PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
Escuela de Posgrado

Exploratory Analysis of Mass Spectrometry Data Based on Graph Embeddings

Tesis para obtener el grado académico de Doctor en Ingeniería que presenta:

Edwin Alvarez Mamani

Asesor:

Dr. Alfredo Jesús Ibáñez Gabilondo

Co-asesor:

Dr. César Armando Beltrán Castañón

Lima, 2025

Informe de Similitud

Yo, Alfredo Jesús Ibáñez Gabilondo, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada “Exploratory Analysis of Mass Spectrometry Data Base on Graph Embedding”, del autor Edwin Álvarez Mamani, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 81%. Así lo consigna el reporte de similitud emitido por el software Turnitin el 11/03/2025. (Este alto índice se debe a que el documento adopta el formato de tesis compuesta, integrando trabajos previamente publicados por el mismo alumno y presentados como capítulos en la tesis.).
- He revisado con detalle dicho reporte y la Tesis, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima, 11/03/2025

Apellidos y nombres del asesor / de la asesora: Alfredo Jesús Ibáñez Gabilondo	
DNI: 10802782	Firma:
ORCID: 0000-0001-9206-1537	

I would like to dedicate this thesis to my family, friends, and professors ...

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification at this or any other university. This thesis is my own work and does not contain anything that is the result of work done in collaboration with others, except as specified in the text and in the Acknowledgements.

Edwin Alvarez Mamani
Lima, 2025

Acknowledgements

I would like to extend my heartfelt gratitude to my supervisors, Dr. Alfredo J. Ibáñez Gabilondo and Dr. César A. Beltrán Castañón, for their invaluable guidance and support throughout this project. My sincere thanks also go to Florian Buettner (Goethe University) and Reinhard Dechant (Calico Life Science), for their insightful contributions and input.

Abstract

Mass spectrometry (MS)-based metabolomics analysis is a powerful tool, but it comes with its own set of challenges. The MS workflow involves multiple steps before its interpretation in what is denominate data mining. Data mining consists of a two-step process. First, the MS data is ordered, arranged, and presented for filtering before being analyzed. Second, the filtered and reduced data are analyzed using statistics to remove further variability. This holds true particularly for MS-based untargeted metabolomics studies, which focused on understanding fold changes in metabolic networks. Since the task of filtering and identifying changes from a large dataset is challenging, automated techniques for mining untargeted MS-based metabolomic data are needed. The traditional statistics-based approach tends to overfilter raw data, which may result in the removal of relevant data and lead to the identification of fewer metabolomic changes. This limitation of the traditional approach underscores the need for a new method. In this work, we present a novel deep learning approach using node embeddings (powered by Graph Neural Networks), edge embeddings, and anomaly detection algorithm to analyze the data generated by MS-based metabolomics called `GEMNA` (Graph Embedding-based Metabolomics Network Analysis), for example for an untargeted volatile study on Mentos candy, the data clusters produced by `GEMNA` were better than the ones used traditional tools, i.e., `GEMNA` has $silhouette\,score = 0.409$, *vs* the traditional approach has $silhouette\,score = -0.004$.

keywords: *Mass spectrometry, Metabolomic networks, Graph neural networks, Graph embeddings.*

Resumen

El análisis metabolómico basado en la espectrometría de masas (EM) es una herramienta poderosa, pero conlleva sus propios retos. El flujo de trabajo de la EM implica múltiples pasos antes de su interpretación, el cual típicamente se denomina minería de datos. La extracción de datos consiste en un proceso de dos pasos. Primero, los datos de la EM se ordenan, organizan y presentan para su filtrado antes de ser analizados. Segundo, los datos filtrados y reducidos se analizan utilizando técnicas estadísticas para eliminar más variabilidad. Esto es especialmente cierto en el caso de los estudios metabolómicos no dirigidos (*untargeted*) basados en EM, que se centran en comprender los cambios en las redes metabólicas. Dado que la tarea de filtrar e identificar cambios a partir de un gran conjunto de datos es un reto, se necesitan técnicas automatizadas para la minería de datos metabolómicos no dirigidos basados en MS. El enfoque tradicional basado en estadísticas tiende a filtrar en exceso los datos en bruto, lo que puede dar lugar a la eliminación de datos relevantes y conducir a la identificación de menos cambios metabolómicos. Esta limitación del enfoque tradicional subraya la necesidad de un nuevo método. En este trabajo, presentamos un nuevo enfoque de aprendizaje profundo que utiliza *node embeddings* (impulsado por *Graph Neural Networks*), *edge embeddings* y un algoritmo de detección de anomalías para analizar los datos generados por la metabolómica basada en EM llamado `GEMNA` (Graph Embedding-based Metabolomics Network Analysis), Por ejemplo, para un estudio de volatilidad no dirigida en caramelos Mentos, los grupos de datos producidos por `GEMNA` fueron mejores que los de las técnicas tradicionales, es decir, `GEMNA` consigue una $silhouette\,score = 0.409$, *vs* el enfoque tradicional que consigue una $silhouette\,score = -0.004$.

Palabras claves: *Espectrometría de masas, Redes metabolómicas, Graph neural networks, Graph embedding*.

Table of contents

List of figures

Nomenclature

Acronyms / Abbreviations

ARGVA Adversarially Regularized Variational Graph Auto-Encoder

CA Correlation Analysis

CE Capillary Electrophoresis

FAB Fast-atom bombardment

CI Chemical Ionization

COPOD Copula-Based Outlier Detection

DDA Data-Dependent Analysis

DGI Deep Graph Infomax

DIA data-independent acquisitio

ECOD Empirical Cumulative Distribution-based Outlier Detection

ESI Electrospray Ionization

GC Gas Chromatography

GEMNA Graph Embedding-based Metabolomics Network Analysis

GGNN Geometric Graph Neural Network

GNN Graph Neural Network

HCA Hierarchical Cluster Analysi

LC Liquid Chromatography

LVGAE Linear Graph Variational Autoencoder

MALDI Matrix-assisted Laser Desorption Ionization

MS Mass Spectrometry

NMR Nuclear Magnetic Resonance

PCA Principal Component Analysis

PLS-DA Partial Least Squares Discriminant Analysis

PLS-DA Projection to Latent Structure Discriminant Analysis

PLS-R Projection to Latent Structure Regression

VGAE Variational Graph Autoencoder

Chapter 1

Introduction

1.1 Problem description

In the field of health, scientific advances are important aspects for the development and well-being of mankind, possessing techniques for data processing and analysis are crucial for making decision. Chemistry and biology are fields that are closely related to health. Specifically the omics sciences, such as genomics, proteomics, metabolomics, etc. However, the processing and analysis of this type of data is very expensive due to its complexity and size. Within metabolomics, the data generated by the analytical technique of mass spectrometry (MS) are subject to several factors that can alter their environment, some of these are: electrical noise, chemical contamination, etc. Thus, in a metabolomics study, the influence of these factors causes the data to be mostly composed of noise (approximately 90%) [2, 10].

1.2 Problem statement

Metabolic networks composed of mass spectrometry data, by their nature, are complex and noisy data. Analyzing (filtering and identifying changes) these data in a controlled and efficient manner for interpretation and making decision still remains a challenge to overcome. Traditionally statistical-based techniques perform the filtering task very aggressively, and there is the possibility of eliminating data that could be valuable, as well as identifying less significant changes, which would affect interpretation.

1.3 Justification

Our proposal promises to help non-specialized MS researchers in chemistry and biology better understand/visualize their MS studies for decision-making and data interpretation. Moreover, in the case of an unknown sample or first-time measurement, these steps together represent approximately 90% of the time in an analytical laboratory. Hence, graph machine learning models plus anomaly detection algorithm could accelerate this process, providing life scientists and health researchers with a new toolbox.

1.4 Objectives

In this work, we present a novel deep learning approach using node embeddings (powered by GNNs), edge embeddings, and anomaly detection algorithm to analyze the data generated by mass spectrometry (MS)-based metabolomics called `GEMNA`, i.e., Graph Embedding-based Metabolomics Network Analysis. Although embeddings have been used previously in mass spectrometry, this is the first time they have been used for MS-based data filtration, and later identify changes in the filtered metabolic networks. Receiving as input the MS data obtained either by a flow injection MS or chromatography coupled-MS system, then, `GEMNA` identifies the "real" signals by using embedding filtration couple with a Graph Neural Network (GNN) model; and generating as output the filtered MS-based signal list and a dashboard with graphs showing the changes between metabolites among two or more samples. To achieve the main goal, the following specific objectives are established:

- Develop a pipeline for the analysis of metabolomic networks.
- Comparing the quality of node embeddings (generated by GNN models) for the analysis of metabolomic networks.
- Evaluate execution times for the analysis of metabolomic networks.
- Implement a web application for the analysis of metabolomic networks.

1.5 Thesis structure

The content of this thesis is focused on the study and exploratory analysis of mass spectrometry data, using machine learning algorithms (graph machine learning, anomaly detection), and is organized as follows:

Chapter 2 contains additional definitions regarding graph theory, graph machine learning, and mass spectrometry.
Chapter 3 contains the first manuscript, which presents a review about graph embeddings and their connection with mass spectrometry data.
Chapter 4 contains the second manuscript, which focuses on experimentally evaluating the use of graph machine learning models for exploratory analysis of mass spectrometry data.
Chapter 5 presents our general conclusions and future works.

Chapter 2

Theoretical framework

2.1 Graph theory

Graph theory is a branch of mathematics that focuses on the study of a structure of data called graphs. Graphs model pairwise relationships between objects (entities). The versatility of graphs has made them a widely used tool across numerous domains, including [6]:

- Computer science, graphs can be employed to model the structure of computer programs, facilitating a clearer understanding of how various system components interact.
- Physics, graphs can be utilized to model physical systems and their interactions, capturing relationships between particles and their properties.
- Chemistry, graphs can be applied to model molecule systems, such as shown in Figure 2.1, atoms or residues are nodes and chemical bonds or chains are edges.
- Biology, graphs can be applied to model biological systems, such as metabolic pathways, representing them as networks of interconnected entities.
- Social sciences, graphs can be used to analyze and understand complex social networks, capturing the relationships between individuals within a community.
- Finance, graphs can be utilized to analyze stock market trends and the relationships between several financial instruments.
- Engineering, graphs can be employed to model and analyze complex systems, including transportation networks and electrical power grids.

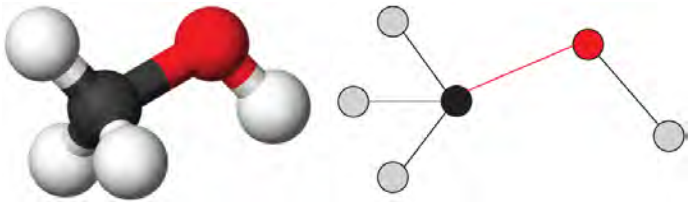


Figure 2.1 A single CH_3OH molecular [8].

Definition 1 (Graph) *In mathematics and computer science, a graph (network) is defined as* $G = (V, E)$, *where* $V = \{v_1, v_2, ..., v_N\}$ *is the nodes (vertices) set and* $E = \{e_1, e_2, ..., e_M\}, e_i \in V \times V$ *a set of edges formed by unordered pairs of nodes [3–5, 7, 20]. Additionally, the definition can be extended to* $G = (V, E, X)$, *where* $X = \{x_1, x_2, ..., x_N\}, x_i \in \mathbb{R}^F$ *is the node features (attributes) matrix, such that* x_i *is associated with* v_i.

Definition 2 (Weighted graphs) *A weighted graph* $G = (V, E, w)$, *where each of the edges has a real number associated with it. This weight is denoted* $w(uv)$ *for edge* (u, v) *[14].*

Definition 3 (Subgraph) *A subgraph of a graph* $G_1 = (V_1, E_1)$ *is a graph* $G_2 = (V_2, E_2)$ *which satisfies that:* $V_2 \subseteq V_1$ *and* $E_2 \subseteq E_1$ *[14].*

2.1.1 Graph measures

Definition 4 (First-order and second-order proximity) *The first-order proximity measures the proximity between a pair of nodes* v_i, v_j, *and represents the weight* w *of the edge* e_{ij} $(w \geq 0)$. *If the edge does not have a weight, then the default value is* 0. *Then, first-order proximity is defined as the neighborhood of the node* v_i *containing a set of adjacent nodes* $N_{v_i} = \{v_k \mid e_{ik} > 0, k \neq i\}$. *The second-order proximity measures the number of 2-hop paths between a pair of nodes* v_i, v_j *[1, 9].*

Definition 5 (Degree) *The degree of a node* $v_i \in V$ *is the number of edges incident in* v_i *and is denoted as* $deg(v_i)$. *If the degree is even, the vertex is called even; if the degree is odd, then the vertex is odd [14].*

Definition 6 (Density) *The density provides a measure of the proportion of edges present in the network compared to the maximum possible number of edges.*

2.1.2 Graph operators

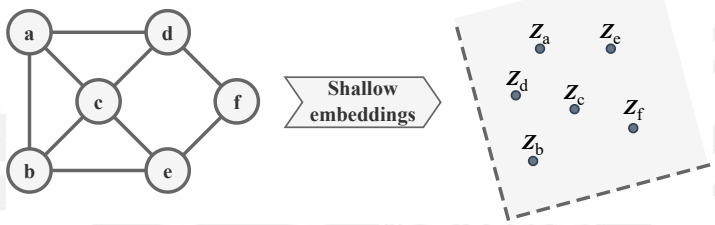
Definition 7 (Union and intersection) *Given two networks* $G_1 = (V_1, E_1)$, *and* $G_2 = (V_2, E_2)$.

- *The union is defined as:* $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$.
- *The intersection is defined as:* $G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$.

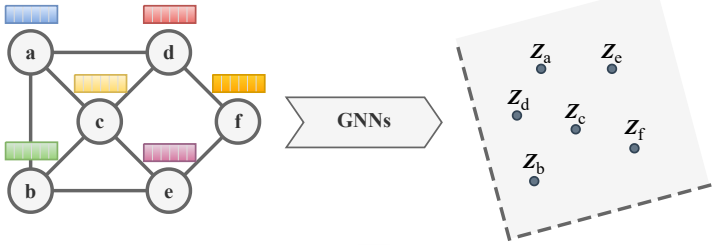
2.2 Graph machine learning

2.2.1 Node-level embeddings

It is a function $f(.)$ that maps each node of the graph $G = (V,E)$ or $G = (V,E,X)$ into a d-dimensional vector. In the case of shallow embeddings models (see Fig. 2.2(a)), the function is defined as: $f : A \rightarrow Z$, and in case of graph neural network based models (see Fig. 2.2(b)) it is defined as $f : (A,X) \rightarrow Z$. Where, $Z = \{z_1, z_2, ..., z_N\}$, $z_i \in \mathbb{R}^d$ with $d \ll N$. Usually, z_i and Z are called node embedding and embedding matrix, respectively.



(a) Shallow embeddings.



(b) Graph neural networks.

Figure 2.2 Graph embedding schemes.

2.2.2 Edge-edge embeddings

The set of edge embeddings $\hat{Z} = \{\hat{z}_1, \hat{z}_2, ..., \hat{z}_M\}$ is calculated for each pair of nodes $(v_i, v_j) \in E$ according to their node embeddings using the operators, such that $e_i \in E$ has its own $\hat{z}_i \in \hat{Z}$ [17].

- Concatenate: $z_i \oplus z_j$

- Sum: $z_i + z_j$
- Average: $\frac{z_i+z_j}{2}$
- Hadamard: $z_i * z_j$
- Weighted-L1: $|z_i - z_j|$
- Weighted-L2: $|z_i - z_j|^2$

2.2.3 Graph-level embeddings

The representation is calculated using the previous node embeddings $Z = \{z_1, z_2, ..., z_N\}$, $z_i \in \mathbb{R}^d$. Thus, it simply aggregates the node embeddings in the (sub) graph by the following operation called global pooling or graph-level readout [6].

- Mean global pooling: $\frac{1}{N}\sum_{i=1}^{N} z_i$
- Max global pooling: $\max_{i=1}^{N}(z_i)$
- Sum global pooling: $\sum_{i=1}^{N} z_i$

2.3 Mass spectrometry

Mass spectrometry (MS) is a microanalytical technique used to detect and determine the amount of an analyte, it is also used to determine the elemental composition and some aspects of the molecular structure of an analyte. Hence, mass spectrometry determine the mass-to-charge ratio (m/z) of the ions [19]. Moreover, mass spectrometry in combination with chromatography and nuclear magnetic resonance are the two major analytical avenues for the analysis of metabolic species in complex biological mixtures [13]. The analytical tools of choice for small-molecule analysis in metabolomics are mass spectrometry and nuclear magnetic resonance (NMR). MS and NMR methods are both supplementary and complementary to one another [13]. To date MS-based metabolomics approaches have been applied to study, among others, the effect of drugs, toxins, and various diseases on metabolite levels, to trace metabolic pathways and measure fluxes [13]. With consideration to the high complexity of biological mixtures, the vast majority of MS analysis methods involve prior separation using liquid chromatography (LC), gas chromatography (GC), or capillary electrophoresis (CE) [12, 13]. The main advantages of a mass spectrometer, are speed of analysis, high sensitivity, high resolution, simultaneous analysis of many

components of mixtures, ability to obtain information on the structure of the compounds, possibility of combining with separation techniques, quantitative analysis, analysis of the elemental composition, analysis of isotopic composition, unambiguous identification of the substance [12, 16]. In order to perform metabolomics research, researchers require specialized knowledge from three main research domains: bioscience, analytical chemistry, and informatics. These interdisciplinary disciplines are essential to achieve successful metabolomics studies [12]. An example of the mass spectrum is shown in the Figure 2.3, where the signal of 195.0 corresponds to the protonated molecule of caffeine.

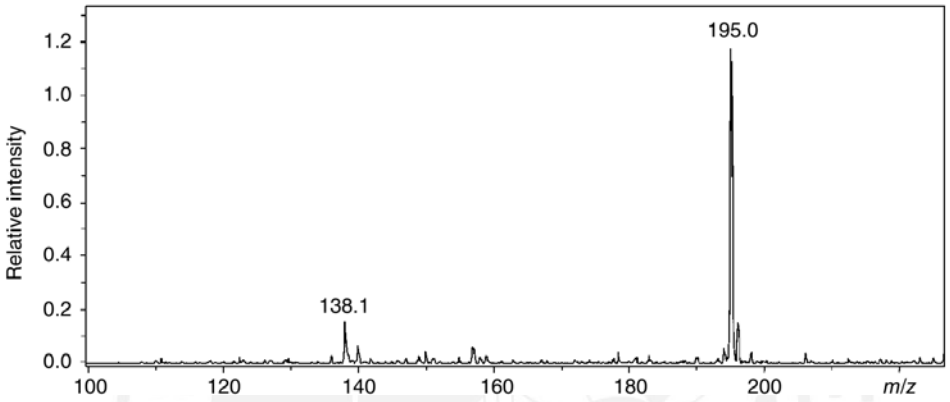


Figure 2.3 Mass spectrum of caffeine obtained by electrospray ionization mass spectrometry (ESI-MS) technique [16].

2.3.1 Mass spectrometer

Components of the mass spectrometer are: ion source, analyzer, detector, and data processing. Figure 2.4 shows the parts of the mass spectrometer with their different settings.

Ion source

Several ionization techniques are selected based on the type of organic compounds. Volatile samples are subjected to either electron or chemical ionization (CI), while non-volatile samples are fast-atom bombardment (FAB), matrix-assisted laser desorption ionization (MALDI), and electrospray ionization (ESI) [11].

In this process, a part of the kinetic energy of the accelerated electron, which is higher than the ionization energy of M, is transferred to the newly formed molecular ion. Thus, the resulting high internal energy of the open-shell $[M]^{+\bullet}$ ion induces its extensive fragmentation. The ions of the product thus formed as well as the eventually survived $[M]^{+\bullet}$ ion are finally extracted through an exit hole to the MS analyzer [16].

Mass analyzer

Mass spectrometer analyzers separate ions based on their m/z ratio, using an electric or magnetic field to achieve this separation. All ions resolved by mass are concentrated on a single focal point [11].

Ion detector

One of the most important, part of MS, is ion detector. This part is very important for sensitivity of the entire instrument. At the beginning of MS, photosensitive plates or Faraday cups were routinely used. Element multipliers are currently used as basic detectors in the majority of mass spectrometers due to their cost efficiency, sensitivity, and reliability. It consists of a few electrodes (usually from 10 to 20) called dynodes. An ion leaving analyzer hits the first dynode (so-called conversion dynode or conversion electrode). Kinetic energy of the ion is enough to remove few electrons from the electrode surface. As a result of this, a simply measurable signal coming from moving electrons (electric signal) rather than moving ions is generated [16].

Data processing

Metabolite data generated by mass spectrometry are normally complex, and hence multivariate statistical methods are often needed to extract information from such complex datasets. Of these, commonly used workhorse approaches are principal component analysis (PCA), logistic regression, and partial least squares discriminant analysis (PLS-DA). PCA, an unsupervised method, PLS-DA, a supervised method, logistic regression analysis enables selection of the highly ranked metabolites that contribute most to the classification. Data preprocessing, including peak picking, peak deconvolution, peak alignment, etc. [13].

2.3.2 Targeted and untargeted analyses

The mass spectrometry analysis includes two approach: targeted analysis and untargeted analysis.

Targeted approach

Target approaches can be classified into two acquisition methods: (i) using MS (protonated [M+H]+ or molecular ion peak M+ or in-source fragment ion) and (ii) using MS and MS/MS (parent and product ions based). During targeted analysis, the molecule of interest or m/z values (for MS and MS/MS) are predefined, and it is one of the most popular analyses for

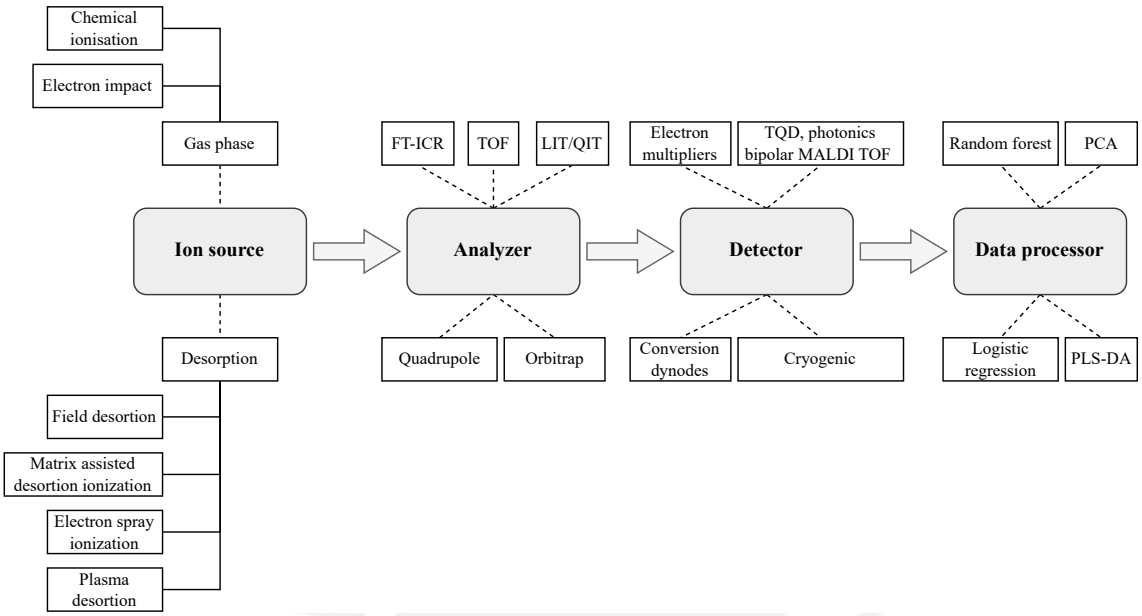


Figure 2.4 Parts of mass spectrometer. Adapted from [11, 16]

quantitation of analytes present in a complex samples matrix, with high sensitivity, accuracy, and specificity. [11].

