity in text analysis. The similarity is defined in Equation 6.

$$s(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||x|| \cdot ||y||} \tag{6}$$

where $||x||$ is the Euclidean norm of vector x, defined as $\sqrt{x_1^2, x_2^2, \ldots, x_n^2}$ and similarly $||x||$ is the Euclidean norm of vector y.

The Pearson correlation(r) coefficient measures the linear relationship between two continuous variables, assuming a normal distribution and homoscedasticity. The Spearman correlation coefficient ($\rho\rho$), on the other hand, is a rank-based measure that assesses the strength and direction of a monotonic relationship between variables. Both coefficients range from -1 to $+1$. We modified the coefficients to allow them to provide similarity between objects by calculating the absolute value. The similarity utilizing the Pearson coefficient is defined in Equation 7, and with the Spearman coefficient is defined in Equation 8.

$$r = \left| \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2} \sqrt{\sum (y_i - \overline{y})^2}} \right| \tag{7}$$

where x_i, y_i are the values of the feature i in the object vectors $\vec{x}$ and $\vec{y}$; $\overline{x}, \overline{y}$ are vector means.

$$\rho\rho = \left| 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \right| \tag{8}$$

where d_i is the difference between the ranks of corresponding values in $\vec{x}$ and $\vec{y}$ and n is the number of features.

Co-occurrence is the frequency of paired terms that are present in a given context together. This is typically used in text analysis, where the task might be to find words that are synonyms or occur in collocations. When a co-occurrence matrix is constructed,

containing the co-occurrence of every examined pair of elements, it is possible to visualize the matrix as a network, where nodes are examined elements, e.g., words, and edges that are created when the co-occurrence is above zero. In this article, co-occurrence is used for binary data, and we define it as the number of positions where both vectors in a pair have the value 1:

$$C(\vec{x}, \vec{y}) = \sum_{i=1}^{n} x_i y_i \tag{9}$$

where x_i, y_i are the values of the feature *i* in the object vectors $\vec{x}$ and $\vec{y}$ and *n* is the number of features.

Jaccard index (Costa 2011) is another similarity measurement that can be adapted to work with vectors and matrices. The value ranges from 0 to 1. The original equation works with sets, and we can modify it to allow calculation between numerical vectors with the same dimension (Ruzicka similarity). The definition is in Equation 10.

$$J_v = \frac{\sum_{i=1}^{N} min(x_i, y_i)}{\sum_{i=1}^{N} max(x_i, y_i)} \tag{10}$$

where *n* is the number of features and x_i (y_i) are the values of the vectors $\vec{x}, \vec{y}$ at position *i*. It is especially useful when working with binary vectors. The index is popular because of its simplicity and low computational time.

Gower similarity (Gower 1871) can be used to measure similarity between objects described by mixed attribute types (numeric, binary, and categorical). Given two vectors $\vec{x}$ and $\vec{y}$ with *n* attributes, the Gower similarity is defined as

$$s(\vec{x}, \vec{y}) = \frac{\sum_{k=1}^{n} s(\vec{x}, \vec{y})_k \, w_k}{\sum_{k=1}^{n} w_k}, \tag{11}$$

where w_k is a weight (typically $w_k = 1$ if both values are not missing, and 0 otherwise), and $s(\vec{x}, \vec{y})_k$ is the partial similarity for attribute *k*, defined as

$$s(\vec{x}, \vec{y})_k = \begin{cases} 1 - \dfrac{|\vec{x}_k - \vec{y}_k|}{R_k}, & \text{if } k \text{ is a feature, with range } R_k, \\ 1, & \text{if } k \text{ is a binary feature or label } \vec{x}_k = \vec{y}_k, \\ 0, & \text{if } k \text{ is a binary feature or label } \vec{x}_k \neq \vec{y}_k. \end{cases} \tag{12}$$

ϵ-Radius and k-NN network construction

The ϵ-network is an undirected network where the set of edges *E* contains pairs of objects (o_i, o_j), if the calculated value of a chosen similarity $\rho(o_i, o_j)$ does not exceed the established threshold. Many efficient algorithms have been developed to determine the optimal value of the threshold ϵ (Bentley et al. 1977; Chazelle 1983). However, this type of construction easily leads to the formation of disjoint components. Therefore, it is very challenging to find a threshold value that results in a network that contains a satisfactory number of edges (Chen et al. 2009).

The network k-NN or *k-Neareast Neighbours* is generally an oriented network where an edge between objects o_i and o_j is created when the calculated value of a chosen similarity $\rho(o_i, o_j)$ is among the *k* smallest values of the set $\{\rho(o_i, o_k) | k = 1, \ldots, i-1, i+1, \ldots, n\}$. This type of construction has proven to be

much more efficient in practice than the construction method mentioned above, and therefore a number of research studies have already been conducted to improve this method (Dong et al. 2011; Chen et al. 2009). The advantage of this method is that we can set the minimum output degree of the network by k and more easily minimize the appearance of disjoint components (Chen et al. 2009).

One thing that might be appealing is that these two algorithms can be combined. This gives an analyst control over many hyperparameters, which can be tuned to find the most suitable outcome. The individual steps of the combined algorithm are as follows:

1. Calculate a similarity matrix S for a set of objects O.
2. Create a set of nodes V of the network N where node v_i represents the object o_i from the set of objects O.
3. Create a set of edges E of network N where E contains an edge e_{ij} between nodes v_i and v_j $(i \neq j)$ if v_j is among the k nearest neighbors of v_i or the calculated value of the chosen similarity does not exceed the established threshold ϵ.

LRNet

The following algorithm for the construction of weighted networks was published in Ochodkova et al. (2017). It is assumed that for each data object, a representativeness can be calculated, which is a local property based on the number of objects that are nearest neighbors of the selected object. Edges are established between pairs of nearest neighbors and between individual objects in numbers corresponding to the representativeness of these objects.

The algorithm LRNet consists of three steps:

1. Calculate a similarity matrix S for set of objects O.
2. Create a set of nodes V of network N where node v_i represents the object o_i from set of objects O.
3. Create a set of edges E of network N where E contains an edge e_{ij} between nodes v_i and v_j $(i \neq j)$ if o_j is the nearest neighbor ofo_i or o_j is a representative neighbor of o_i.

Network layout

In order to allow visual analysis of the network, we need a clear and understandable layout. The most common way to visualize a network is by using a node–link diagram, also known as a graph.

The ForceAtlas2 (Jacomy et al. 2014) algorithm is a force-directed layout designed to visualize complex networks, particularly in large-scale networks. It operates by simulating a physical system where nodes repel each other like charged particles, while edges act as springs that attract connected nodes, creating an intuitive spatial representation of network structure. ForceAtlas2 is particularly effective in revealing clusters in networks, making it widely used in social network analysis, biological networks, and patient similarity networks. Its adaptive speed optimizations and ability to preserve local structures while emphasizing global patterns make it a preferred choice for exploring relationships in complex datasets.

Cluster detection

The most common task we encountered when analyzing PSN with biomedical researchers was to detect clusters in relation to classes. Generally, we consider the group of nodes a potential cluster if the nodes are densely connected inside the group and loosely connected to the rest of the network. In the perfect scenario, each cluster would contain only nodes with one concrete class. When constructing a network, we focus on finding the suitable combination of features and construction processes that produces a cluster structure as close to perfect as possible, or at least to separate some classes.

Louvain algorithm

The Louvain algorithm (Blondel et al. 2008) was designed to divide a large network into partitions with high modularity in a short time and create a complete cluster hierarchy for the input network, thus allowing a choice of how the network will be divided. In the beginning, each node is assigned to its own cluster. The algorithm consists of two steps that are iteratively repeated until modularity no longer increases or the desired number of clusters is obtained:

1. Nodes are moved to neighboring clusters if it increases modularity. The neighbor with highest modularity gain is chosen.
2. The clusters are collapsed into single nodes, forming a new, smaller network.

Modularity

The modularity is a quality function based on the idea that we do not expect a random network to have a cluster structure. By comparing the density of a subgraph of a cluster with the density of the same set of nodes but with randomly connected edges, we can determine whether the cluster network can be considered dense or whether its connectivity is random. For cluster *C*, it is calculated as:

$$Q_c = \frac{L_c}{L} - \left(\frac{k_c}{2L}\right)^2 \tag{13}$$

where L_c is the number of edges in the cluster *C*, *L* the total number of edges in the network, k_c the total degree of nodes in *C*. If $Q_c > 0$, then we can consider it as a potential cluster. If Q_c is zero, then the connectivity between the nodes in *C* is random, based entirely on the degree distribution. Finally, if Q_c is negative, the nodes of *C* do not form a cluster.

Silhouette index

The Silhouette index is a metric used to evaluate the quality of clustering results by measuring how similar a data point is to its assigned cluster compared to other clusters. It is calculated for each data point and then averaged over the dataset to obtain an overall clustering score. We modified the original formula to allow for the use of similarity instead of dissimilarity (Rousseeuw 1987). For a given point *i*, the silhouette score is defined as:

$$S(i) = \frac{a(i) - b(i)}{max\{a(i), b(i)\}} \tag{14}$$

where $a(i)$ is the average similarity between i and all other points in the same cluster, and $b(i)$ is the highest average similarity between i and points in any other cluster (i.e., the most similar neighboring cluster). The silhouette score ranges from -1 to $+1$, where values close to $+1$ indicate well-clustered points, values around 0 suggest overlapping clusters, and values near -1 indicate incorrect clustering.

Matthews correlation coefficient

The Matthews correlation coefficient (MCC) (Matthews 1975) is a robust performance metric for binary classification that provides a balanced evaluation, even when class distributions are imbalanced. Unlike accuracy, which can be misleading in skewed datasets, MCC considers all four elements of the confusion matrix: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). We modified the definition of confusion matrix elements, which allows us to use the coefficient to describe the relationship between a class l and a cluster C in the network. It is calculated as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FN)}} \tag{15}$$

MCC provides values between -1 and $+1$, where $+1$ indicates perfect class coverage by a cluster, 0 corresponds to the random distribution of a class in a cluster, and -1 represents no relationship at all. TP is the number of nodes with class l in cluster C, TN the number of nodes without l outside of C, FP the number of nodes without l in C and FN the number of nodes with l outside of C.

SimNetX user activities

To achieve the goals mentioned earlier, the tool integrates several functionalities that support clinicians in managing and analyzing datasets. These functionalities are designed to provide user-friendly methods for data exploration, transformation, and visualization, allowing effective data analysis without requiring advanced technical expertise. The showcase of the tool user interface is shown in Fig. 2.

Activity 1: data import and setup for network construction

To start an analysis, the vector data set must be loaded into the tool. Once the data file is processed, the default network is constructed and visualized based on the initial construction method. An input data file needs to be a text file with attributes separated by a single unique character. The correct processing of the data file is ensured by stating the presence of column headers, a separator character, and a string representing missing data, which are automatically detected by our tool and handled appropriately, either by replacing missing values with the mean for continuous data or omitting them from calculations (e.g. Gower similarity), ensuring robust and reliable results. The second group is optional and is mainly used to specify the construction method for the initial network.

Activity 2: refining network structure through data transformation and filtering

The data may be in a form that is not suitable for some construction methods. Therefore, after loading the vector data and plotting the network, it is possible in such cases to visually inspect the distribution of the data using a histogram and apply various transformations to it. Data treated in this way can fundamentally alter the resulting shape of the

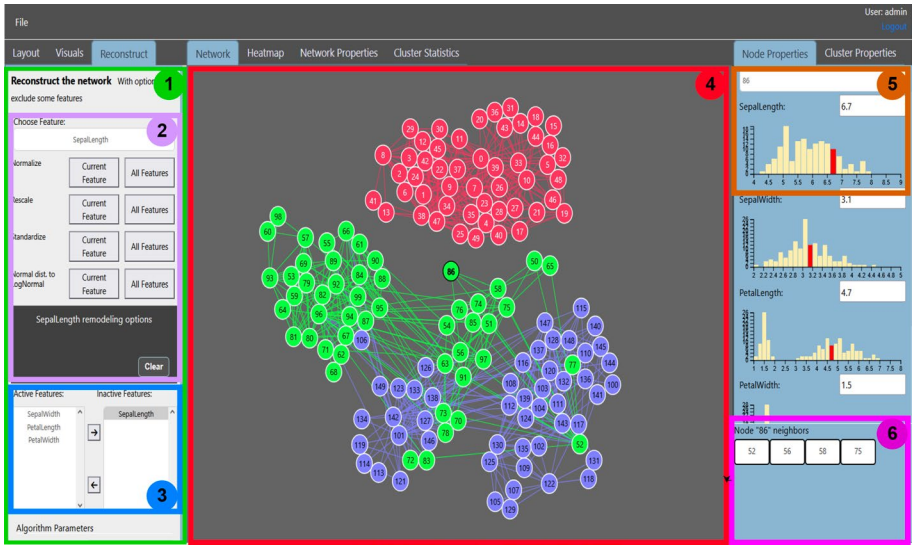

Fig. 2 The tool's GUI overview: 1. Layout, coloring, reconstruction settings panel; 2. Feature transformation options; 3. List of selected features used for network construction; 4. Network canvas; 5. Feature histogram and value for selected node (red bin shows the location of value of node 86 in the distribution); 6. Selected node's neighbor panel

network when reconstructed. It is also important to note that any special transformations or data-processing methods, not currently available in our tool, have to be applied before the data are loaded into it.

Features can be transformed by max normalization, standardization, rescaling to the interval $< 0, 1 >$, or by changing the log-normal distribution to normal. Multiple transformations can be applied at the same time and their order can be freely decided. The features altered in this way can significantly change the structure of the network when reconstructed.

Tools:

Transformations:

- Max normalization (Eq. 1)
- Minmax normalization (Eq. 2)
- Standardization (Eq. 3)
- Log-normal to normal distribution conversion (Eq. 4)

Activity 3: dynamic and customizable network construction

Once the data has been loaded and the default network visualized, the network construction process can then be repeated and customized by choosing a construction algorithm, a similarity measure, and a subset of features that we want to include in the calculation. The features included in the construction process are manually selected, typically based on domain knowledge. At this stage, we assume the tool will be applied to datasets containing on the order of tens of attributes, where manual selection remains feasible and effective. As each construction method and similarity measures have their strengths and weaknesses, different combinations of methods and features may yield different results with various levels of optimality. At the end of each reconstruction call, a network with a unique structure, which is based on chosen parameters, is produced, followed by an automatic Louvain cluster detection. The new network will then be drawn with nodes

colored by their respective cluster. Disabling features might be useful in situations where they do not have interesting patterns that would distinguish certain groups of patients or are simply unimportant in the eyes of a domain expert.

Tools:

Construction methods:

- LRNet
- ϵ-Radius & k-NN

Similarity measures:

- Gaussian Kernel (Eq. 5)
- Cosine similarity (Eq. 6)
- Absolute value of Pearson and Spearman Coefficient (Eq. 7, 8)
- Co-occurrence (Eq. 9)
- A modified Jaccard Index (Ruzicka similarity) for numerical data (Eq. 10)
- Gower similarity (Eq. 11)

Activity 4: interactive visualization for network exploration

Each constructed network is drawn using the ForceAtlas2 algorithm with the default force settings and node appearance. Although adjusting the forces can often improve visual clarity, the appearance of the nodes can be utilized to explore the distribution of features in relation to the structure of the network. Setting the color, size, or violet node label provides a quick way to identify potentially important features for specific parts of the network or to identify hubs.