Untargeted approach

Untargeted analysis can be classified into two types of acquisition modes: data-dependent analysis (DDA) and data-independent acquisition (DIA) full-scans. During both analysis modes, the purpose is to acquire MS and MS/MS spectra. The untargeted analysis (using MS and MS/MS) is the analysis to identify the unknowns in a sample, using parent ion and product or fragment ion analysis (MS/MS) [11].

In [15], to distinguish between the biological and non-biological, they consider human plasma as an example. As shown in the Figure 2.5, non-biological features do not originate from the sample being measured. Unlike artifacts, contaminants do represent real chemicals. Contamination-derived features from contamination are not biologically relevant because the compounds were not originally present in the sample prior to its preparation for metabolomic analysis. One approach to assess the potential conflation of biological and non-biological signals is to analyze a blank, which helps identify background interferences that can then be experimentally or computationally removed. In contrast, these biological features are divided into unique compounds and redundant. There are multiple causes of redundancy, also called peak degeneracy.

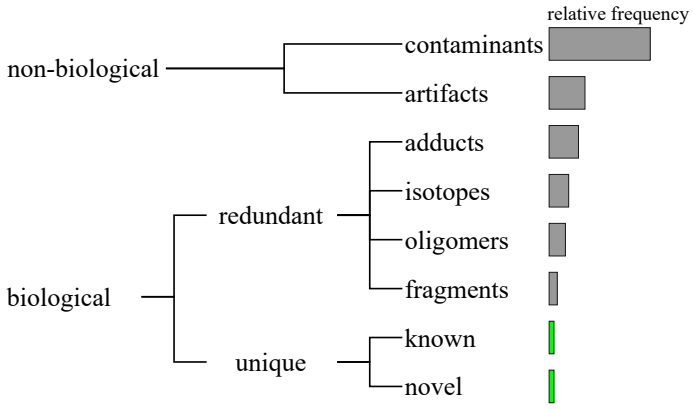


Figure 2.5 Composition of an untargeted metabolomic data. Adapted from [15].

2.3.3 Metabolomics workflow

A typical metabolomics workflow involves: experimental design, sample preparation, metabolite analysis (data acquisition), data processing and analysis; each of them are essential for obtaining reliable and biologically relevant data. An overview of the main steps is shown in Figure 2.6.

Experimental design

This step includes five main points: (1) generation of a working hypothesis, (2) determination of metabolomics targeted or untargeted approach, (3) data acquisition (e.g., analytical conditions), (4) sample preparation (preparation of biological samples, quenching, and extraction methods), and (5) data analysis (use of statistical analyses to visualize the data). This will not only help in the data interpretation but also in making a good experimental design in order to understand how many and what kind of samples are really important for analysis [12].

Sample preparation of biological samples

The sample preparation protocols for metabolomics analysis in wide-ranging studies involving microbiology, plant, animal, medical sciences, and food subjects. There are four processes in sample preparation: harvesting, quenching, grinding, and extraction. The most important processes are quenching and extraction. Quenching is the process of stopping biological reactions in a cell, and extraction is the process of obtaining metabolites from the cell [12].

Metabolite analysis using various instruments

The analysis can be targeted or untargeted. Targeted analysis hunts only "known metabolites" detected by single ion monitoring or selected reaction monitoring mode with the purpose of validating a biological hypothesis or machine-learning studies such as regression and discriminant analyses. On the other hand, untargeted analysis covers all detected peaks by means of scan mode and utilizes both annotated and unannotated peak information for statistical analyses. Scan mode analysis is usually utilized for data acquisition of mass spectra in GC/MS for untargeted analysis. This approach has been widely applied to discovery-phase studies such as biomarker identification for medical diagnostics [12].

Data processing and data analysis

A data matrix (sample vs. metabolites) acquired from a mass spectrometry-based metabolomics study includes not only biological information but also technical noise. There are several techniques for univariate or multivariate statistical analysis. Some statistical methods used are: Principal component analysis (PCA), Hierarchical cluster analysis (HCA), Correlation analysis (CA), Projection to latent structure regression (PLS-R), Projection to latent structure discriminant analysis (PLS-DA), and t-Test. It is very important to use and interpret the statistical result accurate [12].

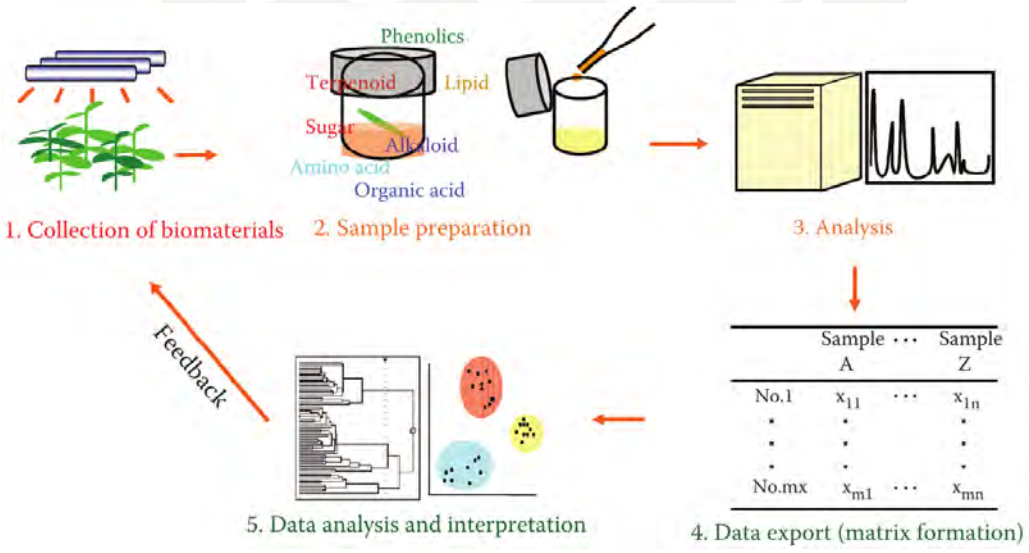


Figure 2.6 Metabolomics workflow [12].

2.3.4 Mass spectrometry data and interpretation

MS data contains real signal and noise. Approximately 90% of signal are noisy or redundant [2, 10]. While interpreting data obtained during analysis, it should be carefully noted that not always the m/z value can be directly related to the molecular mass of the analyzed compound [16]. The interpretation of the combined spectral data is often sufficient to establish the structure of the unknown compound. This approach will depend on the kinds of information desired and the unique style of the analyst [18]. In fact, metabolic pathway interpretation is an important step in metabolomics for connecting different omics domains in systems biology; however, it is highly dependent on the study purposes [13].

2.3.5 Mass spectrometry applications

MS has extensive application in different fields, including pharmaceuticals/nutraceuticals, food, health science, etc. The main applications of MS are categorized in qualitative, quantitative and both [11].

- Qualitative: determination of molecular weight, determination of the molecular formula, determination of the partial molecular formula, determination of the compound Structure
- Quantitative: isotope abundance assessment, determination of isotope ratio, differentiation between cis-isomers and trans-Isomers, mass spectrometry in thermodynamics, measurement of ionization potential, determination of ion–molecule reactions, detection and identification of impurity, identification of unknown compounds
- Both qualitative and quantitative: phytochemical analysis, structural elucidation of unknown phytochemicals, drug metabolism studies, clinical studies, forensic applications

Chapter 3

Graph embedding on mass spectrometry- and sequencing-based biomedical data

This first manuscript was published in the BMC Bioinformatics journal.
Alvarez-Mamani, E., Dechant, R., Beltran-Castañón, C.A. et al. *Graph embedding on mass spectrometry- and sequencing-based biomedical data*. BMC Bioinformatics 25, 1 (2024), https://doi.org/10.1186/s12859-023-05612-6.

In this chapter, we present a review, focused on studying the data generated by the analytical technique known as mass spectrometry, as well as its characteristics, with emphasis on the traditional workflow: experimental design, sample preparation, data acquisition, processing, analysis, and interpretation. Mass spectrometry data can be represented by graphs (networks) and take advantage of graph theory and graph machine learning algorithms. We then survey state-of-the-art models/algorithms/techniques for processing and analyzing mass spectrometry data. At this point we find models based on graph machine learning, specifically graph embeddings models, where the traditional tasks that can be solved are: node classification, edge prediction and community detection. In the study of mass spectrometry data, the type of study, either targeted or untargeted, must be taken into account. This information is important for the choice of a particular model. Graph machine learning models are classified from various points of view. The classical classification consists of: random walk-based, matrix factorization-based, and deep learning-based models. Another classification is from the point of view of the tasks to be solved, that is, methods that solve supervised learning and unsupervised learning tasks. Finally, another classification that we consider important is according to the characteristics of the input data. These can be a data set with static or dynamic graphs; homogeneous or heterogeneous graphs. In this first approach, we also

study the application of this type of models for the analysis of biomedical data, such as protein-protein networks interaction, drug-protein interactions, etc.

BMC Bioinformatics

REVIEW Open Access

Graph embedding on mass spectrometry- and sequencing-based biomedical data

Edwin Alvarez-Mamani[1,2], Reinhard Dechant[2,3], César A. Beltran-Castañón[1] and Alfredo J. Ibáñez[2,4*]

*Correspondence:
aibanez@pucp.edu.pe

[1] Engineering Department, Pontificia Universidad Católica del Perú, San Miguel, Lima, Peru
[2] Institute for Omics Sciences and Applied Biotechnology (ICOBA PUCP), Pontificia Universidad Católica del Perú, San Miguel, Lima, Peru
[3] Present Address: Calico Life Sciences, 1170 Veterans Blvd, San Francisco, CA 94080, USA
[4] Science Department, Pontificia Universidad Católica del Perú, San Miguel, Lima, Peru

Abstract

Graph embedding techniques are using deep learning algorithms in data analysis to solve problems of such as node classification, link prediction, community detection, and visualization. Although typically used in the context of guessing friendships in social media, several applications for graph embedding techniques in biomedical data analysis have emerged. While these approaches remain computationally demanding, several developments over the last years facilitate their application to study biomedical data and thus may help advance biological discoveries. Therefore, in this review, we discuss the principles of graph embedding techniques and explore the usefulness for understanding biological network data derived from mass spectrometry and sequencing experiments, the current workhorses of systems biology studies. In particular, we focus on recent examples for characterizing protein–protein interaction networks and predicting novel drug functions.

Keywords: Graph embedding, Biomedical data, Biological network

Introduction

In the literature, several reviews present graph embedding models used to solve multiple tasks such as pathogen-host protein interactions, predicting drug efficiency, linking a metabolite with a metabolic network, etc [1–3]. However, wide spread application of graph embedding techniques in the life-science community has been scarce, which may be in part because the complex mathematical framework underlying graph embedding requires considerable bioinformatical expertise. To make graph embedding known to a wider research community we have focused our review to be accessible for wet-lab biologists as well as bioinformaticians, mainly using more accessible wording for life scientists and focussing on potential future applications.

Biological data is usually presented as graphs; some of the most famous ones are represented in the book Cellular Biochemical Networks (Editor: Gerhard Michal), which describes the known metabolomic network of eukaryotic cells and comprises most of the cellular metabolites and their interactions (i.e., possible conversions and connections between metabolic pathways such as sugar and amino acid metabolism). Although traditional biology tools have been extremely successful in identifying most components and

some of the major linear interactions contained in the Cellular Biochemical Networks graphs, one of the significant challenges in biology is comprehending the nonlinear or dynamic interactions among the cellular constituents to unravel the organization and interactions within cellular networks. For example, understanding which metabolic subnetworks are active in a particular cell type under specific conditions is critical to decipher the influence of the metabolic network on cellular function.

Mass spectrometry (MS) is an excellent example of a tool for understanding the underlying interactions among large numbers of cellular constituents. MS-based metabolomic and proteomics studies can follow various linear and nonlinear interactions (based on signal abundances) and dynamic interactions from time series measurements. The interactions are visualized via correlation plots of the MS signals [4, 5]. In a correlation plot, metabolites, proteins, etc., are represented as dots (or nodes), and a line illustrates their correlations with other network elements. Using carefully designed experiments and bioinformatic tools makes it possible to model and quantify the different types of interactions between the nodes. Hence, a traditional approach in molecular biology is to compare two or more graphs to identify which metabolites or proteins in the biological network are associated with a particular physiology (i.e., disease) or phenotype of interest [4, 5].

Unfortunately, clear insight into biological information via visual inspection of the correlation plots is challenging due to the large number of biological species present in cells that MS can detect. Furthermore, artifacts such as the presence of ghost peaks or batch effects can futher obscure the information within these graphs [4, 5]. Graph embedding techniques have been developed to analyze complex graphs of diverse origins. A graph embedding technique takes graphs as input and converts the graphs into a matrix of vectors (i.e., a lower-dimensional latent space), thus allowing researchers to better identify the interactions between their different elements. Although graph embedding techniques have been applied to various fields of study, e.g., to analyze relationships between client and providers in financial transactions [6], to recommend locations using recommender systems [7], or to detect malware [8]; they have not been routinely applied to biological systems and are not well-known to life-scientists.

This review discusses the suitability of graph embedding techniques for analyzing masss spectrometry- and sequencing-based biomedical data and explains the theoretical background to understand graph embedding. We classify graph embedding techniques from the perspective of biomedical data, considering the canonical classification, thereby subdividing graph embedding techniques into random walk-based, matrix factorization-based, and deep learning-based algorithms. Additionally, we review articles that applied graph embedding for link prediction, node classification, and node clustering tasks on biomedical data and highlight novel biological insights obtained by graph embedding. In particular, we will focus on protein–protein and drug–protein interactions. Our review will help future readers to identify, which graph embedding models can be applied to solve a given task on biomedical datasets, which datasets can be used, and which metrics are available to evaluate the results.

The paper is structured as follows: section "Theory of graphs embeddings" contains the necessary definitions and summarizes the theoretical background of graph embedding. Then, section "Applications of graph embeddings in mass spectrometry- and

sequencing-based biomedical data" describes the existing applications of graph embedding techniques on biomedical data. Finally, section "Conclusion" discusses conclusions and future applications.

Theory of graphs embeddings

Background techniques

To be able to understand graph embedding, we first must introduce the term word embedding, which transforms a group of words (i.e., text) into a matrix of vectors and is frequently used in natural language processing (NLP) [9]. In more detail, word embedding technique results in the (n-dimensional) vector representation of a word (token) within a text [10]. Since words often occur in the same semantic or syntactic context, a cosine similarity measure among the vectors in the matrix can be used to identify the relationship between words. Hence, the semantic and syntactic similarity between words can be mathematically identified [11, 12]. For example, word embedding is used when a word processing program suggests a phrase after the computer user types just a few words. Two different strategies were proposed for word embedding (i.e., architectures): `Continuous bag-of-words` (CBOW) [13] predicts a word w_i in one particular position in the sentence based on the context of words surrounding that position $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$, while `continuous skip-gram model` [13] predicts the context (surrounding words) with respect to a particular word in the sentence. The first formulation of `skip-gram` model defines the conditional likelihood $P(w_{context} \mid w_{center}) \approx P(w_o \mid w_c)$ utilizing the function `softmax` [13, 14],

$$P(w_o \mid w_c) = \frac{exp(u_o^\top v_c)}{\sum_{i=1}^{|W|} exp(u_i^\top v_c)} \tag{1}$$

where o is the index of the context word (output) in the dictionary, c is the index of the central word (input) in the dictionary, and W is the vocabulary.

Similarly, *continuous bag-of-words* defines conditional likelihood $P(w_c \mid w_{o_1}, \ldots, w_{o_{2m}})$ [13, 14], where $o_1, \ldots, o_{2m}$ are the indexes of the context words in the dictionary.

$$P(w_c \mid w_{o_1}, \ldots, w_{o_{2m}}) = \frac{exp\left(\frac{1}{2m} u_o^\top (v_{o_1} + \cdots + v_{o_{2m}})\right)}{\sum_{i=1}^{|W|} exp\left(\frac{1}{2m} u_i^\top (v_{o_1} + \cdots + v_{o_{2m}})\right)} \tag{2}$$

Although the `skip-gram` architecture performs slightly worse on syntactic tasks than the `CBOW` model, it does much better on semantic tasks [13]. Executing the definition (Equ. 1) has a very high computational cost [13, 14]. Therefore, [15] optimized the training process of the `skip-gram` model by adding the `hierarchical softmax` and `negative sampling` techniques.

Graph embedding is applied to dot (scatter) graphs. In analogy to *word embedding*, in *graph embedding* a point (i.e., node) in a graph is considered as a word, which is surrounded by other points (i.e., words). Furthermore, the graph contains information about the relationship between any two given points (words); this relationship is defined as an edge between two nodes. Hence, graph embedding can be used to create a matrix of vectors for all the nodes in a graph based on their edges by using the following

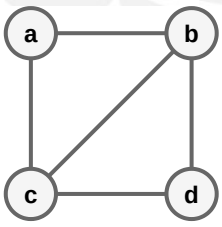
analogy [16–19]: given a sequence of words, $S_1^n = (w_1, w_2, \ldots, w_n)$ where $w_i \in W$, it can be inferred $P(w_n \mid w_1, w_2, \ldots, w_{n-1}) \approx P(v_i \mid v_1, v_2, \ldots, v_{i-1})$ and v_i represents a node in a graph G.

Graph embedding

The following definitions are useful to better understand and develop graph embedding and its applications.

Definition 1 (Graph) In mathematics and computer science, a graph is a scatter plot with a defined data structure. Let G be a graph, defined as $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of nodes (vertices), and E represents the connection (edge) between 2 nodes $(v_i, v_j) \in V$ [1, 20–23]. Given a graph (Fig. 1a), this graph can be represented by an adjacency matrix: is 1 when there is an edge from node v_i to node v_j, and is 0 when there is no edge (Fig. 1b). The adjacency list groups the neighboring nodes of each node v_i (Fig. 1c), while the edge list consists of ordered pairs (v_i, v_j) when there is an edge from node v_i to node v_j (Fig. 1d) [20, 21, 24, 25].

Definition 2 (Homogeneous and heterogeneous graphs) In a homogeneous graph, all nodes and/or edges are of the same type. For example, in the friends' network, each node represents a person, and an edge represents friendship between two people. In contrast, in heterogeneous graphs, nodes and edges can be of different types. Heterogeneous graphs are exemplified by an education network, in which there may be nodes representing teachers and students, and it is possible to have the relationships (edges) between teachers (colleagues), between teachers and students, and between students (classmates) [1, 20, 21, 24]. By their nature, biochemical networks can be defined as homogeneous or heterogeneous graphs. For example, protein–protein interaction studies are represented



(a) Input graph

	a	b	c	d
a	0	1	1	0
b	1	0	1	1
c	1	1	0	1
d	0	1	1	0

(b) Adjacency matrix

a	b	c	
b	a	c	d
c	a	b	d
d	b	c	

(c) Adjacency list

a	b
a	c
b	c
b	d
c	d

(d) Edge list

Fig. 1 Graph representation

in homogeneous graphs [26–29], while miRNA-disease/gene interaction studies are represented by heterogeneous graphs [30–32].

Definition 3 (Directed and undirected graphs) In directed graphs (digraph), the list of nodes (i.e., vertices) that generates the graph is ordered, and each interaction (i.e., edge) has a direction. Traversal in this type of graph is done according to the direction of the interactions among nodes, while in undirected graphs traversal can be done in both directions [1, 20, 21, 24]. In metabolic pathways, both types of graphs are present. Metabolic pathways, in which each product (i.e., node) is solely dependent on its precursor (i.e., a previous node in the pathway), can be defined as directed. However, most metabolic pathways are represented as undirected graphs, since their chemical reactions are reversible and regulated by feedback loops, where downstream products influence the formation of their upstream precursors (e.g., in glycolysis) [33, 34].

Definition 4 (First-order and second-order proximity) The first-order proximity measures the proximity between a pair of nodes v_i, v_j, and represents the weight w of the edge e_{ij} ($w \geq 0$). If the edge does not have a weight, then the default value is 0. Then, first-order proximity is defined as the neighborhood of the node v_i containing a set of adjacent nodes $N_{v_i} = \{v_k \mid e_{ik} > 0, k \neq i\}$. The second-order proximity measures the number of 2-hop paths between a pair of nodes v_i, v_j [2, 24].

Definition 5 (Graph embedding) Given a graph as input $G = (V, E)$, graph embedding (see Fig. 2) is defined as a mapping function $f : v_i \rightarrow Z_i \in \mathbb{R}^d$ (latent space) with $i \in \{1, 2, \ldots, n\}$ where $d \ll |V|$ and Z_i is a vector of dimension d known as an *embedding* [2, 22, 24].

Classification of graph embedding techniques

Most commonly, graph embedding techniques are classified as either matrix factorization-based, random walk based, or deep learning-based [1, 2, 22–24, 35].

However, in the literature, an alternative classification has been introduced based on the point of view of the mathematical problems, which can be *vector point-based*, *gaussian distribution-based*, or based on *dynamic graph embedding* [1]. Vector point-based approaches aim to project the nodes of a high-dimensional graph onto low-dimensional vectors within a vector space [1]. Gaussian distribution-based methods allow the vector representation (embedding) of a node as *potential functions* of *continuous densities* in a

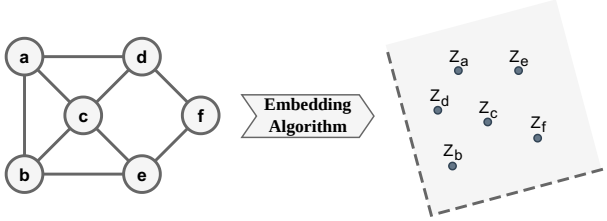


Fig. 2 Graph embedding scheme

Table 1 Network embedding models

Category	Publications
Non-attributed network	
Shallow embeddings	[16, 17, 19, 37–53]
Graph neural networks	[54–62]
Attribute network	
Semantic matching models	[63–78]
Translational distance models	[79–85]
Meta-path-based methods	[18, 86–90]

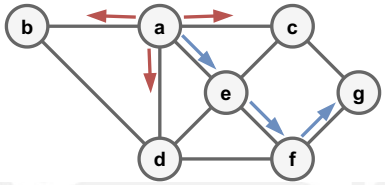


Fig. 3 BFS (red arrows) and DFS (blue arrows) traversals, from node *A* with a path length of 3

vector space. [1]. Dynamic graph embedding is often the method of choice for practical applications, as many networks are dynamic and evolve, leading to the addition of removal of nodes or edges [1].