Tools:

Network Layout:

- ForceAtlas2

Network Layout Forces:

- Center
- Node charge
- Node collision
- Link spring force
- Position forces

Visual Settings:

- Node coloring by feature values, labels or clusters
- Node size and labels by feature values

Activity 5: statistical exploration of clusters and features

In addition to inspecting the features as a whole, they can be further inspected in relation to clusters. To allow statistical exploration, the network needs to be separated into algorithmically detected or label-based groups. The tool offers basic statistical plots for

cluster analysis. These plots can be utilized to potentially find out how well a cluster covers a class and which feature led to the specific cluster structure.

Tools:

- Silhouette score barplot (Eq. 14)
- Matthews correlation coefficient barplot (Eq.15)
- Boxplots of all features in each cluster
- Boxplots of one feature in each cluster
- Cluster-specific histograms

Activity 6: capturing network states for future exploration

During data exploration, there can be cases of network configuration, which might potentially look interesting or require further investigation. The system provides the ability to save the state of the analysis as a file. The state includes all vector data, the network with its layout, visual settings, and unfiltered nodes, and all existing clusters. The state is stored in a json file and can be loaded back into the tool at any time. This allows the creation of network configuration snapshots without the need to remember them. Additionally, it is possible to export the original data together with newly added transformed features and detected clusters back into a CSV file, enabling use in analysis with other tools.

Figure 3 shows the activities associated with the mentioned observations; the arrows show the configurability and repeatability of the experiments. Exploratory analysis in the first step is related to the pre-analysis provided by the clinicians and laboratory staff.

Experiments

We demonstrate how our tool can be utilized to analyze a patient dataset and to possibly form an initial hypothesis that can be further investigated. We decided to conduct experiments on two well-known datasets Iris and Ecoli, as well as the RA patient dataset. The most usual patient analysis goal we encountered is to construct a network that would lead to the most optimal separation of all or at least some classes. Once we find the best separation, we can also study the feature values and distribution in each cluster.

We start the analysis by constructing a default network for each dataset. To find the optimally separated network, we first checked the distribution of each feature to decide whether they can be removed to reduce noise or have to be normalized in order to improve further reconstruction calculations. Then we repeatedly reconstruct the network using the different combinations of remaining features, construction algorithm, and similarity. After each reconstruction, the Louvain cluster detection is automatically called.

The general goal of analysis with our tool is to construct a network in which the detected clusters achieve high silhouette scores, exhibit the strongest possible correspondence with ground truth classes as measured by the Matthews correlation coefficient (MCC), and are characterized by features whose distributions are distinctively associated with one or more clusters. Nevertheless, we recognize that even when these quantitative criteria are satisfied, the resulting clusters still require clinical validation, as their practical relevance may not always align with domain experts' assessments.

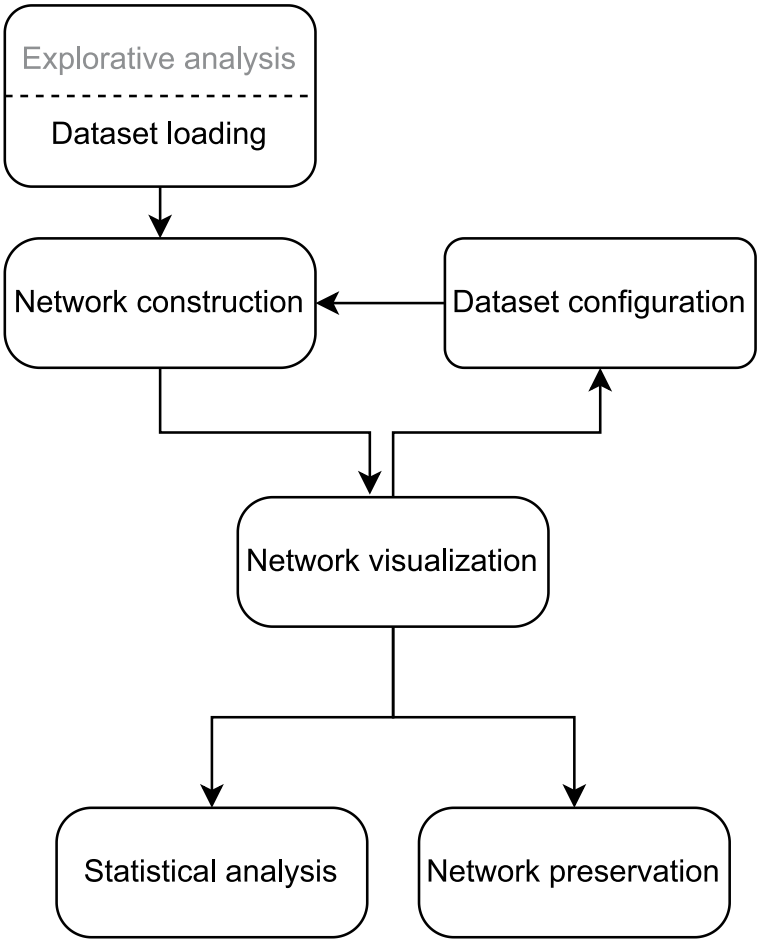

Fig. 3 Overview of observed activities

Table 2 Overview of the basic properties of datasets

Data	No. of samples	No. of features	No. of labels	No. of classes	Proportions of classes in %
Iris	150	4	1	3	Each class contains just one third of the samples
Ecoli	336	7	1	8	42.56, 22.92, 15.48, 10.42, 5.95, 1.49, 0.59, 0.59
RA	76	123	1	4	32.89, 15.79, 30.26, 21.06
AML	125	13	1	3	62.4, 28.8, 8.8

Data

We present the application on both patient datasets (Rheumatoid Arthritis patients and Acute Myeloid Leukemia patients), and on two well-known datasets Iris and Ecoli. The parameters of all data sets are summarized in Table 2. The last two datasets provide standardized and well-studied resources for testing new methods and algorithms in machine learning and bioinformatics.

The Iris dataset, with data on three species of irises, comes from measurements by Anderson (1936) and was first used and analyzed by British statistician Ronald Fisher in Fisher (1936). The dataset consists of 50 samples from each of the three species of irises, with four traits measured for each sample: sepal and petal length and width in centimeters. The Iris dataset is used primarily in machine learning for classification tasks.

Although its use in the context of network science is not common, it is used in methods to construct networks from vector data and to test and evaluate these methods (Parmezan et al. 2021), as benchmark data for cluster detection (Arruda et al. 2012; Taştan et al. 2021) or for clustering with a network science approach (Armano and Javarone 2013).

The Ecoli dataset containing information on protein localization in Escherichia coli cells was used in Horton and Nakai (1996), where a probabilistic classification system was presented. The dataset contains 336 samples with eight attributes (7 features, and one label representing classes) describing different protein characteristics such as signal sequence recognition methods and amino acid content discriminant analysis scores. The Ecoli dataset is used, for example, to develop and test algorithms to predict the cellular localization of proteins in Escherichia coli (which is key to understanding their functions) and to evaluate the performance of various classification methods in bioinformatics.

A typical use of this dataset is in interaction networks (protein-protein, gene-gene interactions). The two applications show how the Ecoli dataset can be used in network science to understand complex biological systems by analyzing their network structure and interactions (Kim et al. 2015; Mao et al. 2022). We used this dataset in Ochodkova et al. (2017) to introduce our LRNet method.