Alternatively, it was proposed that embedding techniques can be grouped from the perspective of biomedical networks, including biomedical relation data, biomedical knowledge graphs, biomedical ontology, or clinical data, in *non-attributed network embedding* and *attributed network embedding* [36]. Below is the classification of non-attributed network embedding [1, 2, 22, 24, 35, 36] and attributed network embedding [2, 36]. Table 1 shows the graph embedding models published by category.

Mathematical concepts behind graph embeddings

Shallow embeddings are the earliest graph embedding technique applied to life-science data based on homogenous networks (i.e., networks based on only one biological entity, such as proteins). Shallow graph embeddings are subdivided into random-walk and matrix-factorization algorithms. Examples of random-walk algorithms are (`DeepWalk` [16] and `Node2vec` [17]; while matrix-factorization examples are graph factorization [43] and GraphRep [44].

DeepWalk [16] was the first graph embedding technique used to represent the vertices (nodes) of a homogeneous graph in vectors [91]. The process begins when the random walk algorithm generates a sequence of vertices. The model is then trained using the `skip-graph` algorithm [13]. Finally, the result is the vector representation for each vertex, also called embedding.

Node2vec [17] is a generalization of `DeepWalk` [16]. The authors added two parameters, p, and q, which drive the generation of paths (see Fig. 3) by using the idea of breadth-first traversal (BFS) and depth-first traversal (DFS). When $q > 1$, the traversal approaches BFS, and the random walks lead to a *micro-view* of node

neighborhoods. In contrast, $q < 1$ is an exploration *macro-view* that approximates a DFS traversal for node neighborhoods [1]. The authors of the base article used the values of $p = 1$, $q = 2$ for a micro-view and the values of $p = 1$, $q = 0.5$ for a macro-view. The parameters p and q also control how fast a path is explored, and the neighborhood of an initial node v_i is left. The authors performed multi-label classification and link prediction experiments to verify their proposal. Results were evaluated using the F1-score metric.

While Deepwalk and Node2vec provided a solution to tasks such as link prediction, node classification, node clustering (community detection), and visualization, two random-walk algorithms, `Netpro2vec` [92] and `Pathway2vec` [33], were proposed to better analyze biomedical datasets.

Netpro2vec [92]: In the techniques described above, nodes of a network were transformed into tokens. Instead, the main concept of `Netpro2vec` is to transform networks into documents. The process is carried out in 3 steps: 1) building the probability distributions representing each graph, 2) extracting tokens from probability distributions, and 3) building the graph embedding using token extraction. The graph is then represented as a word document (a set tokens), and the `Doc2vec` (document embedding) technique is applied to obtain the graph embedding [93]. The proposal was compared with other techniques of whole-graph embeddings to solve classification tasks in gene networks. The results were evaluated based on accuracy, precision, recall, F-measure, and Matthews correlation coefficient (MCC) metrics.

Pathway2vec [33] incorporates multiple random walk-based techniques, `Node2vec` [17], `Metapath2vec`, `Metapath2vec++` [18], `JUST` [94], and `RUST` [33], to represent learning by automatically generating features of metabolic pathways. It consists of three layers that interact: compounds, enzymes, and pathways. This interaction between layers results in a heterogeneous network of multi-layer information, and each layer has associated nodes. The layered architecture captures meaningful relationships to learn a low-dimensional space based on neural embeddings of metabolic features. Finally, applying the `skip-gram` [13] model, the embeddings for each node are extracted. `Pathway2vec` was applied for node clustering, embedding visualization, and pathway prediction tasks. Evaluation of the results was performed using `MetaCyc` software and F1-micro metric.

Graph Factorization (GF) [43]: `GF` is a factorization technique based on partitioning a graph to minimize the number of neighboring vertices instead of edges between partitions. `GF` begins from the assumption that the information regarding the presence of an edge (i, j) with a weight Y_{ij} can be captured by the inner product between vertices with attributes $\langle Z_i, Z_j \rangle$. Finally, the value of the vector Z is determined by the following objective function:

$$f(Y, Z, \lambda) = \frac{1}{2} \sum_{(i,j) \in E} (Y_{ij} - \langle Z_i, Z_j \rangle)^2 + \frac{\lambda}{2} \sum_i \|Z_i\|^2 \tag{3}$$

where λ is the regularization parameters, E is the list of edges. To validate their proposal, the authors applied `GF` on a graph of 200 million vertices and 10 billion edges. In order to evaluate convergence and execution time, they used 3 architectures: a single machine,

a synchronous parallel implementation and an asynchronous parallel implementation. The results showed that asynchronous parallel implementation is very beneficial for scalability.

GraRep [44]: `GrapRep` is a model for learning node representation. This model captures the relational information of different k-steps with different values of k between vertices of the graph, directly manipulating different global transition matrices defined on the graph without slow and complex sampling processes. `GraRep` defines different loss functions and optimizes each model with matrix factorization techniques, constructing global representations of each node by combining the different model representations. Experiments were run to solve the node clustering and node classification tasks on linguistic networks and social networks, respectively. In both tasks, `GraRep` showed an empirical efficiency of the learned representations compared to the `LINE` and `DeepWalk` models.

While shallow-embedding algorithm applications focus on solving link prediction, node classification, and community detection tasks, more complex problems such as graph matching, subgraph matching, and calculating the maximum common subgraphs require more complex models. Graph-neural network (`GNN`) algorithms can address these problems by combinatorial optimization using graph theory. Furthermore, these problems are solved through representation learning (deep learning); for example, in [95] a `GNN` model is proposed that addresses the subgraph matching problem for molecular fingerprint detection.

Graph Convolutional Network (GCN): Kipf et al. [56] present GCN for semi-supervised learning that works directly on graphs. `GCN` is a variation of convolutional neural networks. It scales linearly with the number of edges and encodes the local structure of the graph and features of nodes. The task of node classification is approached on a graph with partially labeled nodes, using a neural network $f(X, A)$ trained in a supervised environment with node feature matrix X and adjacency matrix $A \in \mathbb{R}^{N\times N}$. For this purpose, a multilayer `GCN` is considered with the following layer-wise propagation rule.

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{4}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of undirected graph with added loops, I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ is a layer-specific trainable weight matrix, $\sigma(.)$ is an activation function, $H^{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of activations in the l^{th} layer $H^{(0)} = X$. The experiments were run on 4 datasets (Citeseer, Cora, Pubmed, and NELL), and the results showed that `CGN` significantly outperforms `DeepWalk`.

In the case of attribute data—biomedical data based on heterogeneous networks (i.e., networks based on more than one biological entity, such as drug–protein target interactions)—the graph embedding algorithms must consider both the node distribution and the edge information of the graph. Embeddings are generated that encode the proximity between nodes based on their attributes and connectivity patterns. Graph embeddings algorithms for attribute data can be divided into semantic matching models (e.g., `DDKG`, `DistMult`, etc.), translational distance models (e.g., `TransE`, `TransR`), and meta-path-based methods (e.g., `Metapath2vec`).

DDKG: Xiaori et al. [96] used an approach denominated "attention-based knowledge graph representation learning framework" or `DDKG` to simultaneously consider drug

attributes and triple facts in knowledge graphs (KG). A triple fact is the link between one entity (e.g., metabolite, protein, etc.), usually referred as subject or head, and another entity referred as object or tail. The relationship between this two entities is referred as relationship or label. Xiaori et al.'s work aimed to use all the information available in biomedical KGs and improve the results in the link prediction task in drug–drug interaction (DDI) networks. The proposal was developed in 4 steps: 1) Building the KG, 2) Generating the initial embeddings for each drug according to its KG, 3) Generating the global embeddings of the drugs considering the node-embeddings of their neighbors, 4) finally, DDKG determines the probability of interaction of drugs in pairs with their respective embeddings through a binary classification. The experiments were conducted on two biomedical KGs and compared with ten state-of-the-art models, including `LINE` and `SDNE`. Results obtained from DDKGs were evaluated by metrics of accuracy, sensitivity, specificity, AUC, and AUPR, demonstrating that DDKGs outperformed the state-of-the-art models.

DistMult [67] considered learning entity and relationship representations in knowledge bases (KBs) using the neural-embedding approach. The learning process seeks to learn entity and relationship representations such that valid triple facts (i.e., known facts) receive high scores. The triple facts are denoted by (e_1, r, e_2), where e_1 is the subject, e_2 is the object, and r is the relationship between the two. The first layer of the model projects a pair of entities from the input into low-dimensional vectors, and the second layer combines these two vectors into a scalar to be compared by a scoring function. Entity representation learning can be defined as:

$$y_{e_1} = f(WX_{e_1}), \quad y_{e_2} = f(WX_{e_2}) \tag{5}$$

where f can be a linear/nonlinear function, W is a parameter matrix, W can be initialized randomly/pre-trained, and X is a one-hot/n-hot vector representing the input entities e_1 and e_2. `DistMult` was empirically evaluated for link prediction tasks on the Freebase dataset. The results showed that a bilinear model successfully captures the compositional semantics of the relationships. It is also reported that `DistMult` outperforms `TransE` with a top-10 accuracy of 73.2% versus 54.7%.

TransE: Antoine et al. [80] addressed the problem of embedding different class entities (e.g., metabolites, proteins, etc.) and relationships of multi-relational data in low-dimensional latent spaces. The primary condition is that all the different entities (e.g., protein, metabolite, gene, etc.) must be present in a directed graph. In this directed graph, a triple fact consists of one entity (designated head), which is related to another entity (designated tail) by an edge (designated label). `TransE` is an energy-based model that learns embeddings of low-dimensional entities. For `TransE` the relationships are represented as translations in latent space; if a strong relation (edge) exists among two nodes (i.e., head and tail), then the embedding of the tail entity must be similar to the embedding of the head entity plus some vector that satisfies the relationship. For its simplicity, TransE has a small number of parameters and is scalable. Experiments showed that `TransE` performs well and significantly outperforms the `RESCAL` method in the link prediction task on two large knowledge base, Firebase and Wordnet.

TransR [82]: In contrast to the `TransE` model, where entities and relations (edges) are embedded in the same latent space, in `TransR` it was proposed to build the embeddings

of the entities and the edges in separate latent spaces linked by specific relation matrices, yielding one entity space and multiple relation spaces. `TransR` was based on the idea that entities that have a relationship of the form (head, label, tail) are first projected from the entity space into the r-relation space as h_r and t_r with M_r operation, and then $h_r + r \approx t_r$. The relation-specific projection can make the head/tail entities that actually hold a strong relation (edge) close to each other and also move away those that do not. In the experiments, Lin et al. [82] evaluated the model with three tasks: link prediction, triple classification, and relational fact extraction using the WordNet and Freebase datasets. The results showed that `TransR` obtains significant improvements compared to `TransE`. Additionally, they proposed `CTransR`, a combination of `TransR` and Clustering.

Metapath2vec [18]: Unlike `DeepWalk` [16] and `Node2vec` [17], `Metapath2vec`, guides and generates paths using random walks through meta-path schemes. It captures the structural and semantic relationships between different types of nodes in heterogeneous networks. Formally, a meta-path is defined as a path $\mathcal{P}$ represented by, $\mathcal{P} : V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \dots V_t \xrightarrow{R_t} V_{t+1} \dots \xrightarrow{R_{l-1}} V_l$, where $R = R_1 \circ R_2 \circ \cdots \circ R_{l-1}$ defines complex relationships between node types V_1 and V_l. The `skip-gram` architecture is also used by `Metapath2vec` to determine embeddings. Dong et, al. [18] evaluated their proposal on heterogeneous graphs for solving multi-classification nodes, node clustering, and similarity search problems. The results were evaluated using the F1-score metric.

Applications of graph embeddings in mass spectrometry- and sequencing-based biomedical data

Applications of graph embedding techniques for mass spectrometry- and sequencing-based data covered in this review are summarized in Table 2 [26, 31, 33, 92, 97, 98]. By their nature, certain—OMICs data can be stored in a graph data structure. For example, gene–gene, protein–protein, and metabolite-metabolite interactions can be stored in homogeneous graphs. In contrast, heterogeneous graphs can contain multiple species, e.g., drug–protein, gene–protein interactions, etc. and analyzing these graphs can contribute to biological knowledge. However, computational tools to study graph data structures in biological graphs can suffer from high computational and space costs, especially in large-scale information containing graphs [28]. Graph embedding algorithms can then be used to identify interactions between heterogeneous nodes such as: drug–target [26, 99–101], miRNA-disease [30, 31], miRNA-target [32], miRNA-gene [32], microbe-drug [102], gene–disease [31, 103], gene–pathway [31], cell–gene [104], chemical–disease [31]. On the other hand, the interaction between homogeneous nodes may be protein–protein [26–29], drug–drug [34, 100, 102], microbe-microbe [102], gene–gene [104].

As an example, Su et al. [28] applied graph embedding to improve the identification of protein–protein interactions. To avoid the high computational cost of identifying the possible protein–protein interactions based on previous graph embedding techniques, the authors studied different approaches (algorithms) to accelerate graph embedding and improve its accuracy. The authors' contribution was 2-fold. Firstly, their approach denominated LPPI integrated protein attributes into the graph embedding task. This way, multi-view information was used, improving the accuracy of the

Graph embedding on mass spectrometry- and sequencing-based biomedical data

Table 2 Summary of graph embedding on biomedical data

Techniques	Dataset	Applications	Evaluation Metrics
Combined DeepWalk, LINE, Node2vec, and SDNE [26]	MATADOR, PubTator, and BioGRID	Link prediction	AUC, AUPR, MAP, Avg. R-precision, and Precision@k
HeteWalk [115]	HPRD, MISIM, MimMiner, DisGeNET, and miRTarBase	Link prediction	AUC
Cascade model [97]	BioChem, Drug Bank, and PubChem	Link prediction	Accuracy, hits@10, and AUC
[29]	Krogan, Dip, and BioGRID	Node clustering	Precision, recall, F-score, fraction, geometry accuracy, and MMR
HNERMDA [102]	MDAD and aBiofilm	Link prediction	Accuracy, AUC, and AUPR
PmDNE [30]	HMDD3.0	Link prediction	AUC, AUPR, precision, accuracy, recall, and F1-score
HO-VGAE [27]	HI-II-14, HI-III, Lit-BM-13, BioGRID, and Bioplex	Link prediction	AUPR, Precision@k
HMNE [105]	Lazega, CKM, DBLP, C.elegans, H.genetic, PPI, and Twitter	Link prediction and node classification	F1-micro, F1-macro, and AUC
TriModel [99]	DrugBank_FDA, KEGG_MED, and Yamanishi_08	Link prediction	AUC and AUPR
FactorHNE [103]	DisGeNet, HPO and Orphanet, STRING 10	Link prediction	AUPR, AUC, Precision@K, and Recall@K
[100]	DrugBank_FDA, UNIPROT	Link prediction, node clustering	Accuracy
Hybrid model GVS [31]	GO, HPRD, CTD, HMDD and MATADOR	Link prediction	Accuracy and F1-score
DeepWalk and Node2vec [98]	DrugBank, Bio2RDF, human disease network, SIDER, KEGG, and PharmGKB	Link prediction	AUC and AUPR
Netpro2vec [92]	LFR, MREG, Kidney RNASeq, Brain fMRI COBRE, Breast RNAseq, Breast Microarray, MUTAG	Node classification	Accuracy, precision, recall, F-score, and MCC
BiLSTM [101]	Human, DUD-E, and ChEMBL	Node classification	AUC, precision, and recall
scLINE [104]	Usoskin, Li, Pollen, Patel, Darmanis, Camp, Muraro, and Petropoulos	Node clustering	DBI, NMI, ARI, Jaccard and Purity
PRD [34]	Bio2RDF, and DDI Corpus	Link prediction	AUC, AUPR
ACNE and ACNE-ST [116]	Cora, Citeseer, Wiki, and DBLP_C4	Node classification and node clustering	F1-micro and F1-macro
Pathway2vec [33]	EcoCyc, HumanCyc, AraCyc, YeastCyc, LeishCyc, and TrypanoCyc	Link prediction and node clustering	F1-micro
LPPI [28]	PPI network and GraphSAGE-PPI	Link prediction and Node Classification	Accuracy, sensitivity, precision, MCC, and AUC
MRMTI [32]	miRTarBase, miRBase, HumanNet, and biomaRt	Link prediction	AUC, AUPR, precision, recall, F1-score, and balanced accuracy
CANE [117]	Disease Encylopedia Section of XYWY.com.	Link prediction	Precision@k and recall@k

AUPR area under precision-recall curve, *ROC* receiver operating characteristics, *AUC* area under the curve ROC, *MCC* Matthews correlation coeffici t, *DBI* Davies–Bouldin index, *NMI* normalized mutual information, *ARI* adjusted rand index, *MMR* maximum matching ratio, *MAP* mean average precision

graph embedding process. Secondly, the graph was reconstructed using the `GraphZoom` algorithm to reduce the graph's size. Therefore, the authors could accelerate the efficiency of the embedding algorithms. Combining the above two aspects, the authors' algorithm, LPPI, saves execution time without losing accuracy (AUC 0.99996) in identifying protein–protein interactions in a large dataset.

Despite representing a major advance in the use of graph embedding, Su et al. [28] only used a homogeneous dataset from protein data. However, biological information from systems biology studies is typically derived from multi-omics datasets and contain heterogeneous information (DNA, RNA, protein, and metabolite information). Furthermore, as the network of interactions is time or condition-sensitive, multilayer networks must be considered [4].

Gong et al. [105] proposed the use of a multilayer network embedding to handle data sets with multiple types of nodes and edges found in heterogeneous graphs. This approach becomes extremely useful for evaluating the performance of node embedding in link prediction, which tries to predict edges that most likely will appear in theoretical networks (not experimentally measured data); this is similar to the approach performed by bioinformatics in in-silico studies. As some tested datasets are very large and complex, it is hard to predict links on the whole node sets. Hence, Gong et al. [105] suggested first extracting a core set of nodes of each dataset and conducting link prediction in these core sets. Hence, many authors similar to Gong et al. are encouraging the use of more complex graph-embedding algorithms that are based on combinations of the above-mentioned ones. These combinations of graph-embedding algorithms are known as encores or graph neural networks.

For example, Ray et al. [106] used a combination of graph-embedding algorithms as proposed by Gong et al. to generate a graph embedding encore algorithm approach to identify potential drugs that could affect the protein–protein interaction (PPI) between the SARS-CoV-2 virus and its human target proteins. The SARS-CoV-2 viral protein and human interaction datasets (i.e., protein interaction graph) were based on the experimental data obtained by Gordon et al. [107] by means of affinity-purification mass spectrometry (AP-MS) screening and on the theoretical data by Dick et al. [108].

The graph embedding-based algorithm proposed by Ray et al. [106] to repurpose drugs against COVID-19 considered that the available data was heterogeneous. They suggested to combine three different data sets: (i) SARS-CoV-2—host protein interactions, (ii) human protein–protein interactions, and (iii) drug–human protein interactions to predict possible novel treatments to interfere with infection. As described by Gong et al. [105], these three datasets were very large and complex; hence, Ray et al. [106] had to reduce the dataset complexity by performing the data reduction step, i.e., a first graph embedding based on the `Nod2vec` algorithm to obtain the feature matrix (`X`). In the second step, the novel graph embedding algorithm denominated variational graph autoencoder (VGAE) was used for link prediction tasks. As input, VGAE receives the adjacency matrix (`A`) and the feature matrix (`X`) from the original graph (`X` replaces the one-hot matrix that the VGAE model uses by default and also helps improve prediction precision). The encoder of VGAE converts the input data to lower-dimensional representation (`Z`) and the decoder takes `Z` to reconstruct the

original input in ($\hat{\mathbb{A}}$), where $\hat{\mathbb{A}}$ is similar to $\mathbb{A}$, and in $\hat{\mathbb{A}}$ new connections between the different types of nodes can be discovered.

The results of Ray et al. [106] were compatible with those observed by other authors. For example, Ray et al. [106] identified the angiotensin-converting enzyme-2 (ACE-2) as a potential drug target against SARS-CoV-2 [109, 110]. Interestingly, the authors also found that drugs used to prevent Malaria and pneumocystis pneumonia (PCP) relapses, such as Primaquine, have therapeutic potential against SARS-CoV-2 based on the interaction of Primaquine with the TIM complex, consisting of TIMM29 and ALG11.

Similarly, Zitnik et al. [111] used a graph convolutional network, a combination of graph-embedding algorithms with a convolutional neural network that can work directly on graphs, to predict clinical side effects in patients taking multiple drugs simultaneously.

As in the case of Ray et al. [106], Zitnik et al. [111] combined multimodal graphs of protein–protein interactions, drug–protein target interactions, and known clinical drug side effects. Their new graph embedding algorithm, named Decagon, could accurately predict drug side effects in patients with complex diseases or co-existing conditions necessitating simultaneous medication for their treatment.

The use of shallow embeddings, such as (Nod2vec) is limited as shallow embeddings do not share information between the nodes and do not take advantage of the characteristics of the nodes in the coding process. To mitigate these limitations, graph neural networks (GNN) have more sophisticated encoders that take advantage of the structure, features, and attributes of graphs [112].

Su et al. [113] proposed constrained multi-view nonnegative matrix factorization (CMNMF), a model based on GNN, to determine the similarity between drugs and viruses within their space of characteristics (latent space). Therefore, CMNMF is oriented towards preserving drug and virus similarity information as much as possible. Then, they apply a graph convolutional network (GCN) with attention-based neighbor sampling to optimize the vectorial representation of drugs and viruses in virus-drug associations (VDA) networks, whereas VDA networks are considered heterogeneous graphs. The experiments were executed on three VDA datasets to identify possible drugs against SARS-CoV-2. The embedding algorithm from Su et al. outperformed other models and was evaluated with the accuracy, F1, AUC, and AUPR metrics.

Decagon, a DeepWalk neural graph embedding, outperformed baseline algorithms by up to 69% (accuracy). Specifically, Decagon could automatically predict side effects with a known strong molecular basis with high precision, but still performed well on predicting side effects with a non-molecular basis due to its effective sharing of model parameters across edge types.

Finally, Nelson et al. [114] mentioned the advantages of graph embedding techniques compared to other techniques that operate directly on biological/biomedical networks. One advantage is a more rapid analysis of the learnt embedding. Unlike the tasks mentioned in the other works (link prediction, node classification, and node clustering), Nelson et al. [114] demonstrated the usefulness of graph embeddings for more specific tasks in biology, such as protein network alignment, protein module detection, and protein function prediction. Taken together, these examples establish the high value of graph embedding techniques for the analysis of mass spectrometry—and

sequencing-based—OMICs datasets. Several other applications have been published, which could not be discussed in greater detail, but have been showcased in Table 2 and classified for the use for (i) link prediction, (ii) node classification, and (iii) node clustering tasks.

Although Table 2 shows how graph embedding algorithms have become popular for representing biomedical data, several major limitations are apparent that limit the general applicability of graph embedding to life sciences:

- Most graph embedding algorithms have been developed to accomplish a specific task on a specific dataset, with no standards or even flexibility for incorporating other datasets. For using the same graph embedding algorithm to solve a different task, the new data set must be rewritten, thus limiting the application for other researchers.
- Shallow-embedding algorithm applications are limited in their applications, such as link prediction, node classification, and community detection tasks. More complex problems such as graph matching, subgraph matching, and calculating the maximum common subgraphs require more complex models requiring combinatorial optimization (graph theory). Furthermore, these problems are solved through representation learning (deep learning). However, most deep-learning graph embedding techniques are not deterministic because they use probabilities to perform their tasks, yielding similar, but not identical results for different runs.
- Loss of structural information: graph embedding methods typically aim to preserve the proximity of nodes based on their graph structure. However, they may lose certain structural information during the embedding process. For instance, (i) higher-order relationships within the graph may not be accurately captured. Furthermore, (ii) graph embeddings may not effectively leverage node attributes or features. Node attributes (metadata) can provide valuable information in life sciences, such as measurement conditions. It may be computationally expensive to maintain graph embeddings for (iii) dynamic data sets where nodes and edges are frequently added, removed, or modified (due to experimental conditions).
- Interpretability: The interpretability of graph embeddings can be more challenging compared to other clustering techniques, as it is often difficult to interpret the specific features or relationships each dimension captures.

Addressing these limitations is an active area of research, and researchers continue to develop new techniques and algorithms to enhance the performance and versatility of graph embedding methods to make them more applicable to life-science research questions.

Conclusion

As can be easily appreciated from the by far not exhaustive list of discussed algorithms for graph embedding in this review, there is currently not yet a gold standard for graph embedding for biological data emerging that can provide reliable data for biologists and serve as a reference point for future developments of in the field. So far, the presented applications for graph embedding on biological data have all been developed for the specific data sets at hand. All these studies have thus mainly remained theoretical, focusing

on the development of computational techniques rather than taking the interpretation of the data to the identification of novel biology or drug developments. Yet, with the ever-growing datasets available to life science researchers, the community needs novel tools to understand better the underlying biological processes. Given their nature of reducing the dimensionality of complex data, graph embedding algorithms are an exciting and novel tool for extracting novel insight from large biological datasets (Table 2). We envision that graph embedding will become an essential tool aiding hypothesis generation leading to novel biological discoveries.

Specifically, graph embedding techniques hold significant potential in various biological and biomedical research fields. In the context of the drug–disease association (DDA), disease-gene association (DGA), drug–target interaction (DTI), protein–protein interaction (PPI), and drug–drug interaction (DDI) (Table 2), graph embedding methods can provide valuable insights and aid in understanding complex relationships. By representing drugs, diseases, genes, targets, and proteins as nodes in a graph and capturing their interactions as edges, graph embedding algorithms can (i) infer novel insight into a biological system based on information about its elements (i.e., link prediction), (ii) classify the relevance of biological elements (e.g., proteins, metabolites, etc.) and their interactions within a system (i.e., node classification), and (iii) identify a phenotype or physiology of interest based on the networks formed by their elements (i.e., node clustering).

Furthermore, with the help of low-dimensional representations obtained using graph-neural networks (GNN) algorithms, it is possible to encode the underlying relationships and functional associations to find similarities between individuals sharing the same condition (e.g., graph matching or subgraph matching). These low-dimensional embeddings can then be leveraged to gain an understanding of the underlying molecular events occurring within the biological system (i.e., molecular phenotype characterization).

Hence, the ability to integrate multiple data sources, such as genomic, transcriptomic, proteomic, metabolomic, and clinical data, further enhances the predictive power and potential impact of graph embedding techniques, mainly in the field of personalized medicine, paving the way for improved disease management, identifying potential therapeutic targets, elucidating underlying molecular mechanisms, and exploring drug synergy or adverse interactions.

In conclusion, this increased predictive power gained by using graph embedding techniques on biological data will allow life-science researchers to conduct more targeted experiments by extracting novel unseen links. Developing applications will require substantial further research on the bioinformatic side to identify the most promising approaches to be applied to specific types of datasets, as well as thorough experimental validation of the generated outputs. Despite posing a challenging problem to either field, the rapid rise of AI tools in our everyday life as a researcher will certainly fuel interest in incorporating novel AI-based analysis methods on high dimensional biological data. Therefore, we anticipate that graph embedding applications will soon be invaluable in the broader life science community.

Acknowledgements
EA doctoral studies are funded by *Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica* (CONCYTEC), and *Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica* (FONDECYT), under contract No. 174-2020-FONDECYT "Doctoral Programs in Peruvian Universities". AI thank to "The Max Planck Partner Group" (Max Planck Institute for Chemical Ecology-Jena) for their financial support.

Author contributions
E.A.M.,R.D., C.A.B.C., and A.J.I. wrote the main manuscript text, and E.A.M. prepared figures and tables. All authors reviewed the manuscript. All authors read and approved the final manuscript.

Funding
This work was supported by*Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica(CONCYTEC),Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica(FONDECYT), under contract No. 174-2020-FONDECYT "Doctoral Programs in Peruvian Universities", and the Max-Planck-Gesellschaft "The Max Planck Partner Group" (Max Planck Institute for Chemical Ecology-Jena and the Pontificia Universidad Católica del Perú).*

Availability of data and materials
Not Applicable.

Declarations

Ethics approval and consent to participate
Not Applicable.

Consent for publication
Not Applicable.

Competing interests
We, the authors, declare not to have competing interests as defined by BMC or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Received: 16 January 2023 Accepted: 11 December 2023
Published online: 02 January 2024

References

1. Xu M. Understanding graph embedding methods and their applications. SIAM Rev. 2021;63(4):825–53.
2. Makarov I, Kiselev D, Nikitinsky N, Subelj L. Survey on graph embeddings and their applications to machine learning problems on graphs. PeerJ Comput Sci. 2021;7:357.
3. Park J, Jo J, Yoon S. Mass spectra prediction with structural motif-based graph neural networks. arXiv preprint arXiv:2306.16085 2023.
4. Schmidt AM, Fagerer SR, Jefimovs K, Buettner F, Marro C, Siringil EC, Boehlen KL, Pabst M, Ibánez AJ. Molecular phenotypic profiling of a saccharomyces cerevisiae strain at the single-cell level. Analyst. 2014;139(22):5709–17.
5. Buettner F, Jay K, Wischnewski H, Stadelmann T, Saad S, Jefimovs K, Mansurova M, Gerez J, Azzalin CM, Dechant R, et al. Non-targeted metabolomic approach reveals two distinct types of metabolic responses to telomerase dysfunction in *S. cerevisiae*. Metabolomics. 2017;13(5):1–10.
6. Khazane A, Rider J, Serpe M, Gogoglou A, Hines K, Bruss CB, Serpe R. Deeptrax: embedding graphs of financial transactions. In: 2019 18th IEEE international conference on machine learning and applications (ICMLA). IEEE; 2019. p. 126–33.
7. Xie M, Yin H, Wang H, Xu F, Chen W, Wang S. Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM international on conference on information and knowledge management; 2016. p. 15–24.
8. Ye Y, Hou S, Chen L, Lei J, Wan W, Wang J, Xiong Q, Shao F. Out-of-sample node representation learning for heterogeneous graph in real-time android malware detection. In: 28th International joint conference on artificial intelligence (IJCAI); 2019.
9. Li Y, Yang T. Word embedding for understanding natural language: a survey. In: Guide to big data applications. Berlin: Springer; 2018. p. 83–104.
10. Liu Y, Liu Z, Chua T-S, Sun M. Topical word embeddings. In: Twenty-ninth AAAI conference on artificial intelligence; 2015.
11. Drozd A, Gladkova A, Matsuoka S. Word embeddings, analogies, and machine learning: beyond king–man + woman = queen. In: Proceedings of Coling 2016, the 26th international conference on computational linguistics: technical papers; 2016. p. 3519–30.
12. Orkphol K, Yang W. Word sense disambiguation using cosine similarity collaborates with word2vec and wordnet. Future Internet. 2019;11(5):114.
13. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781; 2013.
14. Zhang A, Lipton ZC, Li M, Smola AJ. Dive into deep learning. arXiv preprint arXiv:2106.11342; 2021.
15. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst. 2013;26:66.
16. Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining; 2014. p. 701–10.
17. Grover A, Leskovec J. node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 855–64.

18. Dong Y, Chawla NV, Swami A. metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining; 2017. p. 135–44.
19. Ribeiro LF, Saverese PH, Figueiredo DR. struc2vec: learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining; 2017. p. 385–94.
20. Goodrich MT, Tamassia R, Goldwasser MH. Data structures and algorithms in python. New York: Wiley; 2013.
21. Lee KD, Lee KD, Steve Hubbard SH. Data structures and algorithms with python. Berlin: Springer; 2015.
22. Goyal P, Ferrara E. Graph embedding techniques, applications, and performance: a survey. Knowl Based Syst. 2018;151:78–94.
23. Cai H, Zheng VW, Chang KC-C. A comprehensive survey of graph embedding: problems, techniques, and applications. IEEE Trans Knowl Data Eng. 2018;30(9):1616–37.
24. Aggarwal M, Murty MN. Machine learning in social networks: embedding nodes, edges, communities, and graphs. Berlin: Springer; 2020.
25. Stamile C, Aldo Marzullo ED. Graph machine learning: take graph data to the next level by applying machine learning techniques and algorithms. Packt Publishing; 2021.
26. Crichton G, Guo Y, Pyysalo S, Korhonen A. Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches. BMC Bioinform. 2018;19(1):1–11.
27. Xiao Z, Deng Y. Graph embedding-based novel protein interaction prediction via higher-order graph convolutional network. PLoS ONE. 2020;15(9):0238915.
28. Su X-R, You Z-H, Hu L, Huang Y-A, Wang Y, Yi H-C. An efficient computational model for large-scale prediction of protein–protein interactions based on accurate and scalable graph embedding. Front Genet. 2021;12: 635451.
29. Zhu J, Zheng Z, Yang M, Fung GPC, Huang C. Protein complexes detection based on semi-supervised network embedding model. IEEE/ACM Trans Comput Biol Bioinf. 2019;18(2):797–803.
30. Li J, Liu Y, Zhang Z, Liu B, Wang Y. Pmdne: prediction of mirna-disease association based on network embedding and network similarity analysis. BioMed Res Int. 2020;2020:66.
31. Bai T, Li Y, Wang Y, Huang L. A hybrid vae based network embedding method for biomedical relation mining. Neural Process Lett. 2021;66:1–12.
32. Luo J, Ouyang W, Shen C, Cai J. Multi-relation graph embedding for predicting mirna-target gene interactions by integrating gene sequence information. IEEE J Biomed Health Inform. 2022;6:66.
33. Basher MA, Rahman A, Hallam SJ. Leveraging heterogeneous network embedding for metabolic pathway prediction. Bioinformatics. 2021;37(6):822–9.
34. Wang M, Wang H, Liu X, Ma X, Wang B, et al. Drug-drug interaction predictions via knowledge graph and text embedding: instrument validation study. JMIR Med Inform. 2021;9(6):28277.
35. Yue X, Wang Z, Huang J, Parthasarathy S, Moosavinasab S, Huang Y, Lin SM, Zhang W, Zhang P, Sun H. Graph embedding on biomedical networks: methods, applications and evaluations. Bioinformatics. 2020;36(4):1241–51.
36. Su C, Tong J, Zhu Y, Cui P, Wang F. Network embedding in biomedical data science. Brief Bioinform. 2020;21(1):182–97.
37. Kruskal JB, Wish M. Multidimensional scaling, vol. 11. London: Sage; 1978.
38. Tenenbaum JB, Silva Vd, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science. 2000;290(5500):2319–23.
39. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. Science. 2000;290(5500):2323–6.
40. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. Adv Neural Inf Process Syst. 2001;66:14.
41. Shaw B, Jebara T. Structure preserving embedding. In: Proceedings of the 26th annual international conference on machine learning; 2009. p. 937–44.
42. Luo D, Ding CH, Nie F, Huang H. Cauchy graph embedding. In: ICML; 2011.
43. Ahmed A, Shervashidze N, Narayanamurthy S, Josifovski V, Smola AJ. Distributed large-scale natural graph factorization. In: Proceedings of the 22nd international conference on World Wide Web; 2013. p. 37–48.
44. Cao S, Lu W, Xu Q. Grarep: learning graph representations with global structural information. In: Proceedings of the 24th ACM international on conference on information and knowledge management; 2015. p. 891–900.
45. Yang C, Liu Z, Zhao D, Sun M, Chang E. Network representation learning with rich text information. In: Twenty-fourth international joint conference on artificial intelligence; 2015.
46. Ou M, Cui P, Pei J, Zhang Z, Zhu W. Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1105–14.
47. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on World Wide Web; 2015. p. 1067–77.
48. Cho H, Berger B, Peng J. Diffusion component analysis: unraveling functional topology in biological networks. In: International conference on research in computational molecular biology. Berlin: Springer; 2015. p. 62–4.
49. Perozzi B, Kulkarni V, Skiena S. Walklets: multiscale graph embeddings for interpretable network classification. arXiv preprint arXiv:1605.02115:043238-23.
50. Li J, Zhu J, Zhang B. Discriminative deep random walk for network classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers); 2016. p. 1004–13.
51. Chen H, Perozzi B, Hu Y, Skiena S. Harp: hierarchical representation learning for networks. In: Proceedings of the AAAI conference on artificial intelligence; 2018. p. 32.
52. Rozemberczki B, Sarkar R. Fast sequence-based embedding with diffusion graphs. In: International workshop on complex networks. Berlin: Springer; 2018. p. 99–107.
53. Rozemberczki B, Davies R, Sarkar R, Sutton C. Gemsec: graph embedding with self clustering. In: Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining; 2019. p. 65–72.

54. Wang D, Cui P, Zhu W. Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1225–34.
55. Cao S, Lu W, Xu Q. Deep neural networks for learning graph representations. In: Proceedings of the AAAI conference on artificial intelligence; 2016. p. 30.
56. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907; 2016.
57. Kipf TN, Welling M. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308; 2016.
58. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. Adv Neural Inf Process Syst. 2017;66:30.
59. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. arXiv preprint arXiv:1710.10903; 2017.
60. Wang H, Wang J, Wang J, Zhao M, Zhang W, Zhang F, Xie X, Guo M. Graphgan: graph representation learning with generative adversarial nets). arXiv; 2017;30(22):11–9.
61. Veličković P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD. Deep graph infomax. arXiv preprint arXiv:1809.10341; 2018.
62. Zeng H, Zhou H, Srivastava A, Kannan R, Prasanna V. Graphsaint: graph sampling based inductive learning method. arXiv preprint arXiv:1907.04931; 2019.
63. Lin Y-Y, Liu T-L, Chen H-T. Semantic manifold learning for image retrieval. In: Proceedings of the 13th annual ACM international conference on multimedia; 2005. p. 249–58.
64. Nickel M, Tresp V, Kriegel H-P. A three-way model for collective learning on multi-relational data. In: ICML; 2011.
65. Jenatton R, Roux N, Bordes A, Obozinski GR. A latent factor model for highly multi-relational data. Adv Neural Inf Process Syst. 2012;25:66.
66. Socher R, Chen D, Manning CD, Ng A. Reasoning with neural tensor networks for knowledge base completion. Adv Neural Inf Process Syst. 2013;26:66.
67. Yang B, Yih W.-t, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 2014.
68. Bordes A, Glorot X, Weston J, Bengio Y. A semantic matching energy function for learning with multi-relational data. Mach Learn. 2014;94(2):233–59.
69. Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmann T, Sun S, Zhang W. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining; 2014. p. 601–10.
70. Yang Z, Tang J, Cohen W. Multi-modal Bayesian embeddings for learning social knowledge graphs. arXiv preprint arXiv:1508.00715; 2015.
71. Nickel M, Rosasco L, Poggio T. Holographic embeddings of knowledge graphs. In: Proceedings of the AAAI conference on artificial intelligence; 2016. p. 30.
72. Ren X, He W, Qu M, Voss CR, Ji H, Han J. Label noise reduction in entity typing by heterogeneous partial-label embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1825–34.
73. Zhang D, Yin J, Zhu X, Zhang C. Homophily, structure, and content augmented network representation learning. In: 2016 IEEE 16th international conference on data mining (ICDM). IEEE; 2016. p. 609–18.
74. Pan S, Wu J, Zhu X, Zhang C, Wang Y. Tri-party deep network representation. Network. 2016;11(9):12.
75. Chen J, Zhang Q, Huang X. Incorporate group information to enhance network embedding. In: Proceedings of the 25th ACM international on conference on information and knowledge management; 2016. p. 1901–4.
76. Tu C, Zhang W, Liu Z, Sun M, et al. Max-margin deepwalk: discriminative learning of network representation. In: IJCAI, vol. 2016; 2016. p. 3889–95.
77. Yang Z, Cohen W, Salakhudinov R. Revisiting semi-supervised learning with graph embeddings. In: International conference on machine learning. PMLR; 2016. p. 40–8.
78. Chen H, Anantharam AR, Skiena S. Deepbrowse: similarity-based browsing through large lists. In: International conference on similarity search and applications. Springer; 2017. p. 300–14.
79. Bordes A, Weston J, Collobert R, Bengio Y. Learning structured embeddings of knowledge bases. In: Twenty-fifth AAAI conference on artificial intelligence; 2011.
80. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. Adv Neural Inf Process Syst. 2013;26:66.
81. Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI conference on artificial intelligence; 2014. p. 28.
82. Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI conference on artificial intelligence; 2015.
83. Ji G, He S, Xu L, Liu K, Zhao J. Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: long papers); 2015. p. 687–96.
84. Feng J, Huang M, Wang M, Zhou M, Hao Y, Zhu X. Knowledge graph embedding by flexible translation. In: Fifteenth international conference on the principles of knowledge representation and reasoning; 2016.
85. Ji G, Liu K, He S, Zhao J. Knowledge graph completion with adaptive sparse transfer matrix. In: Thirtieth AAAI conference on artificial intelligence; 2016.
86. Chang S, Han W, Tang J, Qi G-J, Aggarwal CC, Huang TS. Heterogeneous network embedding via deep architectures. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining; 2015. p. 119–28.D
87. Chen T, Sun Y. Task-guided and path-augmented heterogeneous network embedding for author identification. In: Proceedings of the tenth ACM international conference on web search and data mining; 2017. p. 295–304.
88. Huang Z, Mamoulis N. Heterogeneous information network embedding for meta path based proximity. arXiv preprint arXiv:1701.05291; 2017.

89. Fu X, Zhang J, Meng Z, King I. Magnn: metapath aggregated graph neural network for heterogeneous graph embedding. In: Proceedings of the web conference 2020; 2020. p. 2331–41.
90. Yang L, Xiao Z, Jiang W, Wei Y, Hu Y, Wang H. Dynamic heterogeneous graph embedding using hierarchical attentions. In: European conference on information retrieval. Berlin: Springer; 2020. p. 425–32.
91. Zhou J, Liu L, Wei W, Fan J. Network representation learning: from preprocessing, feature extraction to node embedding. ACM Comput Surv. 2022;55(2):1–35.
92. Manipur I, Manzo M, Granata I, Giordano M, Maddalena L, Guarracino MR. Netpro2vec: a graph embedding framework for biomedical applications. IEEE/ACM Trans Comput Biol Bioinf. 2021;19(2):729–40.
93. Le Q, Mikolov T. Distributed representations of sentences and documents. In: International conference on machine learning. PMLR; 2014. p. 1188–96.
94. Hussein R, Yang D, Cudré-Mauroux P. Are meta-paths necessary? Revisiting heterogeneous graph embeddings. In: Proceedings of the 27th ACM international conference on information and knowledge management; 2018. p. 437–46.
95. Roy I, Velugoti VSBR, Chakrabarti S, De, A. Interpretable neural subgraph matching for graph retrieval. In: Proceedings of the AAAI conference on artificial intelligence, vol. 36; 2022. p. 8115–23.
96. Su X, Hu L, You Z, Hu P, Zhao B. Attention-based knowledge graph representation learning for predicting drug–drug interactions. Brief Bioinform. 2022;23(3):140.
97. Liang X, Li D, Song M, Madden A, Ding Y, Bu Y. Predicting biomedical relationships using the knowledge and graph embedding cascade model. PLoS ONE. 2019;14(6):0218264.
98. Zong N, Wong RSN, Yu Y, Wen A, Huang M, Li N. Drug–target prediction utilizing heterogeneous bio-linked network embeddings. Brief Bioinform. 2021;22(1):568–80.
99. Mohamed SK, Nováček V, Nounu A. Discovering protein drug targets using knowledge graph embeddings. Bioinformatics. 2020;36(2):603–10.
100. Mohamed SK, Nounu A, Nováček V. Biological applications of knowledge graph embedding models. Brief Bioinform. 2021;22(2):1679–93.
101. Chen W, Chen G, Zhao L, Chen CY-C. Predicting drug–target interactions with deep-embedding learning of graphs and sequences. J Phys Chem A. 2021;125(25):5633–42.
102. Long Y, Luo J. Association mining to identify microbe drug interactions based on heterogeneous network embedding representation. IEEE J Biomed Health Inform. 2020;25(1):266–75.
103. He M, Huang C, Liu B, Wang Y, Li J. Factor graph-aggregated heterogeneous network embedding for disease–gene association prediction. BMC Bioinform. 2021;22(1):1–15.
104. Li H, Xiao X, Wu X, Ye L, Ji G. scline: a multi-network integration framework based on network embedding for representation of single-cell rna-seq data. J Biomed Inform. 2021;122: 103899.
105. Gong M, Liu W, Xie Y, Tang Z, Xu M. Heuristic 3d interactive walk for multilayer network embedding. IEEE Trans Knowl Data Eng. 2020;6:66.
106. Ray S, Lall S, Bandyopadhyay S. A deep integrated framework for predicting sars-cov2-human protein–protein interaction. IEEE Trans Emerg Top Comput Intell. 2022;6:66.
107. Gordon DE, Jang GM, Bouhaddou M, Xu J, Obernier K, White KM, O'Meara MJ, Rezelj VV, Guo JZ, Swaney DL, et al. A sars-cov-2 protein interaction map reveals targets for drug repurposing. Nature. 2020;583(7816):459–68.
108. Dick K, Chopra A, Biggar KK, Green JR. Multi-schema computational prediction of the comprehensive sars-cov-2 vs. human interactome. PeerJ. 2021;9:11117.
109. Bakowski MA, Beutler N, Wolff KC, Kirkpatrick MG, Chen E, Nguyen T-TH, Riva L, Shaabani N, Parren M, Ricketts J, et al. Drug repurposing screens identify chemical entities for the development of Covid-19 interventions. Nat Commun. 2021;12(1):1–14.
110. Riva L, Yuan S, Yin X, Martin-Sancho L, Matsunaga N, Pache L, Burgstaller-Muehlbacher S, De Jesus PD, Teriete P, Hull MV, et al. Discovery of sars-cov-2 antiviral drugs through large-scale compound repurposing. Nature. 2020;586(7827):113–9.
111. Zitnik M, Agrawal M, Leskovec J. Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics. 2018;34(13):457–66.
112. Hamilton WL. Graph representation learning. Synth Lectu Artif Intell Mach Learn. 2020;14(3):1–159.
113. Su X, Hu L, You Z, Hu P, Wang L, Zhao B. A deep learning method for repurposing antiviral drugs against new viruses via multi-view nonnegative matrix factorization and its application to sars-cov-2. Brief Bioinform. 2022;23(1):526.
114. Nelson W, Zitnik M, Wang B, Leskovec J, Goldenberg A, Sharan R. To embed or not: network embedding as a paradigm in computational biology. Front Genet. 2019;10:381.
115. Xiong Y, Guo M, Ruan L, Kong X, Tang C, Zhu Y, Wang W. Heterogeneous network embedding enabling accurate disease association predictions. BMC Med Genom. 2019;12(10):1–17.
116. Chen J, Gong Z, Mo J, Wang W, Wang C, Dong X, Liu W, Wu K. Self-training enhanced: network embedding and overlapping community detection with adversarial learning. IEEE Trans Neural Netw Learn Syst. 2021;6:66.
117. Zhang Z, Xiong H, Xu T, Qin C, Zhang L, Chen E. Complex attributed network embedding for medical complication prediction. Knowl Inf Syst. 2022;64(9):2435–56.