The third dataset that represents the data of patients with Rheumatoid Arthritis (RA) 76 consists of clinical data, protein data, and blood count. The dataset contains a total of 124 attributes, of which one is a label representing classes, and the rest are features. The original experiment with the data is described in Janca et al. (2023).

The last dataset is a set of patients with AML (Acute Myeloid Leukemia) sourced from TCGA (The Cancer Genome Atlas). More specifically, we work with preprocessed data from Rappoport and Shamir (2018). This dataset originally includes 200 patients and 98 attributes. After consulting with a domain expert, we selected 15 attributes from it, one representing patient ID, one representing label, and another 13 representing features. The final number of patients, 125, does not include patients with incomplete classes.

Iris

Using the Iris dataset, we want to demonstrate how excluding a particular important feature from the network construction process may leads to a patient similarity network with poorer clustering quality, as reflected by lower silhouette scores, and weaker correspondence with the ground truth classes, as measured by the Matthews correlation coefficient.

The network with the best separation and class coverage was acquired using LRNet and the Gaussian kernel with $\sigma = 0.2$ (Fig. 4). The network was divided into four detected clusters. Inspecting the silhouette plot, we can see that most of the nodes in each cluster have a strong affiliation to their clusters. The isolated red cluster perfectly covers the class *Setosa*, containing all nodes of this class and none of the nodes of any other class. The teal cluster has the worst MCC score and contains the nodes that are the most difficult to classify, with one class only slightly more dominant than the others. The boxplots for each feature in each cluster show us how each feature contributes to this specific network separation, with the red cluster having very distinct values for each feature.

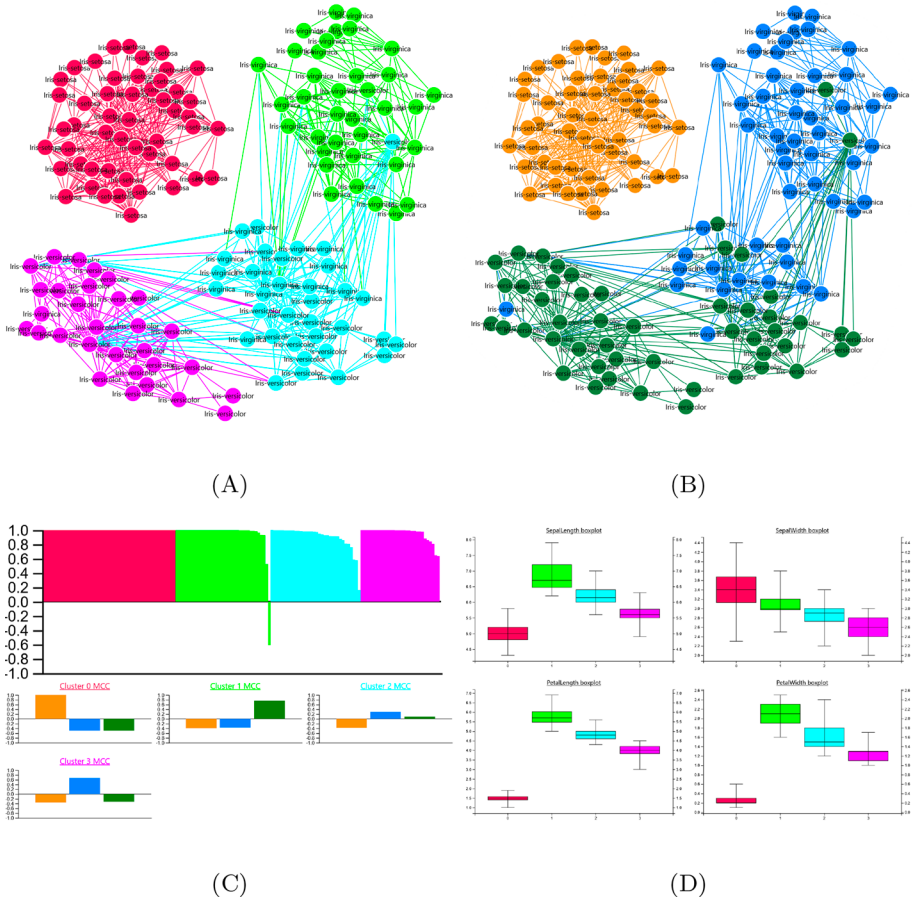

Fig. 4 Iris: network with best separation; **A** Constructed network with detected clusters; **B** Network colored by classes (orange—Setosa, light blue—Virginica, dark green—Versicolor); **C** Silhouette and MCC plots; **D** Boxplots in each cluster for every feature

The features of the Iris network were left unchanged. Each attempt to exclude any subset of features from the network construction process resulted in lower silhouettes and MCCs. An example is shown with feature *Petal Length* in Fig. 5. We can confirm the importance of this feature for distinguishing classes of iris plant by inspecting network gradient colored by values, and comparing silhouettes and MCCs of the worse network to the best we found.

Ecoli

The experiment with the Ecoli dataset demonstrates that the best clustering result can be achieved using only three of the seven available features. In this case, each of the selected features shows a distinct distribution within a single cluster, making all three crucial for producing the final multivariate clustering structure. The features that were used for the calculation of the presumably best network were: *gvh*, *aac*, *alm1*. The network with the best silhouettes and MCCs together with statistical plots is shown in Fig. 6. *LRNet* was used as a construction algorithm with two minimum neighbors per node and Gaussian similarity with $\sigma = 0.2$.

Three clusters (red, green, yellow) were the best according to MCC. Each of these clusters had a single dominant class with decent scores, which means that, to some extent, we were able to distinguish the nodes with these classes from the rest.

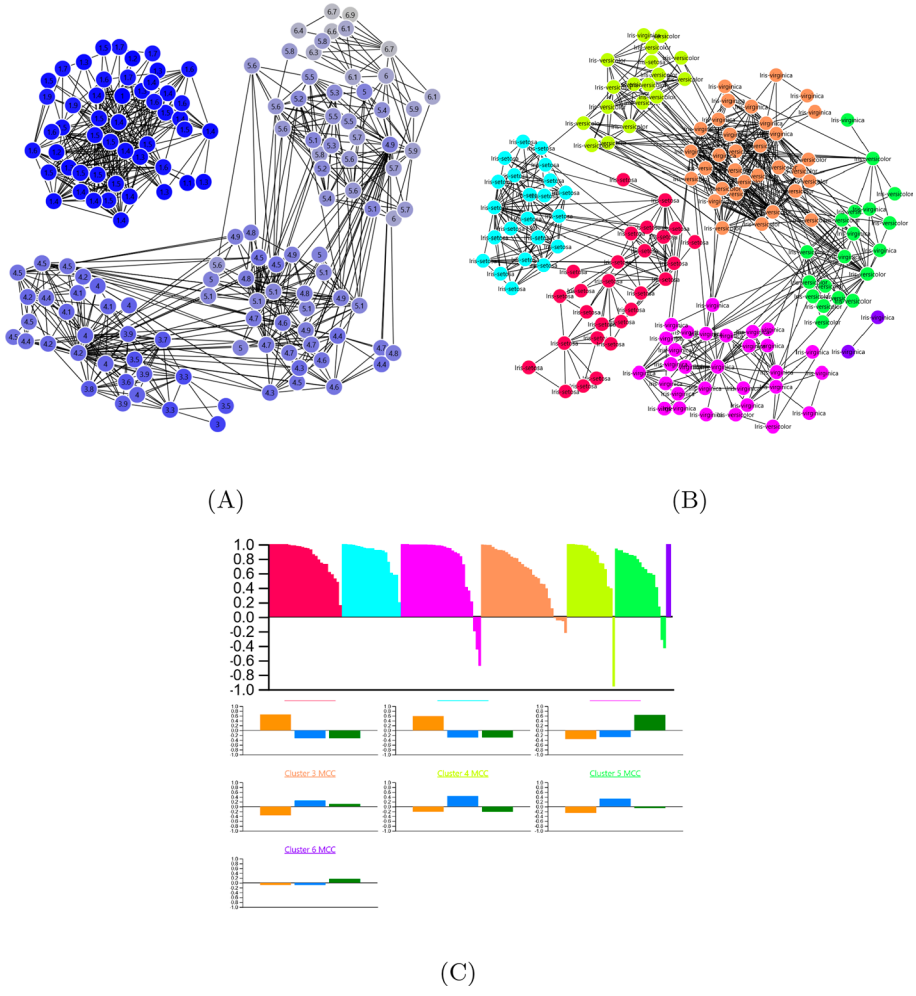

Fig. 5 Iris: network with *Petal Length* excluded from calculation; **A** The distribution of *Petal Length* on a dark-to-light scale shown before reconstruction in the best network; **B** Newly constructed network without *Petal Length* with detected clusters; **C** Silhouette and MCC scores