Publisher's Note

Chapter 4

Exploratory analysis of metabolic changes using MS data and graph embeddings

This second manuscript was published in the Scientific Reports journal.
Alvarez-Mamani, E., Buettner, F., Beltran-Castañón, C.A. et al. *Exploratory analysis of metabolic changes using mass spectrometry data and graph embeddings*. Scientific Reports 14, 29570 (2024), https://doi.org/10.1038/s41598-024-80955-5.

In our second manuscript, we experimentally evaluated Graph neural network models for exploratory analysis of Mass spectrometry data. The exploratory analysis consisted mainly of: 1) filtering the MS data that we previously represented by a network; 2) similarity analysis, where we compared the metabolomics networks in groups of two phenotypes/genotypes (control vs. other) to identify the change in correlation between the edges of both networks. Identifying these changes are important to understand how different metabolites are interacting in the metabolomic networks and how the behavior of one or more metabolites impacts the behavior of other metabolites. For our experimental study, we use data from an untargeted metabolomics study, so we apply graph neural network models, proposed to solve unsupervised learning tasks. In order to properly evaluate the results, we performed repeated experiments. The collected results demonstrate that our approach is a new strategy for MS data analysis. Where the four GNN models applied, find similar results, so the selection of the best models is done based on the results in filtering, similarity analysis and execution times. Finally, we implemented a web application called `GEMNA`, to perform the exploratory analysis in a more friendly and interactive way. The source code is available on GitHub.

scientific reports

Check for updates

OPEN

Exploratory analysis of metabolic changes using mass spectrometry data and graph embeddings

Edwin Alvarez-Mamani[1,2], Florian Buettner[3,4,5], Cesar A. Beltran-Castanon[1] & Alfredo J. Ibanez[2,6]✉

Mass spectrometry (MS)-based metabolomics analysis is a powerful tool, but it comes with its own set of challenges. The MS workflow involves multiple steps before its interpretation in what is denominate data mining. Data mining consists of a two-step process. First, the MS data is ordered, arranged, and presented for filtering before being analyzed. Second, the filtered and reduced data are analyzed using statistics to remove further variability. This holds true particularly for MS-based untargeted metabolomics studies, which focused on understanding fold changes in metabolic networks. Since the task of filtering and identifying changes from a large dataset is challenging, automated techniques for mining untargeted MS-based metabolomic data are needed. The traditional statistics-based approach tends to overfilter raw data, which may result in the removal of rel vant data and lead to the identification of fewer metabolomic changes. This limitation of the traditional approach underscores the need for a new method. In this work, we present a novel deep learning approach using node embeddings (powered by GNNs), edge embeddings, and anomaly detection algorithm to analyze the data generated by mass spectrometry (MS)-based metabolomics called GEMNA (Graph Embedding-based Metabolomics Network Analysis), for example for an untargeted volatile study on Mentos candy, the data clusters produced by GEMNA were better than the ones used traditional tools, i.e., GEMNA has $silhouette\ score = 0.409$, *vs.* the traditional approach has $silhouette\ score = -0.004$.

Keywords Mass spectrometry, Metabolomic networks, Graph neural networks, Graph embeddings

Mass spectrometry (MS)-based metabolomics analysis is frequently used due to broad analyte coverage, high sensitivity, high selectivity, and high performance[1]. The MS-based metabolomic analytical workfl w involves multiple steps before its interpretation in what is called data mining. The requirement for data mining is that the raw data generated by MS techniques requires normalization and noise filtration to obtain more precise results for data interpretation. Furthermore, when the MS instrument is coupled to other orthogonal analytical techniques such as *Gas chromatography-mass spectrometry* (GC-MS) and *Liquid chromatography-mass spectrometry* (LC-MS), more complex processes such as peak identifi ation and peak alignment are required.

In more detail, the data mining process aims to deconvolute the relevant data, i.e., the data associated with the property of interest (i.e., phenotype) from the rest of the data (i.e., noise based are products of electrical noise, chemical contamination, or any other artifact). Thus, data mining consists of a two-step process. During the fi st step, the MS data is ordered, arranged, and presented for filtering (data reduction) before being analyzed. Subsequently, in the second step, the filtered and reduced data are analyzed using descriptive and inferential statistics to remove further variability. Th s two-step approach facilitates data interpretation[2,3]. Th s holds true particularly for MS-based untargeted metabolomics studies, which focused on understanding fold changes in metabolic networks by monitoring the changes in as many individual metabolites as possible. These changes are studied by comparing at least two types of samples, one denominated baseline and the others in an altered state. The idea of changes in the metabolic network between metabolites due to a stress or mutation induction was developed by[4,5], where the authors proposed monitoring changes in the metabolite network. Since the task of filtering and identifying changes from a large dataset to determine its quality is unrealistic, automated techniques for mining untargeted MS-based metabolomic data are needed. The traditional approach based on

[1]Engineering Department, Pontifici Universidad Catolica del Peru, Lima, Peru. [2]Institute for Omics Sciences and Applied Biotechnology, Pontifi ia Universidad Catolica del Peru, Lima, Peru. [3]Goethe University, Frankfurt, Frankfurt am Main, Germany. [4]German Cancer Consortium (DKTK), Frankfurt am Main, Germany. [5]German Cancer Research Center (DKFZ), Frankfurt am Main, Germany. [6]Science Department, Pontificia Universidad Catolica del Peru, Lima, Peru. ✉email: aibanez@pucp.edu.pe

statistics (ANOVA, t-Test, coeffici t of variation, etc.)[6,7] filter raw data very strongly, and there is the possibility of eliminating relevant data, as a result, identify less metabolomic changes.

In a recent review article[8], we described the use of graph representation learning on biomedical data (i.e., protein-protein interaction networks, and metabolite-metabolite networks). For instance, *Shallow embeddings*-based and *Autoencoders*-based models focus on simpler tasks such as solving link prediction, in terms of biology tasks to fi d a drug-target prediction. Within *Shallow embeddings*-based there are DeepWalk[9], Node2vec[10], and Struc2vec[11]; *Autoencoders*-based: SDNE[12], DNGR[13], etc. More complex problems, such as graph matching, in biology terms to compare two metabolic phenotypes require more advanced models such as *Graph neural network*-based: Variational Graph Autoencoder VGAE[14], Deep Graph Infomax DGI[15], Adversarially Regularized Variational Graph Auto-Encoder ARGVA[16] and Linear Graph Variational Autoencoder LGVAE[17]. The advantage of GNN-based models over the more straightforward graph embedding models is that they are more robust because GNNs can share parameters between nodes (through message passing), work directly on the graphs, and take advantage of the node features and the graph structure[18].

In this work, we present a novel deep learning approach using node embeddings (powered by GNNs), edge embeddings, and anomaly detection algorithm to analyze the data generated by mass spectrometry (MS)-based metabolomics called GEMNA, i.e., Graph Embedding-based Metabolomics Network Analysis (Supplementary Fig. S1 and S2). Although embeddings have been used previously in mass spectrometry, this is the fi st time they have been used for MS-based data filtration, and later identify changes in the filtered metabolic networks. Receiving as input the MS data obtained either by an (i) fl w injection MS or chromatography coupled-MS system, GEMNA identifies the 'real" signals by using embedding filtration couple with a GNN model; and generating as output the filtered MS-based signal list and a dashboard with graphs showing the changes between metabolites among two or more samples.

Our proposal promises to help non-specialized MS researchers in chemistry and biology better understand/visualize their MS results for decision-making and data interpretation. Moreover, in the case of an unknown sample or fi st-time measurement, these steps together represent approximately 90% of the time in an analytical laboratory. Hence, embedding models could accelerate this process, providing life scientists and health researchers with a new toolbox.

Experiments and results

In this section, we evaluate our proposal, on three real datasets containing metabolomic studies and one synthetic dataset. The source code was divided in backend and frontend. The backend was implemented in Django rest framework, with PyTorch Geometric, PyOD libraries, and it is available at https://github.com/win7/GEMNA_Backend.git. Meanwhile, the fronted was implemented in Vue.js with Nuxt framework, and it is available at https://github.com/win7/GEMNA_Frontend.git. The experiments were run on an AMD Ryzen Th eadripper PRO 5955WX computer with 32 cores, 256 GB of RAM, 2x NVIDIA RTX A6000 with 48 GB of VRAM. However, the experiments could be run on a computer with 16 GB of RAM and 8 GB of VRAM. For example, our Mentos dataset took on average 8.45 min to be analyzed using the AMD Ryzen system, while it took on average 12.66 min on the other.

Dataset

The datasets used to perform the experiments are described below. Additionally, further biological details of the real raw data are shown in Table 1.

Synthetic dataset

We generated synthetic data using a multivariate normal distribution (Gaussian distribution), the distribution depends on the mean and the covariance matrix. The mean was in the range of 0 to 100 000 and the covariance matrix values were based on positive semi-defin te matrices. Two datasets were created, each one representing a phenotype. The dataset size was 1 000 samples with 200 features. After that, for the experiments, a number of features were selected to generate groups representing biological and analytical repetitions. Then each phenotype may have 2, 3, 4 or 5 biological repetitions and 2, 3, 4 or 5 analytical repetitions, where an analytical repetition is a feature.

Dataset	Metabolites	Phenotypes	Biological rep.	Analytical rep.
Mutant	6 245	WT	5	40, 40, 40, 40, 40
		PFK1	2	40, 39
		ZWF1	3	40, 40, 40
Leaf	943	Control	3	3, 3, 3
		Treatment	3	3, 3, 3
Mentos	1 782	Orange	2	3, 3
		Red	2	3, 3
		Yellow	2	3, 3

Table 1. Raw data, biological details.

Exploratory analysis of metabolic changes using MS data and graph embeddings

Mutant dataset
The dataset was generated by Buettner et al.[6]. In their publication, the data was acquired using an ESI-MS metabolomics (non-targeted) mass spectrometry-based to study Saccharomyces cerevisiae, auxotrophic, BY4742 strains. The ZWF1 mutant strain lacks one of the reactions from glycolysis into the PPP and the major NADPH-producing step; meanwhile, the PFK1 mutant strain lacks one of the *Phosphofructokinase* enzymes, hence making the glycolysis pathway less effici t. According to the publication[6], cellular extracts of these phenotypes were analyzed on a fl w injection tandem (quadrupole time-of-fli ht) mass spectrometer (Agilent 6520) operating in negative mode scanning from 50 to 1000 m/z at a frequency of 1.4 full spectra per second.

Leaf dataset
The dataset was generated by González-Teuber et al.[7]. The authors used an ESI-MS metabolomics experiment (non-targeted) to evaluate the effects of warming on plant (Aristolochia chilensis) performance (growth, leaf area and chlorophyll) and to estimate the overall changes in leaf metabolites between the ambient control and when treated with the warming simulation. Aristolochia chilensis' leaf samples were obtained from both WT and heat-stress plants in the fi ld at Praderas de lo Aguirre, *Región Metropolitana, Santiago, Chile* ($N = 10$ plants per treatment) by Dr. González-Teuber's team. The Dr. González-Teuber's team collected the plant material following the Pontifi al Catholic University of Chile, Chilean, and international guidelines and legislation. According to the publication[7], leaf samples were shock-frozen in liquid nitrogen immediately after dissection for transportation, lyophilized, and subsequently stored at $-80\,^{\circ}C$. Stock solutions were prepared by grinding $40\,mg$ of leaves under cold conditions. The leaf powder was dissolved in 1 mL of MeOH, vortexed for a period of $5\,min$, and then filtered with a $0.20\,\mu M$ syringe disk filter. Stock solutions were stored at $-20\,^{\circ}C$ until measurement. A dilution solvent, comprising a mixture of MeOH and ddH2O containing 0.5% formic acid (FA; 1:1), was used to dilute stock solutions at a ratio of 1:200. The samples were directly injected into an Orbitrap Q-Exactive HF mass spectrometer (MS; Thermo Fisher Scientific, United States) at a fl w rate of $20\,\mu L/min$. The MS data were acquired at a resolution of 60 000 in positive mode data-dependent acquisition (DDA) with a scan range of $50 - 750\,m/z$. Raw data from this experiment were fi st extracted using an in-house data processing tool built in MatLab (vR2022a). Tentative identifi ation was performed using the Metlin[19] and the mzCloud (https://www.mzcloud.org/) repository.

Mentos dataset
The samples were collected by placing $3\,g$ of one flavor (strawberry, orange, or lemon) of Mentos candy in a $20\,mL$ headspace vial (Thermo Scientific, USA). A $50/30\,\mu m$ DVB/CAR/PDMS coated SPME fiber (Supelco-Aldrich, Bellefonte, PA, USA) was thermally cleaned and desorbed between samples. The fibers were conditioned for $30\,min$ at $250\,^{\circ}C$. Then, the fibers were introduced inside the headspace vial and incubated at $40\,^{\circ}C$ for $30\,min$ with agitation. The SPME fiber was placed into the injector port at $250\,^{\circ}C$ for $3\,min$. Helium carrier gas (99.999%) was set at a constant $1.2\,mL/min$ fl w. The oven temperature was programmed from $40\,^{\circ}C$ (held for $3\,min$) to $250\,^{\circ}C$ at $10\,^{\circ}C/min$ (held for $10\,min$). A DB-5 column ($30\,m \times 250\,\mu m, 0.25\,\mu m$ film thickness; Agilent) was used. The Thermo TSQ 9610 was used as a detector, with the MS transfer line temperature set at $250\,^{\circ}C$ and the ion source (electron impact, $70\,kV$) maintained at $250\,^{\circ}C$. The TSQ operated in full-scan acquisition mode with a mass range of $45 - 1100\,m/z$. The raw data, a comprehensive record of our experiments, were meticulously processed with MS-DIAL, ensuring the accuracy and reliability of our results.

Experiments setup

According to the pipelines, the experiments involve several steps, one of the most relevant steps is the networking filtering through of the node and edge embeddings. Table 2 shows the possible configur tions, each configur tion was run 5 times to determine common subnetworks and the runtimes.

Features	Values
Node embedding	VGAE, LVGAE, ARGVA, DGI
Positional encoding	Laplacian PE, Random Walk PE
Dimensions	2, 3, 4, 8, 16, 32, 64, 128
Optimizer	Adam with $lr = 10^{-2}, weight\ decay = 10^{-4}$
Early stopping	$Patience = 10$
Epochs	100
Edge embedding	Weighted-L2
Outliers detection	COPOD with ($contamination = 0.1$)
Dataset	Synthetic, Mutant, Leaf, Mentos
Network variations	*None* (0:0), *one-to-one* (1:1), *many-to-many* (m:m)

Table 2. Experiments setup. Laplacian eigenvector, Random Walk were applied only on VGAE, LVGAE, ARGVA, DGI

Results

This section is divided into two parts, in the first part we show the influence of the number of biological repeats and analytical repeats on the task of finding a subnetwork. In the second part, we show the results for each proposed pipeline, including the runtimes to generate node embeddings, and edge embeddings.

Influence of raw data

Using our algorithm on our synthetic data set illustrates that the efficiency of network filtering depends on the number of biological repetitions and analytical repetitions for each sample type (phenotype). Fig. 1 shows the number of common edges (i.e., pattern) is influenced by the number of biological repetitions (e.g., 2, 3, 4, or 5) and their respective analytical repetitions (e.g., 2, 3, 4, or 5). While a low number of analytical repetitions show a better coverage (i.e., sensitivity), a higher number of analytical repetitions increases the variability, and more common edges are filtrated, i.e., the common subnetworks (i.e., phenotype pattern) contain fewer common edges (i.e., robustness). This result is similar to the one observed in DNA sequencing. The more measurements one has (depth), the more it is possible to discover single nucleotide polymorphisms. After the experiments with the synthetic dataset, it was concluded that the minimal data set must consist of 2 biological repetitions (each of them with two analytical measurements) to generate partial correlations with coefficients of -1, 1, or *null*. However, for generating correlation coefficients with normal distribution, minimum requirements for 6 data points are needed for one phenotype, Fig. 8(b). Ideally, these 6 data points for one phenotype consist of 2 biological repetitions, each with three analytical measurements. Meanwhile, the greedy (filtering) version finds all common edges, including abnormal edges (outliers).

Pipeline results

In this section, we display and describe the results obtained by each pipeline of our proposal.

Network generation Since the signal filtering will happen using embeddings, it is important that the original networks will be in the same latent space for an optimum comparison. Thus, adding variations to the partial correlation networks, such as linking one common node among biological repeats or linking each common node among biological repeats, forces all embeddings will be present in the same dimensional space. The details of the Synthetic, Leaf, and Mentos networks are shown in Tables 3, 4, and 5, for each biological feature or class (i.e., phenotype) and with their respective network variations (*none*, *one-to-one*, *many-to-many*). For each network, the edge features are partial correlation coefficients.

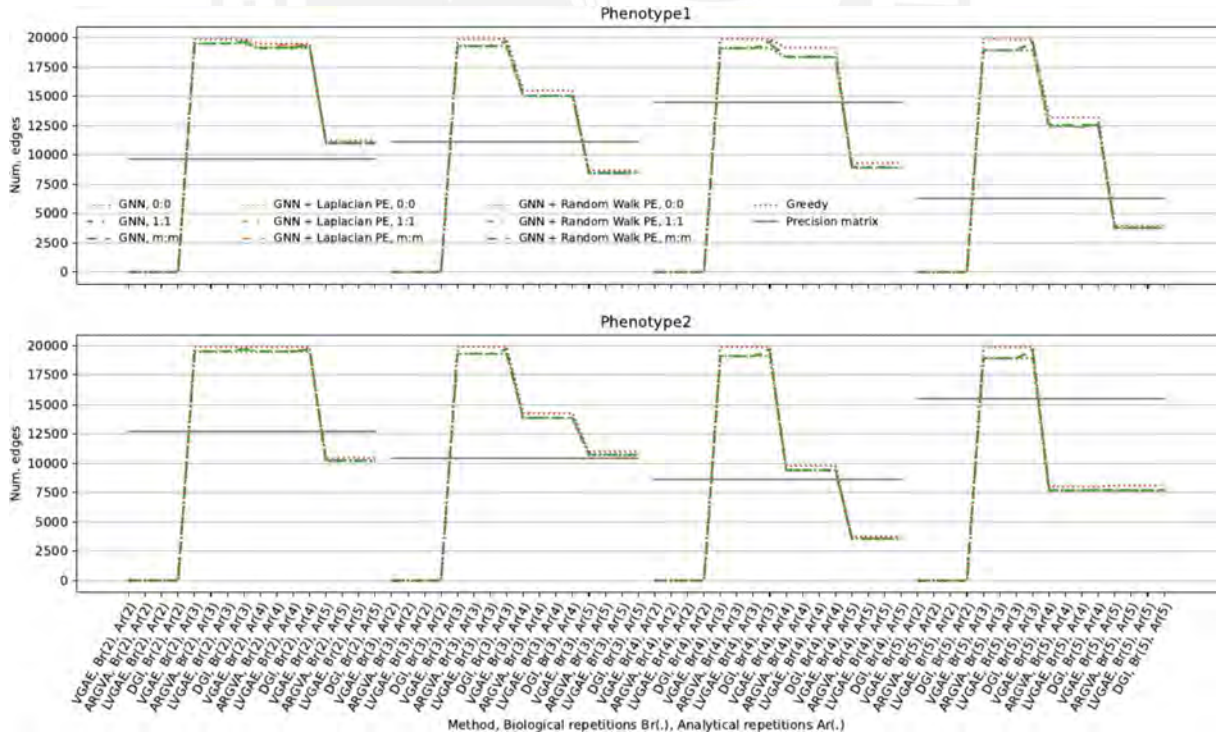


Fig. 1. Performance of biological repetitions (Br) and analytical repetitions (Ar). The red dotted line represents the number of common edges found by a greedy algorithm, which does not eliminate edges considered as noise (outlier). The solid grey line represents the number of common edges based on the inverse of the covariance matrix (Precision matrix).

Exploratory analysis of metabolic changes using MS data and graph embeddings

Phenotype	Variation	Num. nodes	Num. edges	Density	Node feat. (dim.)
Phenotype1	0:0	200	19 900	1.0	Signal intensity (3)
		200	19 900	1.0	Signal intensity (3)
	1:1	400	39 801	0.49	Signal intensity (3)
	m:m	400	40 000	0.50	Signal intensity (3)
Phenotype2	0:0	200	19 900	1.0	Signal intensity (3)
		200	19 900	1.0	Signal intensity (3)
	1:1	400	39 801	0.49	Signal intensity (3)
	m:m	400	40 000	0.50	Signal intensity (3)

Table 3. Synthetic networks details.

Phenotype	Variation	Num. nodes	Num. edges	Density	Node feat. (dim.)
Control	0:0	890	255 685	0.65	Signal intensity (3)
		826	227 941	0.67	Signal intensity (3)
		837	237 457	0.68	Signal intensity (3)
	1:1	2 553	721 085	0.22	Signal intensity (3)
	m:m	2 553	722 709	0.22	Signal intensity (3)
Warming-treated	0:0	877	249 858	0.65	Signal intensity (3)
		825	228 861	0.67	Signal intensity (3)
		851	235 812	0.65	Signal intensity (3)
	1:1	2 553	714 533	0.22	Signal intensity (3)
	m:m	2 553	716 144	0.22	Signal intensity (3)

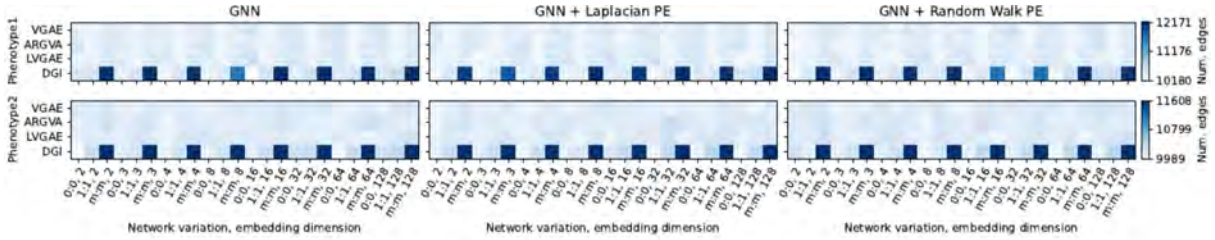
Table 4. Leaf networks details.

Phenotype	Variation	Num. nodes	Num. edges	Density	Node feat. (dim.)
Orange	0:0	1 779	1 098 769	0.69	Signal intensity (3)
		1 771	1 064 854	0.68	Signal intensity (3)
	1:1	3 550	2 163 624	0.34	Signal intensity (3)
	m:m	3 550	2 165 392	0.34	Signal intensity (3)
Red	0:0	1 777	1 066 283	0.68	Signal intensity (3)
		1 772	1 084 732	0.69	Signal intensity (3)
	1:1	3 549	2 151 016	0.34	Signal intensity (3)
	m:m	3 549	2 152 784	0.34	Signal intensity (3)
Yellow	0:0	1 780	1 060 719	0.67	Signal intensity (3)
		1 771	1 053 119	0.67	Signal intensity (3)
	1:1	3 551	2 113 839	0.34	Signal intensity (3)
	m:m	3 551	2 115 608	0.34	Signal intensity (3)

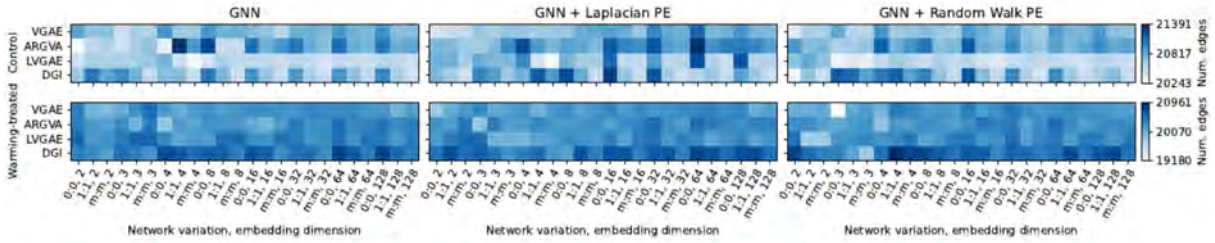
Table 5. Mentos networks details.

Network filtering Traditionally, filtering MS data is based on the CV% of a signal or the peak shape. Here, filtering is focused on obtaining a common subnetwork from the metabolic correlation network (corpus networks) of a phenotype. The filtering process depends essentially on the generation of node embeddings, edge embeddings, and anomaly detection. The Supplementary Fig. S3 displays this process of embeddings manipulation.

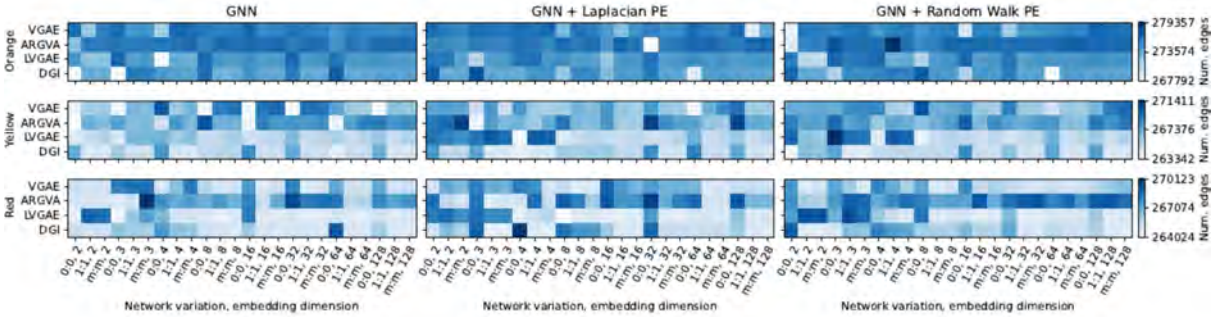
To evaluate the over-smoothing issue in the generation of the node embeddings, we used the Mean Average Distance (MAD) metric[20], in Fig. 3, we show the behavior of Laplacian PE, and RandonWalk PE, a value close to 0, represents that the node embeddings are indistinguishable. In the same way, Fig. 2 shows the number of common edges that could be filtered, according to the features in the Table 2. Thus, from Fig. 2, we can observe that, the *many-to-many* network variation fi d more common edges. Moreover, adding Laplacian PE helps to fi d a more signifi ant number of common edges. Regarding the dimension of embeddings, they are visualized that the results are similar, however, the runtimes to generate the node embeddings and especially to generate the edge embeddings grows according to the dimension of embeddings, where the curve looks like an exponential growth. The plots of the runtimes for the generation of the node embeddings are shown in Fig. 4(a) and for the case of edge embeddings in Fig. 4(b). To select the best model we performed a ranking (see Table 6) with the normalized data of Fig. 2 with *MinMax scaler*, as of result the LVGAE and ARGVA models fi d more



(a) Synthetic networks.



(b) Leaf networks.



(c) Mentos networks.

Fig. 2. Number of common edges in filtering networks.

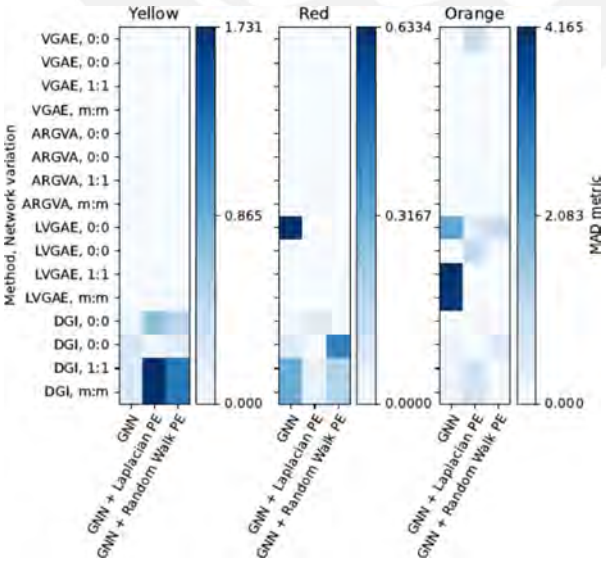
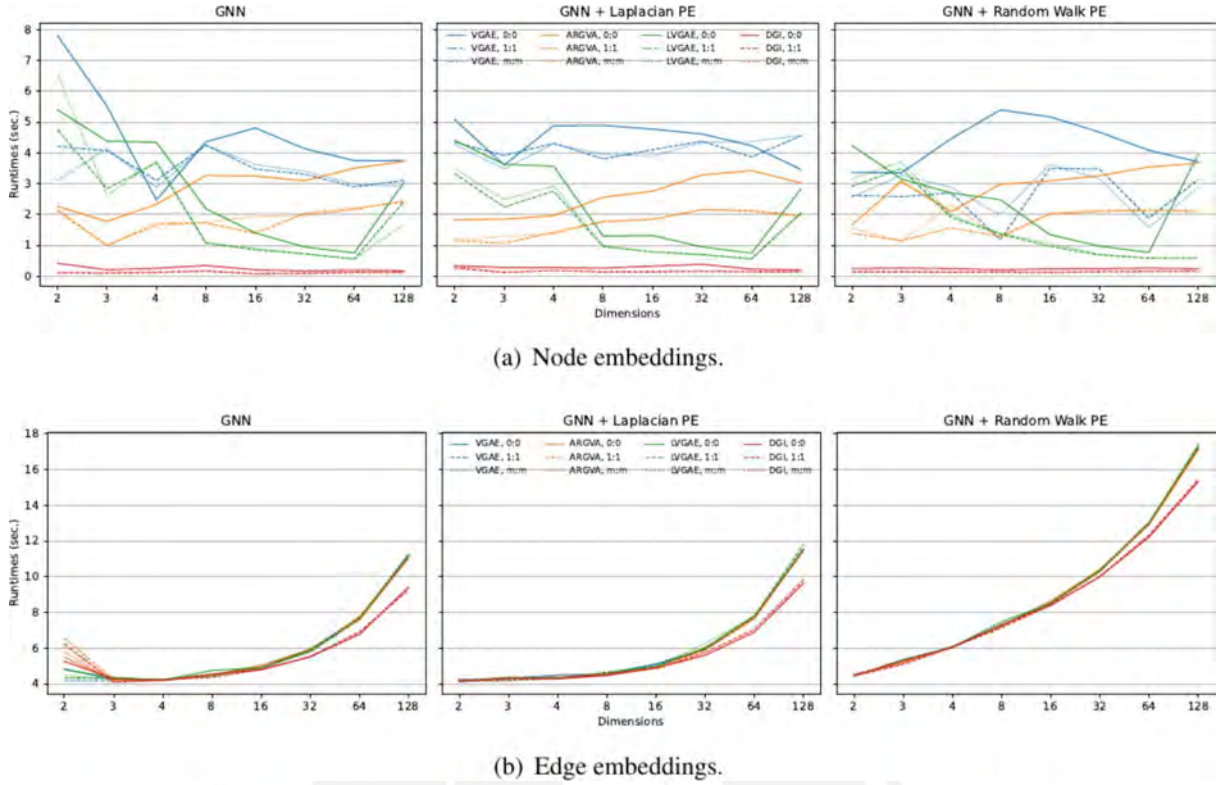


Fig. 3. MAD values on Mentos dataset.



(a) Node embeddings.

(b) Edge embeddings.

Fig. 4. Runtimes to generate embeddings on Synthetic networks.

Num.	GNN	GNN + Laplacian PE	GNN + Random Walk PE
1	ARGVA	DGI	ARGVA
2	DGI	ARGVA	DGI
3	VGAE	LVGAE	LVGAE
4	LVGAE	VGAE	VGAE

Table 6. Ranking of models to fi d common subnetwork.

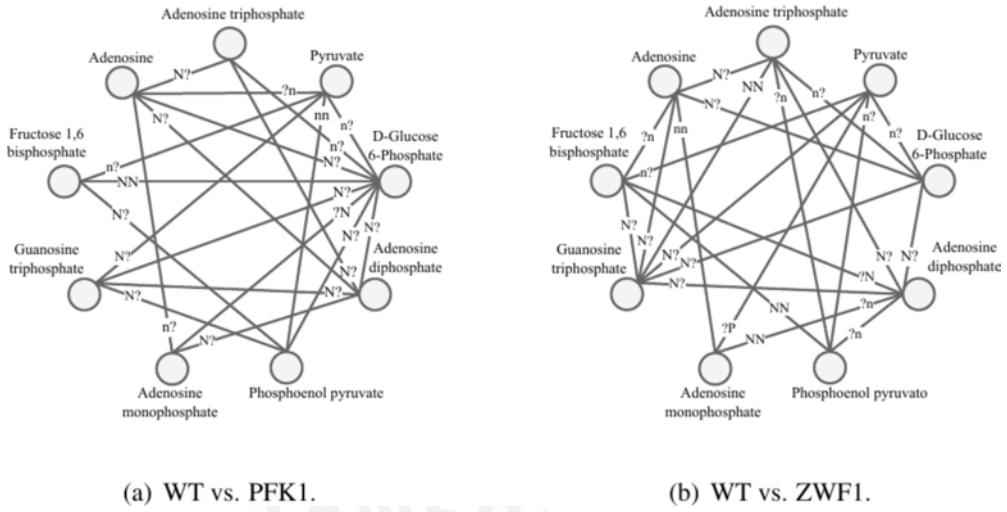
edges in common. By default, our proposal approximately keeps 90% of common edges (it depends on anomaly detection algorithm parameters). Therefore, we expect the GNN models to be close to this percentage, and it keeps a maximum number of common edges.

In summary, to select the best performing configur tion (experiments setup), we consider the configur tion that filters a larger number of edges (see Fig. 2) in a shorter runtime (see Fig. 4). Based on the results, the best choices are the LVGAE and ARGVA models, with a network variation of *many-to-many*, and an embedding dimension of 3.

Similarity analysis Th s fi al step of the GEMNA workfl w is associated to the visualization of the data. The objective of the similarity analysis is to identify the changes between a pair of metabolites in the partial correlation network. Note that one of the networks must be a control network. In Fig. 5 the signifi ant correlation changes (metabolomic changes) for Mutant networks are visualized. The software can also generate heatmaps to visualize changes in terms of the intensity ratio between two metabolites. In Fig. 6 changes in intensity of selected MS signals exhibiting signifi ant differences (One-way ANOVA) between ambient control (C) and warming-treated (W) *Aristolochia chilensis* plants. The Fig. 7 displays that GEMNA identifies more metabolites than standard approaches based on CV%. We performed a systematic quantitative comparison on the Mentos dataset. Here, we quantify the quality of clusters using the Silhouette score, and Intra-cluster distance. GEMNA outperforms traditional methods including, signal intensity, ANOVA, and CV% by a large margin, as reported in Table 7.

Method

The methodology used in network generation, network filtering and similarity analysis (metabolomic changes) between two or more metabolic networks is presented in this section.



(a) WT vs. PFK1. (b) WT vs. ZWF1.

Fig. 5. Similarity analysis on Mutant network.

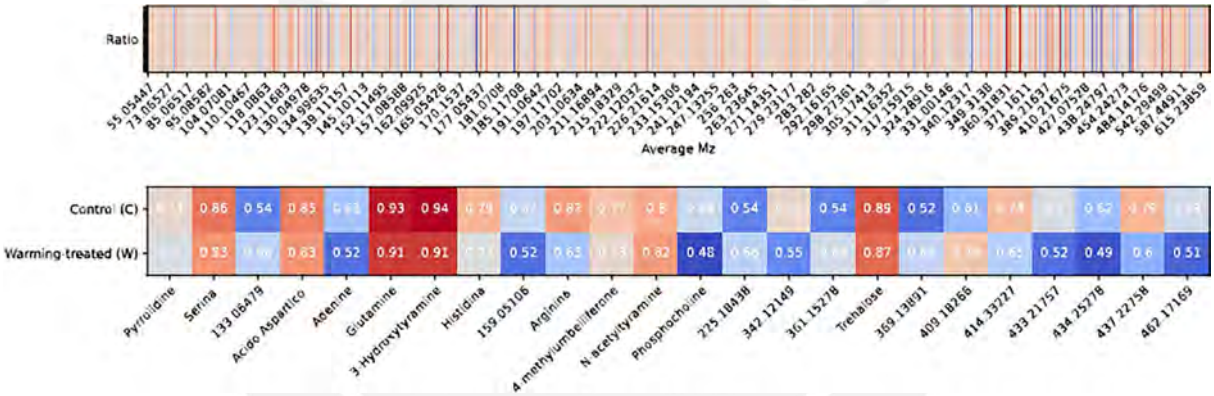


Fig. 6. Heatmap on Leaf filter data.

First, we will describe the format of the mass spectrometry data (raw data), and the requirements needed. The fi st 4 columns of the tables are metadata, while the rest of them are signal intensity (measurements), divided into biological and analytical repetitions for each phenotype, as shown in Fig. 8(a). We expect the data to have at least three analytical repetitions for each biological repetition (see Fig. 8b). If there are only two analytical repetitions, a third column can be added using the bootstrapping technique. Our proposal is described below.

Network generation

The raw data generated after a metabolomics study undergoes preprocessing, where the intensities of the identifi d compounds are aligned. Then from this, an additional preprocessing is applied to build the metabolomics networks. The steps of this pipeline 1 are described below.

i Remove not required variables that will not be used in the network filtering, and similarity analysis steps. The variables we keep are: *Alignment ID*, *Retention time* (if given), *Isotopic molecular ion (m/z)*, *Metabolite name* (if given) and signal intensity. Figure 8a shows the result.
ii Transform measurements by log_{10} (only if necessary).
iii Calculate the partial correlations on the measurements of analytical repetitions.
iv Build networks, the edge between a pair of nodes (metabolites) (v_i, v_j) exists if, $|correlation(v_i, v_j)| \geqslant 0.5$, this condition involves moderate, strong or very strong correlations between those two metabolites influence for the rest of metabolites.The output of this pipeline 1 is a set of partial correlation networks for each phenotype. Figure 9a shows pipeline 1 with the network generation from raw data. Additionally, we add variations on the original networks (see Fig. 9(b)). The variation called *one-to-one* (1:1) consists of joining by an edge to a single node that belongs to the networks of the set of networks. Likewise, the variation called *many-to-many* (m:m) consists of joining by an edge all the nodes with the same identifier within the set of

Exploratory analysis of metabolic changes using MS data and graph embeddings

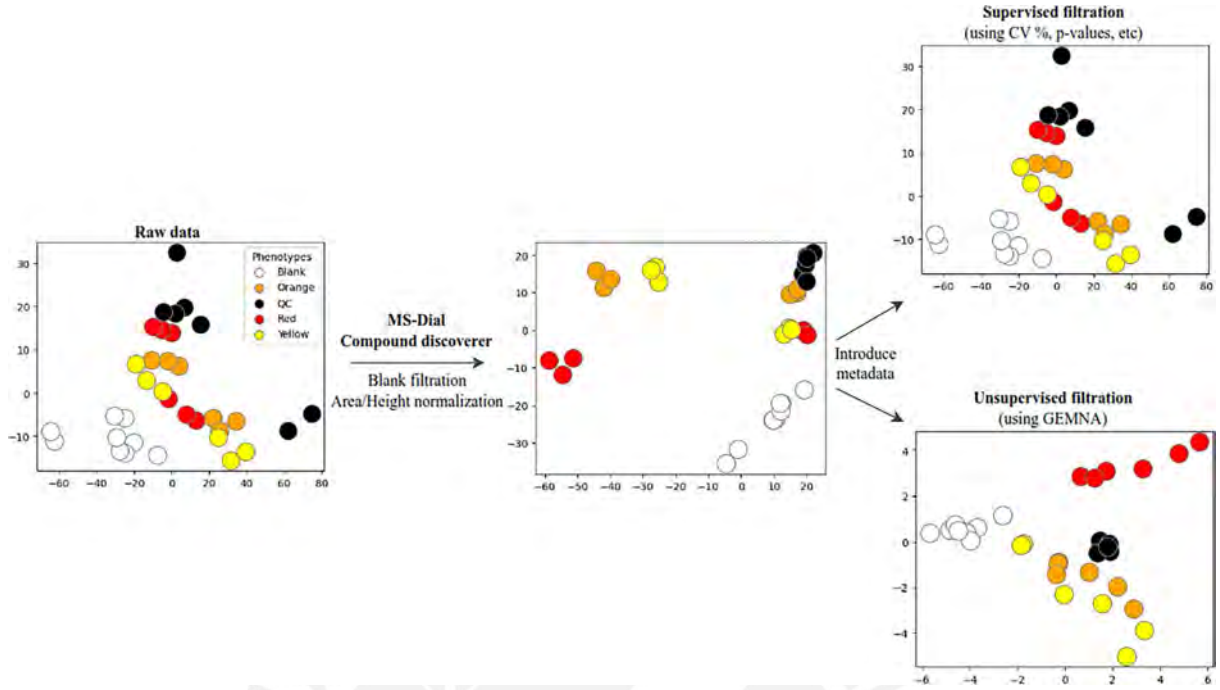


Fig. 7. PCA on Mentos filter data workfl w.

	Filtration (nodes remaining)				Intra-cluster distance				
Methods	**MS-Dial features**	**+ ANOVA (> 0.05%)**	**+ CV % (< 40%)**	**+ GNN**	**Red**	**Orange**	**Yellow**	**Silhouette score**	**Nodes lost**
Baseline	1782	-	–	–	46.32	36.33	27.10	0.242	0
ANOVA	1782	622	–	–	3.96	5.99	9.59	0.154	1160
ANOVA + CV%	1782	622	215	–	17.01	23.88	32.60	− 0.004	1567
GEMNA	1782	-	-	301	2.60	1.96	3.69	0.409	1481

Table 7. Comparison between GEMNA vs. Statistics approach.

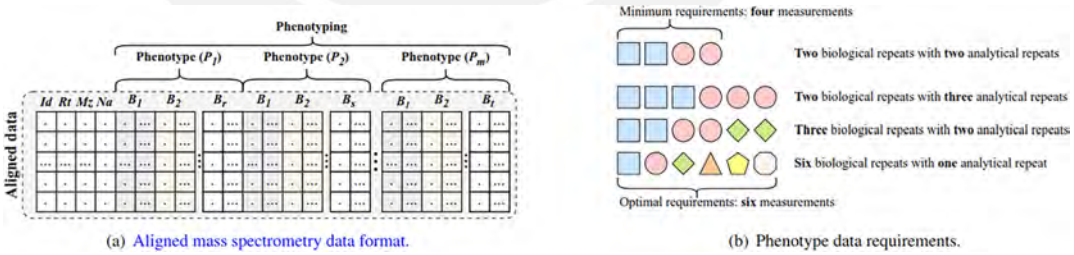


Fig. 8. Mass spectrometry data details. (**a**) $\boldsymbol{Id}$ is Alignment ID, $\boldsymbol{Rt}$ is Retention time, $\boldsymbol{Mz}$ is Average Mz, $\boldsymbol{Na}$ is Metabolite name, B_1, B_2, ..., B_* are ***B***iological repetitions. In addition, each biological repetition has two analytical repetitions at least.

networks, also consecutively. These ideas were found from[11,21] respectively. Adding these variations to the original set of networks helps to bring the node embeddings of each biological repeat to closer latent spaces and, therefore, to fi d anomalies more properly.

Network filterin

It receives as input two or more metabolomic networks of a similar phenotype $C = \{B_1, B_2, ..., B_q\}$, with some network variations, as shown in Fig. 9b. Pipeline 2 includes the following steps.