The teal cluster is made up of nodes that are presumably the most difficult to classify, as this cluster has a mix of many distinct classes. The pink cluster almost exclusively contains proteins located in the inner membrane, with *im* and *imU* being the most dominant. It appears to be challenging to distinguish between them, probably because they seem very similar. It is also important to note that the classes of this dataset are heavily imbalanced, which usually makes classification more difficult, simply because there is not enough data to reliably distinguish between them.

The network was constructed for each combination of three or more features. The decision on which features to keep in the construction process was made on the basis of silhouettes and MCCs. The goal was to find the network with the best MCC while maintaining high Silhouette scores. The features that were excluded from the construction process of the best-scoring network were: *mcg* (continuous), *lip* (continuous with single dominant value), *chg* (continuous with single dominant value), and *alm2* (continuous). The histograms for these features are shown in Fig. 7. The feature *chg* was excluded because all nodes shared the same value, except for one outlier. Adding or removing it had no effect on the network structure. The rest of the features negatively affected MCC

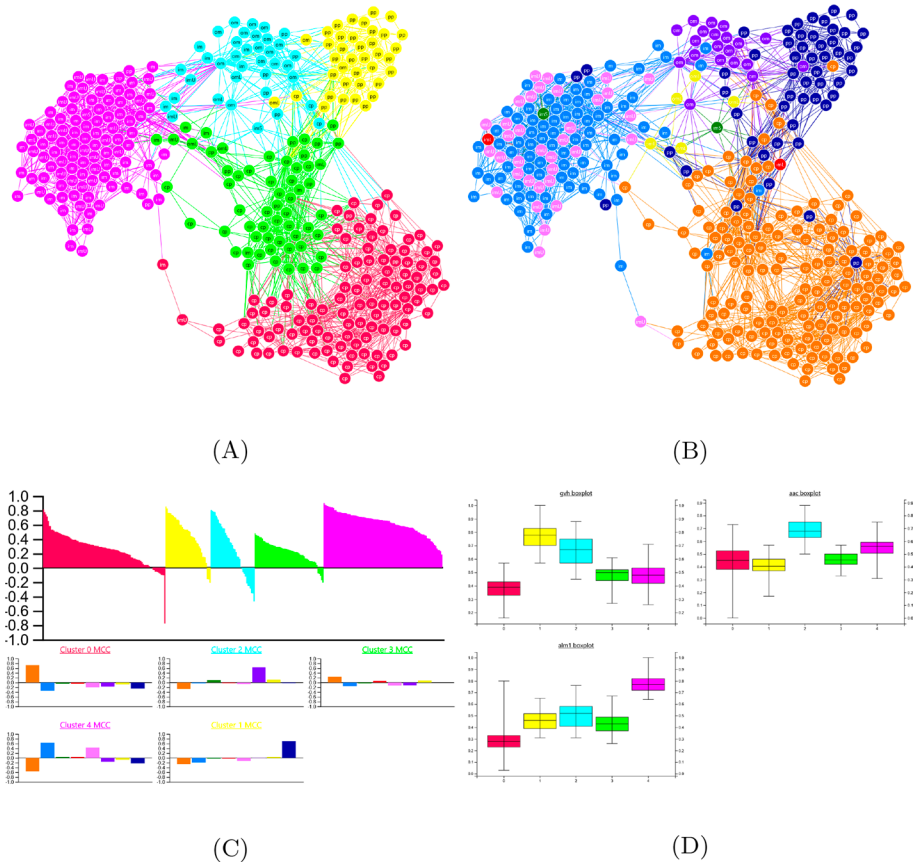

Fig. 6 Ecoli: network with best separation; **A** Constructed network with detected clusters; **B** Network colored by classes (orange—cp, light blue—im, dark green—imS, red—imL, pink—imU, violet—om, yellow—omL, dark blue—pp); **C** Silhouette and MCC plots; **D** Boxplots for features in a each cluster

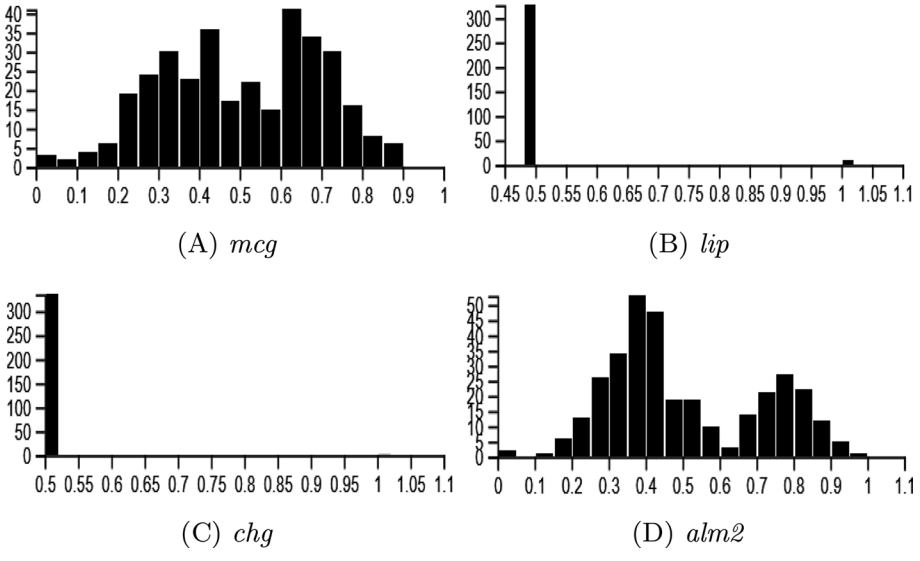

Fig. 7 Ecoli: Histograms of excluded features

or led to the network with a larger number of detected clusters with high silhouettes but low MCC.

RA patients

Next, we demonstrate the analysis process on a real-world patient dataset, which contains information about patients suffering from rheumatoid arthritis (RA). The goal is to show, when analyzing real patient data, that there are often clusters containing patients of mixed classes, being very difficult to separate based on the available features. What is clinically significant, however, is that we can usually still identify one or more clusters with a strong association to a specific class, along with the features driving that relationship.

The data set contains a large number of features; therefore, we used selected features in the final PSN of the original article as a starting point with the goal of finding the network with best-class coverage. The classes in this data set represent four groups of patients with distinct disease activity and treatment strategies. We began by adding or removing features while watching the number of clusters and any increases in MCC and silhouette scores. The greatest positive gain was observed when we added feature *SDAI_AToS* to the calculation. Most of the patients with higher disease activity had higher values of the feature (Fig. 8).

The final PSN, shown in Fig. 9, was constructed from 7 features: *cholesterol*, *triglycerides*, *CRP*, *LDL*, *SDAI_AToS*, *CCL8 sPDL1*. The features were normalized for the network construction process. Excluding any of the seven features negatively affected the MCC scores. We expected four clusters to be detected in the network, as can be seen in the patient similarities heat map (Fig. 9c) as larger diagonal blocks.

Eventually, four clusters were detected in the network, the same as the number of classes. The cluster statistics with feature boxplots are shown in Fig. 10. Each cluster covers one class better than the others to a certain degree. The green cluster contains the most patients with the highest disease activity and the best MCC score, while the orange cluster was on the opposite side of the network and had the most patients with low activity. Class 2 is the most difficult to classify with its highest MCC score reached still being

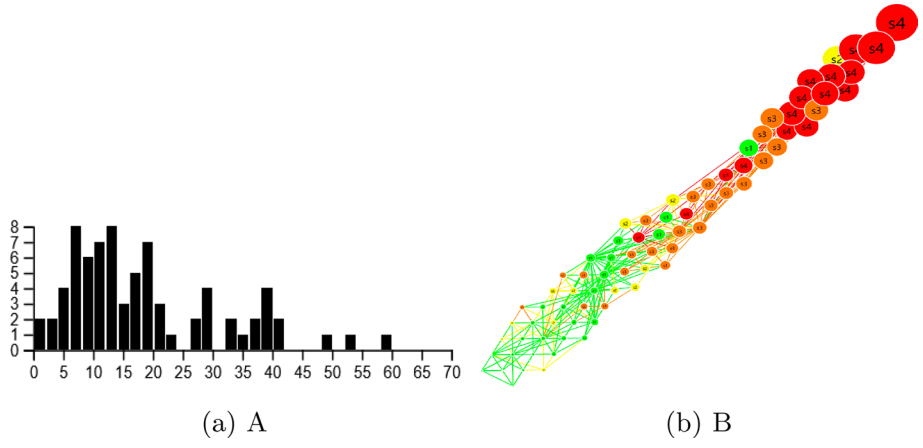

Fig. 8 Patient: distribution of feature *SDAI_AToS*; **A** feature histogram, **B** Patient nodes with fixed positions based on *SDAI_AToS* feature values. The value of the feature increases from the bottom left corner to the top right corner, and with the node size. Non-active patients are colored green (class 1) and yellow (class 2). Active patients are colored orange (class 3) and red (class 4)

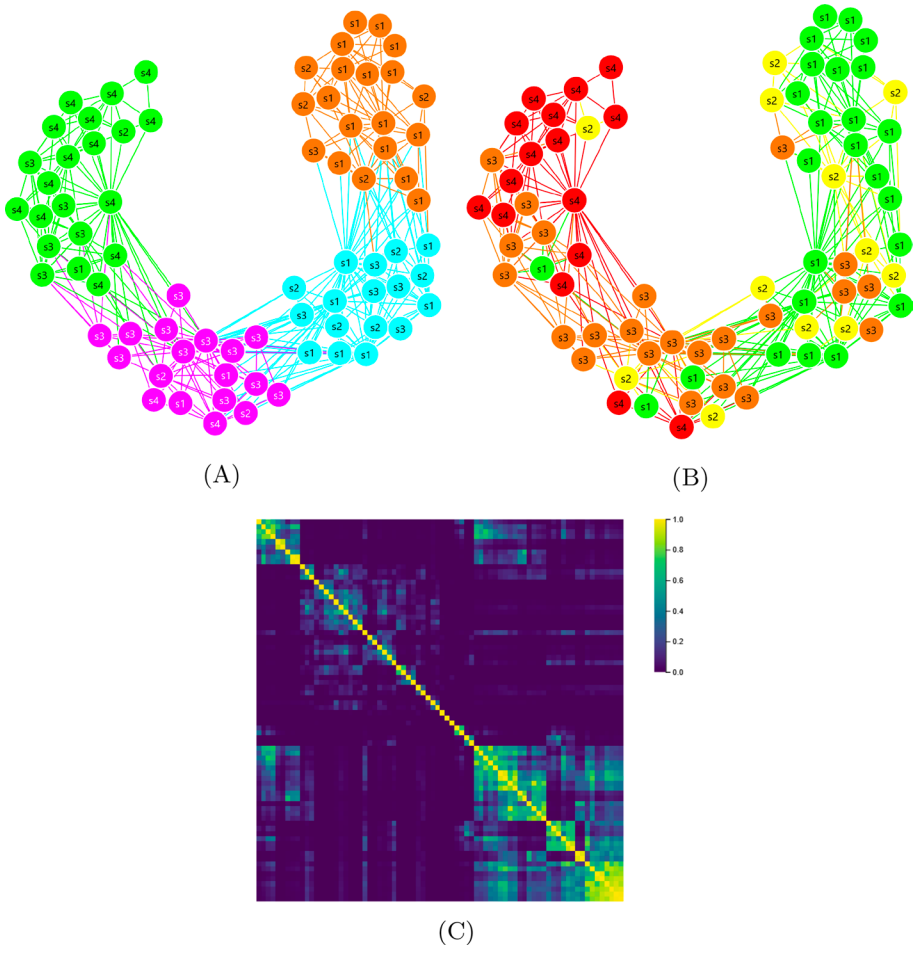

Fig. 9 RA Patients: network with best separation; **A** Constructed network with detected clusters; **B** Network colored by classes. Non-active patients are colored green (class 1) and yellow (class 2). Active patients are colored orange (class 3) and red (class 4); **C** Heatmap of patient similarities

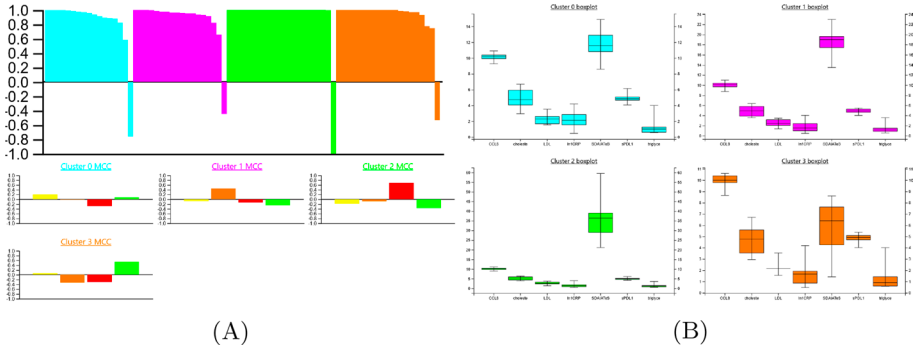

Fig. 10 RA Patients: cluster statistics; **A** Silhouette and MCC plots; **B** Boxplots for each feature in a specific cluster

low in the teal cluster. It appears to be difficult to make a decision about when a patient should begin to be monitored before disease activity increases. Clinicians could use the selected features as a basis for further investigation, which could potentially result in a better choice of treatment strategy.

AML patients

The last dataset consists of patients with Acute Myeloid Leukemia (AM)L. We want to demonstrate a similar situation as with the previous dataset, where its not always possible to find a relationship between each cluster and class. This dataset has proven to be especially difficult to analyze, with only one cluster showing potentially meaningful results.

From the original classes for the *FISH_test_component* class label, we reduced the dataset to only patients with three specific classes (after consulting with a domain expert). The classes represent biomarkers (recurrent genetic abnormalities) that define diagnostic subtypes of AML, namely *BCR-ABL*, *PML-RAR*, and a combination of both. The proportions of classes in the dataset can be found in Table 2. For the construction process, we worked only with laboratory parameters, and the final network, see Fig. 11, was constructed using Gower similarity and a subset of five features (see plot titles in Fig. 12) with two transformed by logarithmization due to their distribution. The features were not normalized.