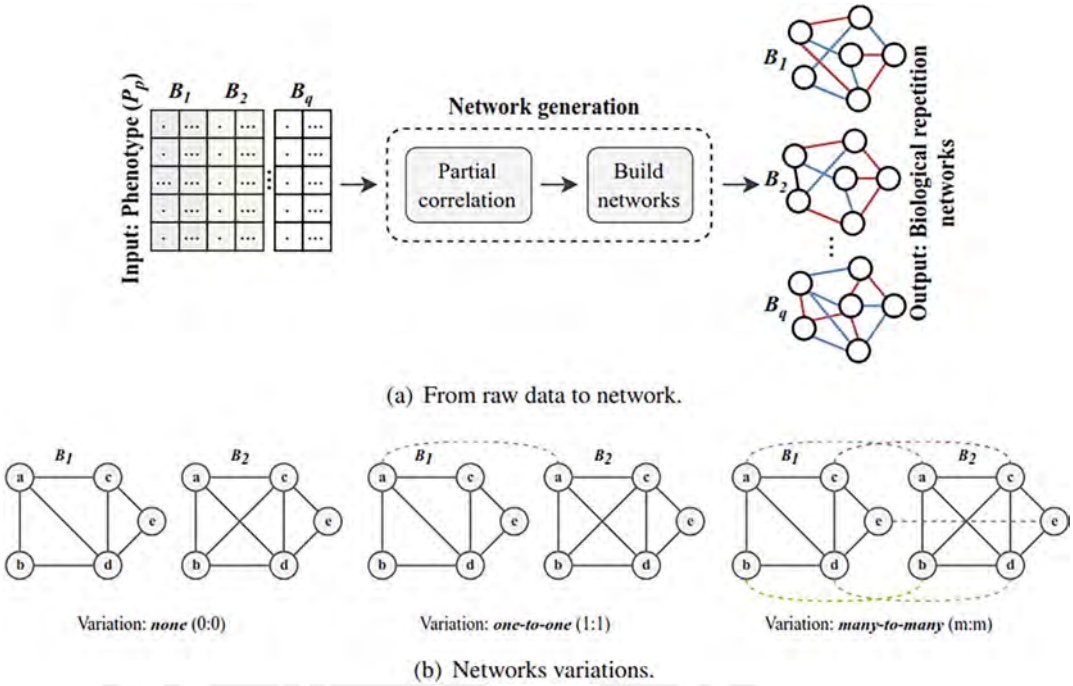


Fig. 9. Pipeline 1: Network generation. Note: ***none*** means, network without variation.

i Generate node embeddings $Z_k, \forall G_k \in C$, using VGAE[14], DGI[15], ARGVA[16] or LGVAE[17] models. The architecture of VGAE, ARGVA, LGVAE contains encoders and decoders. The encoder of VGAE contains two Graph Convolutional Network (GCN)[22] layers, the encoder of LGVAE replaces GCN layers with linear models, the encoder of ARGVA include an adversarial training scheme to regularize the latent space by GCN. On the other hand, DGI has four components: corruption function, encoder based on maximizing local mutual information, readout function, and discriminator. Thus, the decoder is trained to reconstruct the original graph structure. All of them were proposed to unsupervised tasks. Because of, the non-existence of positional information of the nodes; in the experiments we concatenate the node features with two positional encodings (PE), Random Walk PE[23] and Laplacian PE[24] to generate the node embeddings and leverage the GNNs. The positional encodings are a strategy that supports the positional and structural representation of GNNs.

ii Calculate edge embeddings $\hat{Z}_k, \forall G_k \in C$, using Weighted-L2 operator, defi ed as: $|z_i - z_j|^2$.

iii Concatenate (only for 0:0 network variation) and detect outliers (anomalous edges), according to edge embeddings. Therefore, $\hat{Z}_{concat} = \hat{Z}_1 \oplus \hat{Z}_2 \oplus ... \oplus \hat{Z}_q$ and $inliers(\hat{Z}_{concat}) = \hat{Z}_{concat} - outliers(\hat{Z}_{concat})$, using Copula-Based Outlier Detection (COPOD) algorithm[25]. Formally, a copula is the cumulative distribution function (CDF) of a vector with uniform marginals on the interval [0, 1]. COPOD, includes three steps, 1) calculate the CDFs of the dataset, 2) use the empirical CDFs to generate the cupula function, 3) use the empirical copula to approximate the tail probability.The output of this pipeline is a filtered network, i.e., a common subnetwork for a biological network of one phenotype (sample type). Note: Th s common phenotype can be built by focusing on sensitivity (optimizing for the maximum number of nodes) or robustness (optimizing for the most stable interactions between nodes), for both cases using the common edges obtained. Pipeline 2 is shown in Fig. 10.

Similarity analysis

It receives as input two filtered metabolomic networks F_1, F_2 obtained in pipeline 2 and consists of the following steps, as shown in Fig. 11(a).

i Operations on networks via union of F_1 and F_2 into a single network R_o.

ii Determine the change in correlation (metabolomic change) over the network R_o, based on a pair of filtered networks. To determine whether two correlations are signifi antly different; fi st, transform the correlations into fisher Z-scores; second, calculate the standard error of the difference of the Z-scores, then calculate the ratio between the difference and the standard error; fi ally, compare this ratio to a standard normal distribution. The p-values for each change of correlations are obtained after the calculation[26].The output of this pipeline is a directed network R_o with labelled edges with their own correlation changes (metabolomic changes). The network R_o can be divided into two subnetworks, the fi st subnetwork R'_o with signifi antly different correlations (p-value < 0.05 or label contains the "?" mark) and the second subnetwork R''_o with non-signifi antly different correlations (p-value $\geqslant 0.05$ and labels with identical letters). The labeling of each

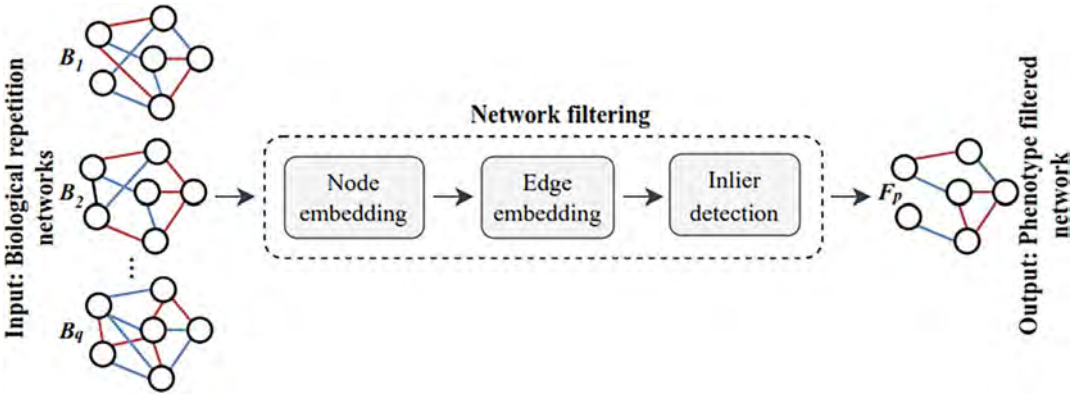


Fig. 10. Pipeline 2: Network filtering. Note: The edges in red color have positive correlation and the edges in blue color have negative correlation.

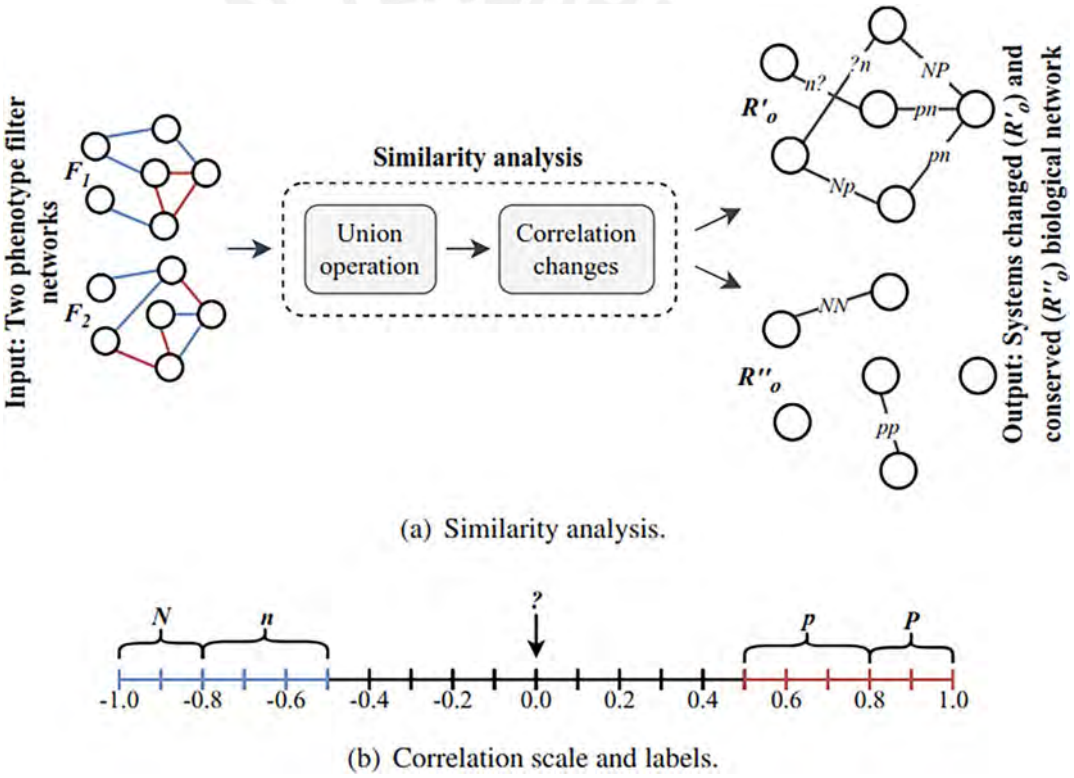


Fig. 11. Pipeline 3: Similarity analysis. Note: the "?" mark means that there is no correlation.

edge $(v_i, v_j) \in R_o$ is obtained according to a threshold (th) and the weights $w_{1_{i,j}} \in F_1$ and $w_{2_{i,j}} \in F_2$. We consider a $th = 0.8$, then, the high negative correlation (N) is in $[-1.0, -th]$, the moderate negative correlation (n) is in $\langle -th, -0.5]$. In contrast, the moderate positive correlation (p) is in $[0.5, th\rangle$, the high positive correlation (P) is in $[th, 1.0]$. For example, given an edge (v_1, v_2) with weights $w_{1_{1,2}} = 0.6$ and $w_{2_{1,2}} = -0.9$, then, according to the scale in Fig. 11(b), the label for the edge is $pN\,(v_1 \xrightarrow{pN} v_2)$.

In summary, all steps are visualized in Fig. 12, from the network generation of the biological repetitions with their respective analytical repetitions, to obtaining the biological network systems with their respective metabolomic changes. The toy example in Fig. 12 has three phenotypes, P_1, P_2, P_3 with P_2 considered as the control. These phenotypes pass through the network generation (pipeline 1), after that, they are filtered (pipeline 2) and fi ally, similarity analysis (pipeline 3) to identify signifi ant changes in the biological networks, it is performed for each

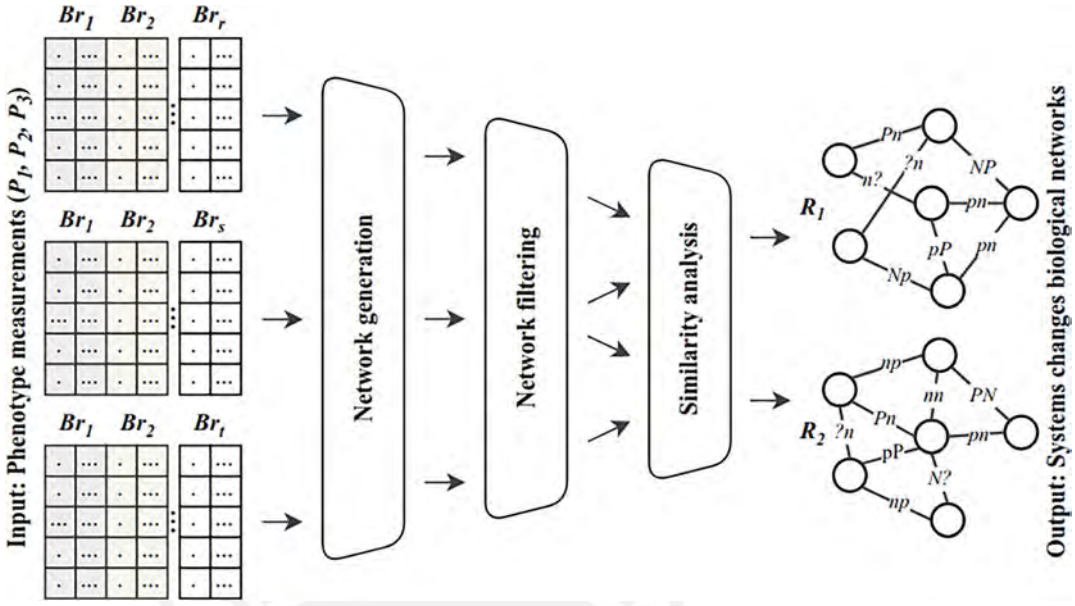


Fig. 12. Our method with a toy example. Note: Br_1, Br_2, ..., Br_* are ***B***iological ***r***epetitions.

phenotype regarding the control, in this case P_1 with P_2 and P_3 with P_2. The entire pipeline was implemented in a web application called GEMNA (Supplementary Fig. S1 and S2).

Discussion

Due to the large datasets produced in traditional mass spectrometry-based experiments, specialized software is always required to identify better and interpret the results. The current metabolomics software uses normalization tools to eliminate confounding factors, particularly the so-called batch effects. Normalizing the peaks is usually performed using a regression curve (e.g., LOESS) or Random forest (SERRF) algorithm to reduce these co-founding factors. These normalization techniques allow for the reduction of batch variations or amplitude drifts. However, this is not always enough. More advanced software uses coeffici t of variations between analytical repeats or peak shape profiles. Th s strategy assumes that MS noise is a mixture of "real (unique)" signals and "not interesting" signals. The latter combines redundant signals (i.e., fragments or clusters of the unique signals), contaminants, and electronic noise. Therefore, since contaminants and electronic noise are not strongly correlated to "real (unique)" signals, these signals could be filtered using a combination of embedding algorithms.

In addition, GEMNA minimizes MS noise by removing those signals whose interaction with the rest (correlation network) has the most variability. Hence, we are improving the output data quality by zooming in on the signals with the most robust interactions. Furthermore, GEMNA also facilitates data interpretation for non-MS experts by introducing a graphic user interface with simple data visualization tools. Hence, once the most effici t parameters have been identifi d for building the embeddings (pipeline 1), and subsequently, the network was filtered from the signals with the highest variability (pipeline 2), the software allows the researcher to visualize the data under different formats.

For example, in the similarity analysis for different phenotypes of *S. cerevisiae* measured with ESI-MS[6], the differences between the metabolic networks formed by the WT (control) and PFK1 properties can easily be observed in Fig. 5(a). Here, it is noted that for the WT phenotype, there was a strong negative correlation (*N*) between the metabolites *Fructose 1; 6-bisphosphate* and *Phosphoenol pyruvate* as it would be expected. Th s correlation signifi antly differs from the one observed in the PFK1 system, where one PFK enzyme is removed, while it remains slightly changed in the ZWF1 system, i.e., the algorithm shows that the strong negative correlation stays, but the magnitude of the correlation is (statistically) signifi antly different. On the Leaf dataset measured using an ESI-MS instrument[7], the Fig. 6 shows ratio of change between warming-treated and WT plants. The zoom version of the figu e highlights the signals with the highest variation, among them are selected amino acids as reported in[7]. In summary, the software is an open-source web application, which is compatible with any mass spectrometry experiments and can contribute non-specialists to better filtrate their data from MS noise and visualize the results in a friendly user environment.

Another advantage of GEMNA is its ability to improve clustering-based analysis, such as PCA. Figure 7, and Table 7 showcases the results of GEMNA ($silhouette\ score = 0.409$) *vs.* the traditional approach ($silhouette\ score = -0.004$) for noise filtration after peak normalization in a headspace analysis of Mentos using a GC-MS instrument. Moreover, after the network filtering pipeline over Mentos, GEMNA on average obtains 301 nodes (metabolites), while the statistical approach only obtains 215. As in many untargeted metabolomics systems, PCA is difficult to interpret due to the presence of confounding factors, in particular, the

so-called batch effects and MS noise. Here, the graph neural network (graph embedding AI) can automatically choose relevant signals and produce a better PCA to the one obtained by manually choosing the signals of interest after removing "noisy peaks" based on their CV% and peak shape.

Conclusion

We propose GEMNA, a novel open-source tool for metabolic study based on node embeddings, edge embeddings, and anomaly detection. GEMNA allows filtering raw data and fi ds signifi ant metabolomic changes between two or more metabolic networks. Experiments report that in network filtering, on average 90% of edges in common are kept, the others are removed by the anomaly detection algorithm. The runtime is related to the dimension of the embeddings, i.e., a smaller dimension needs a shorter runtime.

In conclusion, the most competitive models were LVGAE and ARGVA, the best network variation was *many-to-many*, and the dimension of the embeddings was 3. In addition, the minimum requirement for the good performance of these models is to have two or more phenotypes, which one of them is the control. Furthermore, each phenotype (and control) consists of at least two biological repetitions, and each with at least 3 measurements.

Data availibility

The MS files for the Mutant and Leaf (Aristolochia chilensis) datasets can be found in their original articles ([6,7]). The MS files for the Mentos dataset can be downloaded from the following link: https://drive.google.com/drive/folders/11jFi1AxMmjHFPqETv9VLPyEg9nVTZVna

Received: 30 August 2024; Accepted: 22 November 2024
Published online: 28 November 2024

References

1. Liebal, U. W., Phan, A. N., Sudhakar, M., Raman, K. & Blank, L. M. Machine learning applications for mass spectrometry-based metabolomics. *Metabolites* **10**, 243 (2020).
2. Fan, S. et al. Systematic error removal using random forest for normalizing large-scale untargeted lipidomics data. *Anal. Chem.* **91**, 3590–3596 (2019).
3. Bahado-Singh, R. O. et al. Artific al intelligence and the analysis of multi-platform metabolomics data for the detection of intrauterine growth restriction. *PLoS ONE* **14**, e0214121 (2019).
4. Sauer, U. & Zamboni, N. From biomarkers to integrated network responses. *Nat. Biotechnol.* **26**, 1090–1092 (2008).
5. Patti, G. J., Yanes, O. & Siuzdak, G. Metabolomics: the apogee of the omics trilogy. *Nat. Rev. Mol. Cell Biol.* **13**, 263–269 (2012).
6. Buettner, F. et al. Non-targeted metabolomic approach reveals two distinct types of metabolic responses to telomerase dysfunction in S. cerevisiae. *Metabolomics* **13**, 1–10 (2017).
7. González-Teuber, M., Palma-Onetto, V., Aguirre, C., Ibáñez, A. J. & Mithöfer, A. Climate change-related warming-induced shifts in leaf chemical traits favor nutrition of the specialist herbivore battus polydamas archidamas. *Front. Ecol. Evol.* **11**, 1152489 (2023).
8. Alvarez-Mamani, E., Dechant, R., Beltran-Castañón, C. A. & Ibáñez, A. J. Graph embedding on mass spectrometry-and sequencing-based biomedical data. *BMC Bioinformatics* **25**, 1 (2024).
9. Perozzi, B., Al-Rfou, R. & Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 701–710 (2014).
10. Grover, A. & Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864 (2016).
11. Ribeiro, L. F., Saverese, P. H. & Figueiredo, D. R. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 385–394 (2017).
12. Wang, D., Cui, P. & Zhu, W. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 1225–1234 (2016).
13. Cao, S., Lu, W. & Xu, Q. Deep neural networks for learning graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30 (2016).
14. Kipf, T. N. & Welling, M. Variational graph auto-encoders. Preprint at arXiv:1611.07308 (2016).
15. Veličković, P. *et al.* Deep graph infomax. arXiv preprint arXiv:1809.10341 (2018).
16. Pan, S. *et al.* Adversarially regularized graph autoencoder for graph embedding. Preprint at arXiv:1802.04407 (2018).
17. Salha, G., Hennequin, R. & Vazirgiannis, M. Simple and effective graph autoencoders with one-hop linear models. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, 319–334 (Springer, 2021).
18. Hamilton, W. L. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning* **14**, 1–159 (2020).
19. Smith, C. A. et al. Metlin: a metabolite mass spectral database. *Ther. Drug Monit.* **27**, 747–751 (2005).
20. Chen, D. et al. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. *Proc. AAAI Conf. Artif. Intell.* **34**, 3438–3445 (2020).
21. Kazemi, S. M. *Dynamic Graph Neural Networks* 323–349 (Springer Nature Singapore, 2022).
22. Kipf, T. N. & Welling, M. Semi-supervised classifi ation with graph convolutional networks. Preprint at arXiv:1609.02907 (2016).
23. Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y. & Bresson, X. Graph neural networks with learnable structural and positional representations. Preprint at arXiv:2110.07875 (2021).
24. Dwivedi, V. P. et al. Benchmarking graph neural networks. *J. Mach. Learn. Res.* **24**, 1–48 (2023).
25. Li, Z., Zhao, Y., Botta, N., Ionescu, C. & Hu, X. Copod: copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)* (ed. Li, Z.) 1118–1123 (IEEE, 2020).
26. Snedecor, G. W. G. W. & Cochran, W. G. W. G. *Statistical methods / [by] George W. Snedecor, William G. Cochran.* 7th Edn. (Iowa State University Press, Ames, Iowa, 1980).

Acknowledgements

E.A.M. doctoral studies are funded by *Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica* (CONCYTEC), and *Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica* (FONDECYT), under contract No. 174-2020-FONDECYT "Doctoral Programs in Peruvian Universities". A.J.I. thank to *The Max Planck Partner Group* (Max Planck Institute for Chemical Ecology-Jena), and the *Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica* (CONCYTEC-Prociencia funding call E041-2024-01; N°

PE501086715-2024-PROCIENCIA) for their financial support. We also thank Prof. Dr. Marcia González-Teuber (Pontific a Universidad Católica de Chile) for the Aristolochia chilensis' leaf dataset, and PD Dr. Reinhard Dechant (Calico) & Dr. Madina Mansurova (PUCP) for their valuable recommendations and review of this work.

Author contributions

E.A.M. and A.J.I. designed the experiments, E.A.M. ran the experiments, E.A.M., F.B., C.A.B.C., and A.J.I. wrote the main manuscript text, and E.A.M. prepared figures and tables. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-80955-5.

Correspondence and requests for materials should be addressed to A.J.I.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Chapter 5

Conclusions and future works

5.1 Conclusions

In this work, we first review the data generated by the analytical technique of mass spectrometry and the traditional algorithms (based on statistics) for the analysis of this type of data. We noticed the disadvantages of statistical methods and investigated new approaches based on machine learning for the analysis of mass spectrometry data. As a result of this first approach, we found models based on graph embeddings. These models capture the information contained in metabolomic networks (formed by mass spectrometry data) at the level of nodes, edges, or graph/subgraphs and represent them in a d-dimensional vector called node-, edge-, or graph/subgraph embeddings respectively. Typically, these models solve node classification, link prediction and community detection tasks. The latter task was used as a starting point for the analysis of the mass spectrometry data. Furthermore, in the state-of-the-art we found that these models are used for different applications, for example in the field of chemistry and biology, where there are several data sources including genomics, proteomics, metabolomics.

In the second part, we designed a pipeline for a metabolomic study using node embeddings (based on GNN), edge embeddings, and anomaly detection algorithm. The study consisted of, filters raw data and identifies significant metabolomic changes across multiple metabolic networks. Our proposal, called `GEMNA`, is a new approach to the exploratory analysis of metabolic networks. Experiments show that our approach is an improvement on traditional statistical-based techniques. For example, for the Mentos candy dataset, the data clusters produced by `GEMNA` were better than the ones used in traditional techniques based on statistics; `GEMNA` has $silhouette\,score = 0.409$, *vs* the traditional approach has $silhouette\,score = -0.004$. Finally, we developed an interactive web application to facilitate metabolomics studies.

5.2 Future work

Mass spectrometry data are known to be very complex data. Because of this, it is important to continue research on new strategies for the analysis of data generated by untargeted metabolomics studies. We propose to investigate Geometric Graph Neural Networks (GGNN) for the analysis of metabolomic networks. Compared to GNNs, GGNNs generate node embeddings using a feature tensor features, e.g.s the node coordinates in the $\mathbb{R}^3$ space.