Based on the statistics (Fig. 12), the green class (*BCR-ABL*) has a higher MCC in the orange cluster because most patients with this biomarker are in that cluster and the proportion of patients with other classes is small (relative to all patients). The separation of the orange cluster with the highest representation of *BCR-ABL* is most likely caused by higher values of the second parameter (*bone_marrow_band_cell_result_percent_value*). This cluster is impure because a small proportion of patients from other classes in that cluster also have a high value for that parameter.

In the red cluster, *PML-RAR* class predominates and shows higher MCC, but it contains only a small fraction of the total samples in this class. Its drawback is that the first and third features span a wide range of values, so the internal consistency is not very strong.

The last feature appears to have small distribution differences between clusters, yet removing it worsened the result. This shows that even features that seem insignificant might be important.

Examining patients with blue and orange classes in the orange cluster may help clinicians understand why they were initially grouped with the green class. The results are not statistically convincing, but even in such cases, they often prove useful for clinical decision-making.

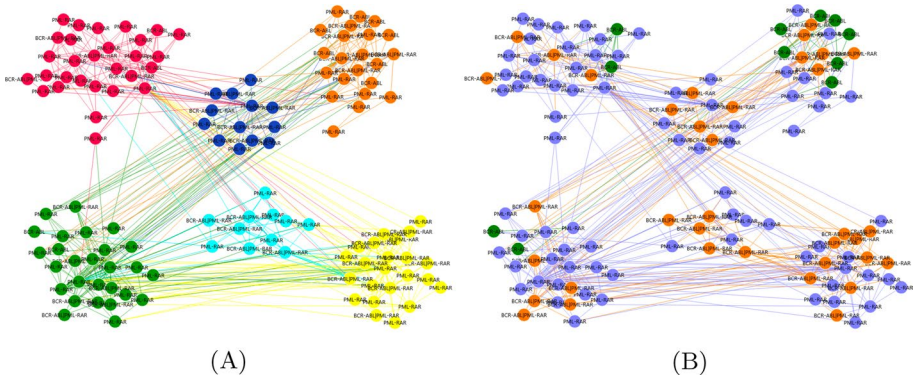

Fig. 11 AML Patients: network with best separation; **A** Constructed network with detected clusters; **B** Network colored by classes. *BCR-ABL* class is colored green. *PML-RAR* is blue colored and *BCR-ABL/PML-RAR* is orange

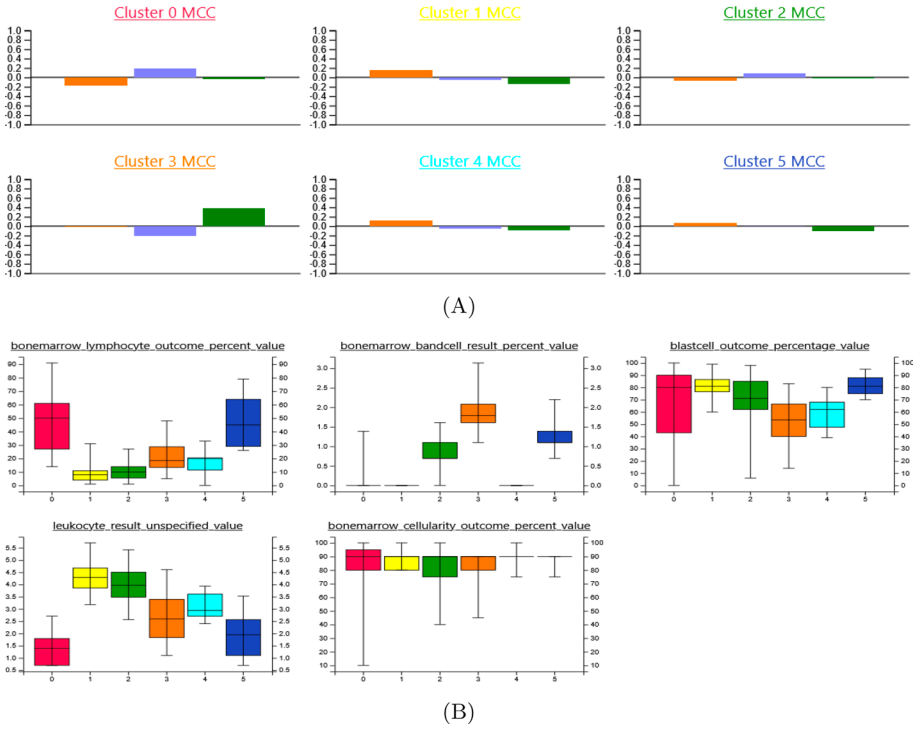

Fig. 12 AML Patients: cluster statistics; **A** MCC plots; **B** Boxplots for features in a each cluster

Conclusion

In this article, we have explored the use of patient similarity networks in biomedical data analysis. To the best of our knowledge, no interactive software tool currently supports both the construction of networks from vector data and the real-time visualization necessary for effective interpretation. The tool we present, SimNetX, was developed to address this gap and is the result of several years of collaboration within a multidisciplinary biomedical research team. Its primary advantage lies in its interactivity and no-code approach, enabling researchers without advanced technical knowledge to analyze their data efficiently.

So far, the feedback from clinicians and researchers has been positive, reinforcing the potential of the tool to facilitate the exploration of biomedical data. Moving forward, we plan to further refine SimNetX based on user input and evolving research needs. In its current version, users control most of the data processing and configuration settings, which means that results may not always be immediately useful without prior domain expertise. A key challenge for future development will be the integration of automated or semi-automated feature selection and network generation, allowing users to more efficiently identify networks with meaningful analytical and clinical insights.

In future developments, we may also consider extending our framework to support multilayer networks, enabling the integration of multi-modal data within a unified representation of patient similarity. This would allow us to combine heterogeneous sources of information and exploit complementary signals across modalities. In this context, it would also be valuable to incorporate approaches for PSN integration, such as Similarity Network Fusion (SNF) Wang et al. (2014) and NEMO (Rappoport and Shamir 2019), which provide state-of-the-art strategies to generate integrated patient similarity networks from diverse data types.

Author contributions
T.A. contributed to all sections and conducted the experimental part; K.K. prepared figures 1–2 and contributed to section 1, 2 and 6; E.O. contributed to section 5 and reviewed the manuscript text, and M.K. contributed to section 4 and reviewed the manuscript text; E.K. provided testing data and constructive feedback in the development of the tool described in the manuscript.

Funding
This work was supported by SGS, VŠB-Technical University Ostrava, under grant no. SP2025/018.

Data availability
Iris data that are used for first experiments of this study have been deposited in the UC Irvine Machine Learning Repository at https://archive.ics.uci.edu/dataset/53/iris. Ecoli data that are used for first experiments of this study have been deposited in the UC Irvine Machine Learning Repository at https://archive.ics.uci.edu/dataset/39/ecoli. RA patient data that are used for third experiments of this study are available from the original author upon reasonable request.

Materials availability
Not applicable.

Code availability
The code for SimNetX is available at https://github.com/Anim64/SimNetX.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no Conflict of interest.

Received: 1 April 2025 / Accepted: 26 September 2025
Published online: 31 October 2025

References

Allegri SA, McCoy K, Mitchell CS (2022) Compositeview: a network-based visualization tool. Big Data Cogn Comput 6(2):66
Anderson E (1936) The species problem in iris. Ann Mo Bot Gard 23(3):457–509
Anlauf T, Kubikova K, Ochodkova E, Kriegova E, Kudelka M (2024) SimNetX: interactive support for biomedical data analysis using patient similarity networks. In: International conference on complex networks and their applications. Springer, pp 3–14
Armano G, Javarone MA (2013) Clustering datasets by complex networks analysis. Complex Adapt Syst Model 1:1–10
Arruda GF, Fontoura Costa L, Rodrigues FA (2012) A complex networks approach for data clustering. XXPhys A 391(23):6174–6183
Auer F, Mayer S, Kramer F (2022) MetaRelSubNetVis: referenceable network visualizations based on integrated patient data with group-wise comparison. bioRxiv
Bentley JL, Stanat DF, Williams EH Jr (1977) The complexity of finding fixed-radius near neighbors. Inf Process Lett 6(6):209–212
Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):10008
Chazelle B (1983) An improved algorithm for the fixed-radius neighbor problem. Inf Process Lett 16(4):193–198
Chen J, Fang H-R, Saad Y (2009) Fast approximate KNN graph construction for high dimensional data via recursive Lanczos bisection. J Mach Learn Res 10(Sep):1989–2012
Costa LdF (2011) Further generalizations of the Jaccard index. Preprint at arXiv:2110.09619
Dong W, Moses C, Li K (2011) Efficient k-nearest neighbor graph construction for generic similarity measures. In: Proceedings of the 20th international conference on world wide web, pp 577–586
Elzen S, Wijk JJ (2014) Multivariate network exploration and presentation: from detail to overview via selections and aggregations. IEEE Trans Visual Comput Graph 20(12):2310–2319
Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7(2):179–188
Giudice L, Mohamed A, Malm T (2024) StellarPath: hierarchical-vertical multi-omics classifier synergizes stable markers and interpretable similarity networks for patient profiling. PLoS Comput Biol 20(4):1012022
Gliozzo J, Soto Gomez MA, Bonometti A, Patak A, Casiraghi E, Valentini G (2025) miss-SNF: a multimodal patient similarity network integration approach to handle completely missing data sources. Bioinformatics 41(4):150
Gliozzo J, Patak A, Gallardo AP, Casiraghi E, Valentini G (2023) Patient similarity networks integration for partial multimodal datasets. In: Bioinformatics, pp 228–234
Gower JC (1871) A general coefficient of similarity and some of its properties. Biometrics 857–871
Han J, Kamber M, Pei J (2011) Data mining: concepts and techniques third edition. The Morgan Kaufmann Series in Data Management Systems 5(4):83–124
Horton P, Nakai K (1996) A probabilistic classification system for predicting the cellular localization sites of proteins. In: Ismb. St. Louis, vol 4, pp 109–115
Huang H-z, Lu X-d, Guo W, Jiang X-b, Yan Z-m, Wang S-p (2021) Heterogeneous information network-based patient similarity search. Front Cell Dev Biol 9:735687
Jacomy M, Venturini T, Heymann S, Bastian M (2014) ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. PLoS ONE 9(6):98679

Janca O, Ochodkova E, Kriegova E, Horak P, Skacelova M, Kudelka M (2023) Real-world data in rheumatoid arthritis: patient similarity networks as a tool for clinical evaluation of disease activity. Appl Netw Sci 8(1):57
Kesimoglu ZN, Bozdag S (2022) Supreme: a cancer subtype prediction methodology integrating multiomics data using graph convolutional neural network. bioRxiv
Kim H, Shim JE, Shin J, Lee I (2015) EcoliNet: a database of cofunctional gene network for *Escherichia coli*. Database 2015:001
Li X, Ma J, Leng L, Han M, Li M, He F, Zhu Y (2022) MoGCN: a multi-omics integration method based on graph convolutional network for cancer subtype analysis. Front Genet 13:806842
Li H, Zhou M, Sun Y, Yang J, Zeng X, Qiu Y, Xia Y, Zheng Z, Yu J, Feng Y (2024) A patient similarity network (CHDmap) to predict outcomes after congenital heart surgery: development and validation study. JMIR Med Inform 12(1):49138
Liu Y, Zhang Z, Qin S, Salim FD, Bian J, Yepes AJ (2024) Fine-grained patient similarity measuring using contrastive graph similarity networks. In: 2024 IEEE 12th international conference on healthcare informatics (ICHI). IEEE, pp 1–10
Mao Z, Huang T, Yuan Q, Ma H (2022) Construction and analysis of an integrated biological network of *Escherichia coli*. Syst Microbiol Biomanuf 2(1):165–176
Matthews BW (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochimica et Biophysica Acta (BBA) Protein Struct 405(2):442–451
Mikulkova Z, Manukyan G, Turcsanyi P, Kudelka M, Urbanova R, Savara J, Ochodkova E, Brychtova Y, Molinsky J, Simkovic M (2021) Deciphering the complex circulating immune cell microenvironment in chronic lymphocytic leukaemia using patient similarity networks. Sci Rep 11(1):322
Navaz AN, T. El-Kassabi H, Serhani MA, Oulhaj A, Khalil K (2022) A novel patient similarity network (PSN) framework based on multi-model deep learning for precision medicine. J Pers Med 12(5):768
Ochodkova E, Zehnalova S, Kudelka M (2017) Graph construction based on local representativeness. In: International computing and combinatorics conference. Springer, pp 654–665
Pai S, Bader GD (2018) Patient similarity networks for precision medicine. J Mol Biol 430(18):2924–2938
Pai S, Hui S, Isserlin R, Shah MA, Kaka H, Bader GD (2019) Netdx: interpretable patient classification using integrated patient similarity networks. Mol Syst Biol 15(3):8497
Parmezan ARS, Lee HD, Spolaôr N, Wu FC (2021) Automatic recommendation of feature selection algorithms based on dataset characteristics. Expert Syst Appl 185:115589
Rappoport N, Shamir R (2018) Multi-omic and multi-view clustering algorithms: review and cancer benchmark. Nucleic Acids Res 46(20):10546–10562
Rappoport N, Shamir R (2019) Nemo: cancer subtyping by integration of partial multi-omic data. Bioinformatics 35(18):3348–3356
Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
Sova M, Kudelka M, Raska M, Mizera J, Mikulkova Z, Trajerova M, Ochodkova E, Genzor S, Jakubec P, Borikova A (2022) Network analysis for uncovering the relationship between host response and clinical factors to virus pathogen: lessons from SARS-CoV-2-2. Viruses 14(11):2422
Tanvir RB, Islam MM, Sobhan M, Luo D, Mondal AM (2024) MOGAT: a multi-omics integration framework using graph attention networks for cancer subtype prediction. Int J Mol Sci 25(5):2788
Taştan A, Muma M, Zoubir AM (2021) Sparsity-aware robust community detection (SPARCODE). Signal Process 187:108147
Wang B, Mezlini AM, Demir F, Fiume M, Tu Z, Brudno M, Haibe-Kains B, Goldenberg A (2014) Similarity network fusion for aggregating data types on a genomic scale. Nat Methods 11(3):333–337
Wang T, Shao W, Huang Z, Tang H, Zhang J, Ding Z, Huang K (2021) MOGONET integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. Nat Commun 12(1):3445
Wang Y, Wang Z, Yu X, Wang X, Song J, Yu D-J, Ge F (2024) More: a multi-omics data-driven hypergraph integration network for biomedical data classification and biomarker identification. Brief Bioinform 26(1)
Werner E, Clark JN, Hepburn A, Bhamber RS, Ambler M, Bourdeaux CP, McWilliams CJ, Santos-Rodriguez R (2023) Explainable hierarchical clustering for patient subtyping and risk prediction. Exp Biol Med 248(24):2547–2559
Wu J, Dong Y, Gao Z, Gong T, Li C (2023) Dual attention and patient similarity network for drug recommendation. Bioinformatics 39(1):003

Publisher's Note