References

[1] Aggarwal, M. and Murty, M. N. (2020). *Machine learning in social networks: embedding nodes, edges, communities, and graphs*. Springer Nature.

[2] Awan, M. G. and Saeed, F. (2016). Ms-reduce: an ultrafast technique for reduction of big mass spectrometry data for high-throughput processing. *Bioinformatics*, 32(10):1518–1526.

[3] Cai, H., Zheng, V. W., and Chang, K. C.-C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.

[4] Goodrich, M. T., Tamassia, R., and Goldwasser, M. H. (2013). *Data structures and algorithms in Python*. John Wiley & Sons Ltd.

[5] Goyal, P. and Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94.

[6] Labonne, M. (2023). *Hands-On Graph Neural Networks Using Python: Practical techniques and architectures for building powerful graph and deep learning apps with PyTorch*. Packt Publishing Ltd.

[7] Lee, K. D., Lee, K. D., and Steve Hubbard, S. H. (2015). *Data Structures and Algorithms with Python*. Springer.

[8] Liu, Z. and Zhou, J. (2022). *Introduction to graph neural networks*. Springer Nature.

[9] Makarov, I., Kiselev, D., Nikitinsky, N., and Subelj, L. (2021). Survey on graph embeddings and their applications to machine learning problems on graphs. *PeerJ Computer Science*, 7:e357.

[10] Mujezinovic, N., Schneider, G., Wildpaner, M., Mechtler, K., and Eisenhaber, F. (2010). Reducing the haystack to find the needle: improved protein identification after fast elimination of non-interpretable peptide ms/ms spectra and noise reduction. *BMC genomics*, 11:1–8.

[11] Nollet, L. and Winkler, R. (2022). *Mass Spectrometry in Food Analysis*. Food Analysis and Properties Series. CRC Press, Taylor & Francis Group.

[12] Putri, S. and Fukusaki, E. (2014). *Mass Spectrometry-Based Metabolomics: A Practical Guide*. Taylor & Francis.

[13] Raftery, D., Press, H., and Media, S. S. (2014). *Mass Spectrometry in Metabolomics: Methods and Protocols*. Methods in Molecular Biology. Springer New York.

[14] Saoub, K. R. (2021). *Graph Theory: an introduction to proofs, algorithms, and applications*. Chapman and Hall/CRC.

[15] Sindelar, M. and Patti, G. J. (2020). Chemical discovery in the era of metabolomics. *Journal of the American Chemical Society*, 142(20):9097–9105.

[16] Smoluch, M., Grasso, G., Suder, P., and Silberring, J. (2019). *Mass Spectrometry: An Applied Approach*. Wiley Series on Mass Spectrometry. Wiley.

[17] Stamile, C., Marzullo, A., and Deusebio, E. (2021). *Graph Machine Learning: Take graph data to the next level by applying machine learning techniques and algorithms*. Packt Publishing Ltd.

[18] Thompson, J. (2017). *Mass Spectrometry*. Jenny Stanford Publishing.

[19] Watson, J. T. and Sparkman, O. D. (2007). *Introduction to mass spectrometry: instrumentation, applications, and strategies for data interpretation*. John Wiley & Sons.

[20] Xu, M. (2021). Understanding graph embedding methods and their applications. *SIAM Review*, 63(4):825–853.

Appendix A

User guide of GEMNA web application

GEMNA, is an open source web application. A user guide is presented below. The user guide consists of two sections. The first section displays the form wizard to load the raw data, and choose the parameters to perform the processing and analysis. The second section displays a visualization panel of the networks, heatmap, in order to observe the metabolomic changes. In addition, it contains forms for more specific queries on the results.

GEMNA
Guide User

Version 1.0.2

win7

November 27, 2024

Contents

1 Introduction

The guide is divided into two parts; the first part (Sec. 2) deals with processing and analyzing mass spectrometry data, and the second (Sec. 3) deals with visualizing and finding patterns in the results. For this guide, we use the Mutant dataset containing three phenotypes: WT, pck1, and zwf1.

The source code was divided in backend y fronted. The backend was implemented in Django rest framework, with PyTorch Geometric, PyOD libraries, and it is available at `https://github.com/win7/GEMNA_Backend.git`. On the other hand, the fronted was implemented in Vue.js with Nuxt framework, and it is available at `https://github.com/win7/GEMNA_Frontend.git`.

2 Process and analysis

2.1 Format input

The format of the input file must be in the *.csv* format, and the "|" character delimits the columns. The first four columns are the metadata, and the remaining are the intensity measurements. An example is shown in Figure 1.

	Alignment ID	Average Rt	Average Mz	Metabolite name	WT_3.3.3	WT_5.1.9	WT_5.2.4	WT_5.2.8	WT_3.1.5	WT_3.4.4	...	pck1_1.1.6	pck1_2.2.9	pck1_2.1.2	pck1_1.4.3	zwf1_1.2.7	zwf1_3.2.5	zwf1_3.1.2	zwf1_3.3.6	zwf1_1.4.2
0	1	0	59.0049	Unknown	169	286	575	340	939	410	...	913	640	1078	639	466	502	184	1008	851
1	2	0	59.0137	Unknown	48964	60211	195240	148489	81334	54320	...	69318	162356	135568	118566	51114	84501	66010	99974	129520
2	3	0	59.0291	Unknown	1553	2288	7911	5562	4064	2006	...	3124	6693	5744	4682	2067	3571	2798	4384	5416
3	4	0	59.0370	Unknown	1	257	1262	1012	1247	186	...	616	1549	1205	961	553	913	448	902	1195
4	5	0	59.0453	Unknown	1	112	321	1	634	65	...	217	624	699	399	234	167	212	330	398
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6240	6241	0	996.5509	Unknown	1745	1280	2179	2526	1805	2533	...	2536	3262	2728	2512	2103	2031	2844	2479	2639
6241	6242	0	996.7096	Unknown	1593	361	979	934	1766	1758	...	2092	1507	2126	2590	2188	3212	3354	4749	2695
6242	6243	0	997.5542	Unknown	1724	982	2727	2481	2318	2253	...	2729	1543	2266	2259	2063	2206	3114	2799	2499
6243	6244	0	997.7131	Unknown	1490	711	968	691	1319	1660	...	1719	216	1971	1449	863	1623	2412	2773	1302
6244	6245	0	998.4845	Unknown	2412	856	2737	2287	1875	0	...	3187	3000	3067	2161	1104	2486	2707	1690	2194

Figure 1: Aligned mass spectrometry data format.

2.2 Process mass spectrometry data

2.2.1 Load your data

The form (see Figure 2) requires an e-mail address, to which you will receive a code to be used later in the visualization section.

2.2.2 Parameters selection

The form (see Figure 3) requires the choice of the following parameters: the method to generate the node embeddings (the best `LVGAE`), network

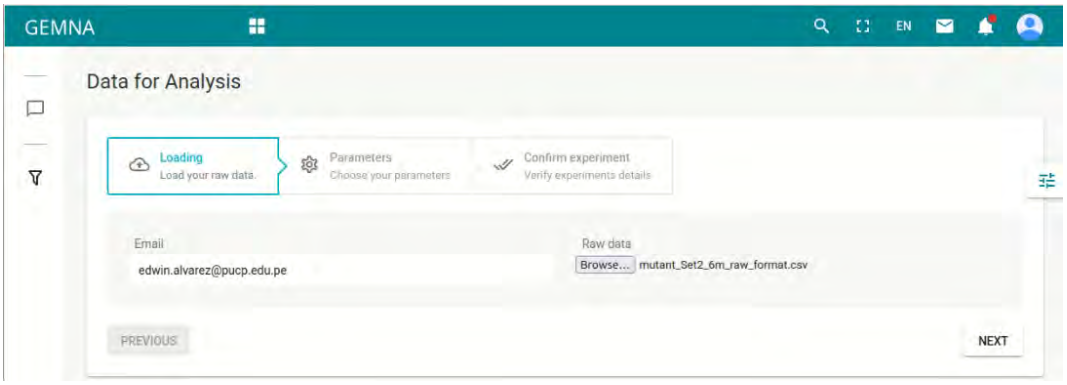


Figure 2: Load your data.

variation (the best many-to-many), dimension of the embeddings (the best 3), control (for this dataset is WT), the data has no transformation, the correlation threshold to generate the networks (default 0.5), and the threshold to filter the significant changes between two metabolomic networks (default 0.05).

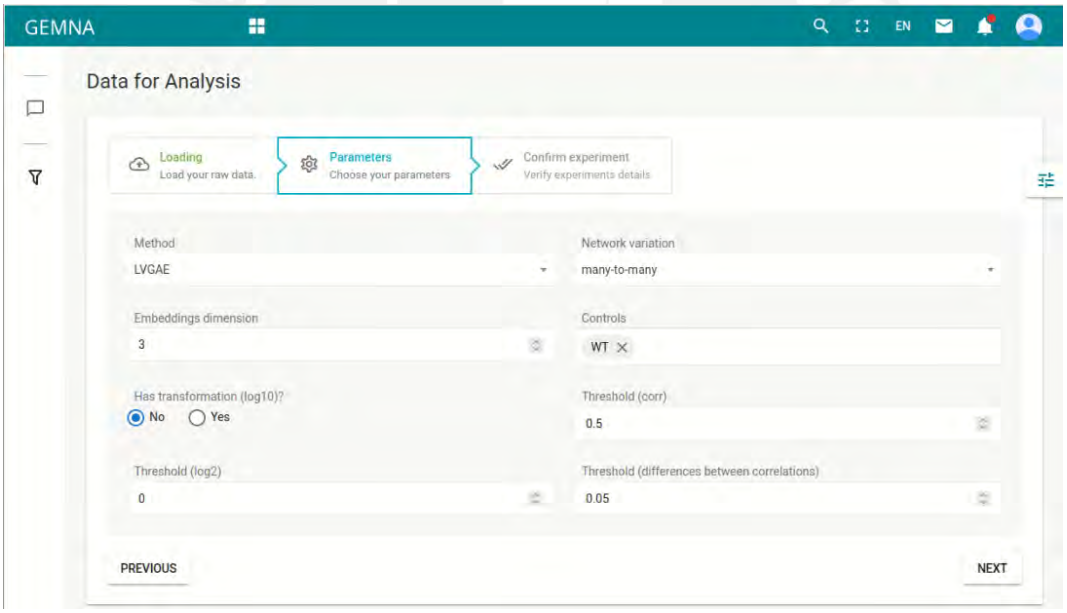


Figure 3: Choose your parameters.

2.2.3 Confirm experiment

In this last step, all selected parameters are displayed (see Figure 4). To start the process, click on the FINISH button. Then, the web application will start processing and analyzing the data. At the end, the user will receive an email

with a code, as shown in Figure 5. Additionally, you will receive .csv files with the correlation changes between the phenotypes. These files are already formatted to upload the results to the `BioCyc` database.

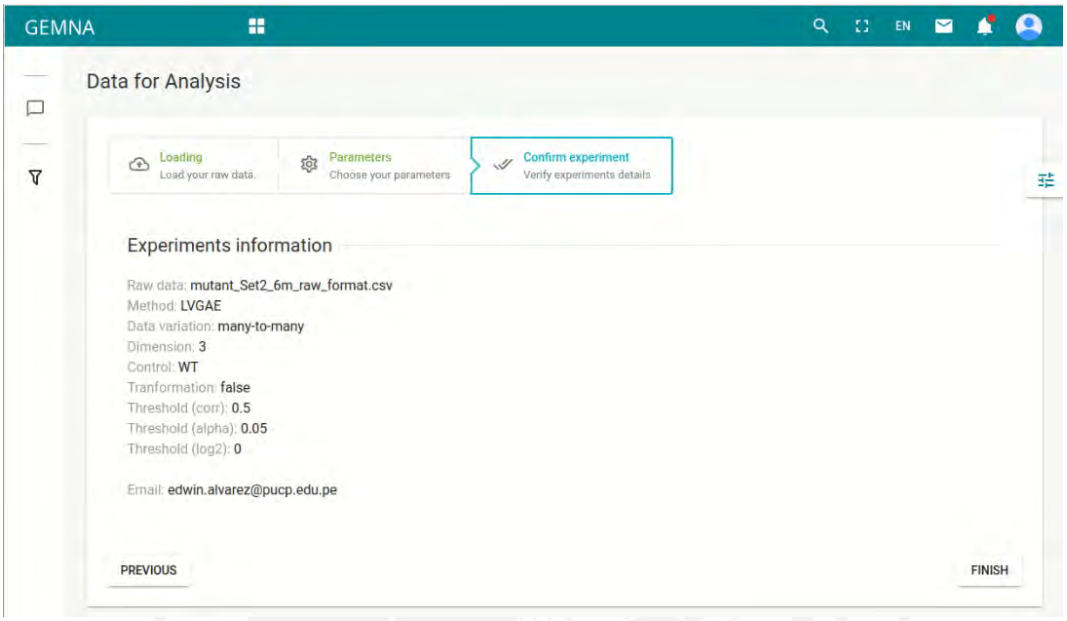


Figure 4: Verify experiment details.

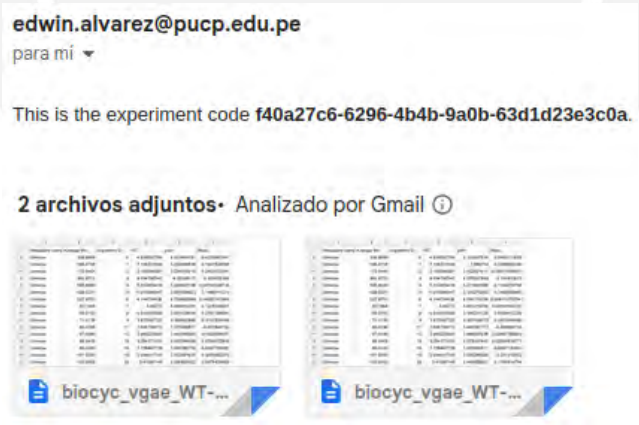


Figure 5: Received e-mail.

3 Visualization

3.1 Load experiment

Type the experiment's code in the form (see Figure 6), and click the LOAD button. After that, two summary tables are displayed. There, you can see the details of the networks and a table with the number of correlation changes, as shown in Figure 7.

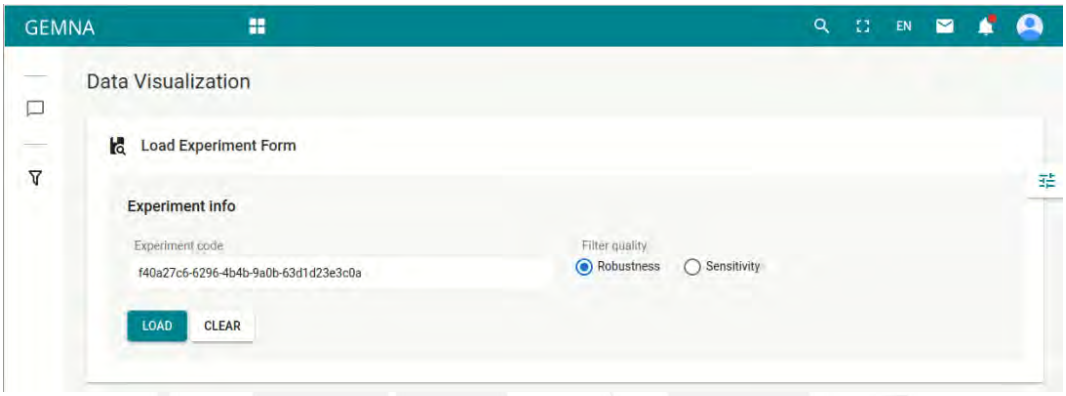


Figure 6: Load experiment form.

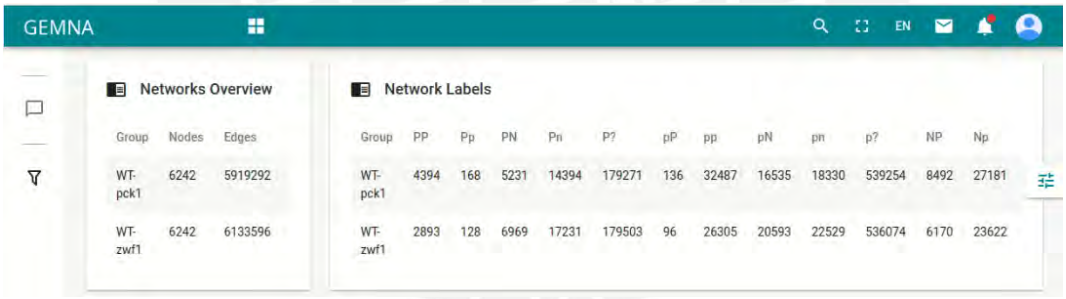


Figure 7: Network overview and network labels.

3.2 Analysis of specific result

After the previous step, a form will be displayed to perform a more specific analysis of the results (see Figure 8). First, you select the groups to compare (Wt-pck1 or WT-zwf1). Second, you select the metabolites of interest, and the search can be done using any of the metadata (ID, Retention time, Average Mz, Metabolite name). Thirdly, the type of visualization selected can be *Correlation nodes*, where the network shows only the correlation between the

selected metabolites. In contrast, the *Correlation + neighbors nodes* option shows the previous network plus all the neighbors of the selected nodes. To see the results, click on the FILTER button.

3.3 Similarity analysis and heatmap

Finally, examine the interaction between the metabolites of interest, Figure 9 (left) shows the change in correlation between the metabolites of interest. On the other hand, Figure 9 (right) shows the heatmap of the correlation changes.

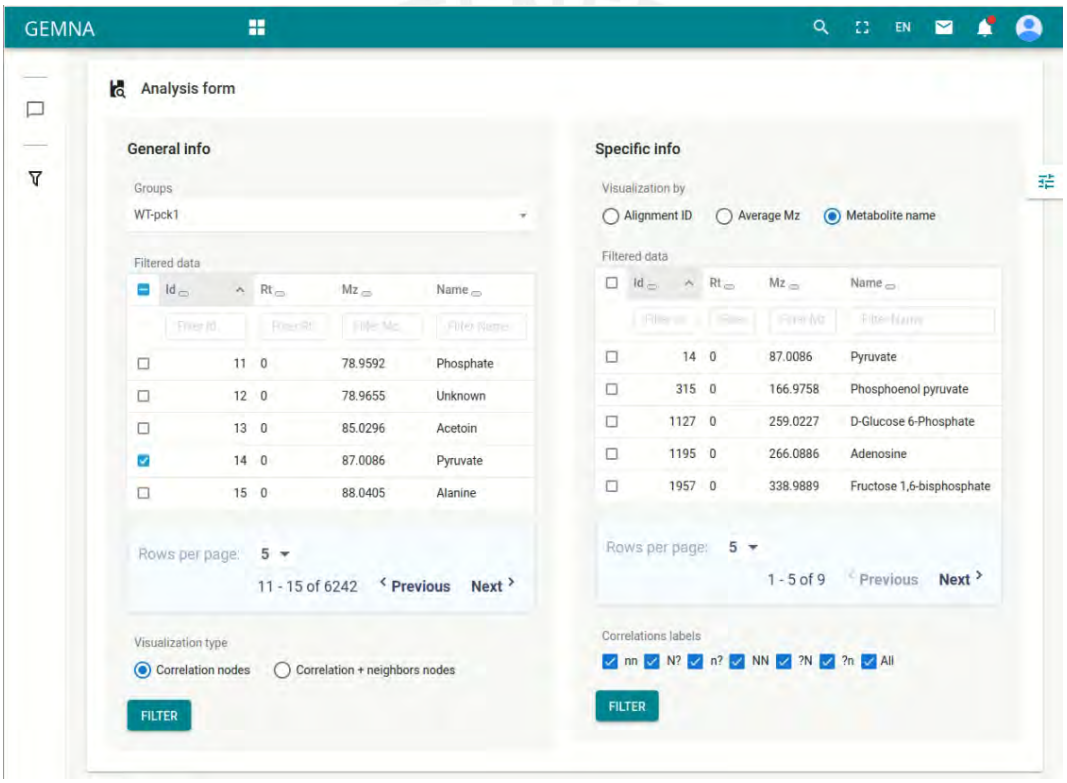


Figure 8: Analysis form.

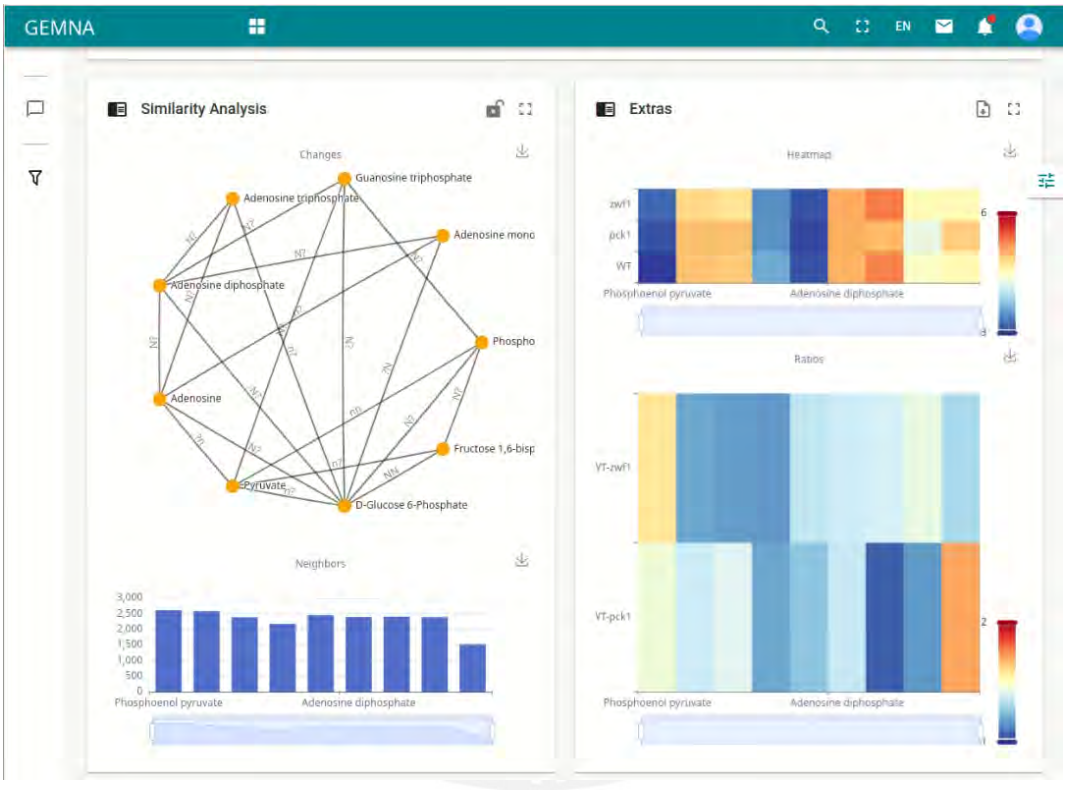


Figure 9: Similarity analysis and heatmap.