2022

SEGCECO: Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication

Sheena Hora
University of Windsor

SEGCECO: Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication

By

Sheena Hora

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2022

SEGCECO: Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication

by

Sheena Hora

APPROVED BY:

N. Zhang
Department of Electrical and Computer Engineering

J. Lu
School of Computer Science

L. Rueda, Advisor
School of Computer Science

April 19, 2022

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

I. Co-Authorship

I hereby certify that this thesis contains the following content that is the outcome of joint research:

Chapter 2 of the thesis was co-authored with Akram Vasighizaker and Luis Rueda. L. Rueda contributed with initial thoughts about this research area and the main ideas, and assisted in elaborating on the new novel approach implemented in this work. All the authors contributed in finalizing the ideas and follow-up discussions. A. Vasighizaker helped in the data collection, preprocessing step. S. Hora implemented the pipeline, experimental design, and the data analysis. S. Hora, A. Vasighizaker, and L. Rueda wrote the contents of the chapter. L. Rueda supervised the project.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis. I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Previous Publication

This thesis consists of one RECOMB 2022 (The 26th Annual International Conference on Research in Computational Molecular Biology) accepted poster and a paper to be submitted to a scientific journal, as follows:

Thesis chapter	Publication title	Publication Status
Abstract	S. Hora, A. Vasighizaker and L. Rueda, "SEGCECO: Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication	Accepted as Poster in RECOMB 2022
Chapter 2 (primary version)	S. Hora, A. Vasighizaker and L. Rueda, SEGCECO: Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication	Submitted to Bioinformatics

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution. I understand that my thesis may be made electronically available to the public.

ABSTRACT

Motivation: Recently, graph-structured data has become increasingly developed in a variety of fields from biological networks to social networks. While link prediction is one of the key problems in graph theory, cell-cell communication regulates individual cell activities and is a crucial part of tissue structure and function. In this regard, recent advances in single-cell RNA sequencing technologies have eased routine analyses of intercellular signaling networks. Previous studies work on various link prediction approaches. These approaches have certain assumptions about when nodes are likely to interact, and thus, showing high performance for some specific networks.Subgraph-based methods have solved this problem and outperformed other approaches by extracting local subgraphs from a given network.

In this work, we present a novel method, called Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication (SEGCECO), which uses an attributed graph convolutional neural network to predict cell-cell communication from single-cell RNA-seq data. SEGCECO captures the latent as well as explicit attributes of undirected, attributed graphs constructed from gene expression profile of individual cells. High-dimensional and sparse single-cell RNA-seq data make the process of converting the data to a graphical format a daunting task. We successfully overcome this limitation by applying SoptSC, a similarity-based optimization method in which the cell-cell similarity matrix is learned from single-cell gene expression data. The cell-cell communication network is then built using this similarity matrix.

Results: To evaluate our proposed method, we performed experiments on six scRNA-seq datasets extracted from the human and mouse pancreas tissue. Our comparative analysis shows that SEGCECO outperforms latent feature-based approaches, as well as the state-of-the-art method for link prediction, WLNM, with 0.99 ROC area under the curve and 99% prediction accuracy.

Keywords: cell-cell communication, link prediction, single-cell RNA seq, latent feature approaches, graph convolutional neural network, subgraph embedding.

DEDICATION

I would like to dedicate this thesis to my family for their constant support and encouragement during my graduate studies. A special thanks to my mother and husband who have always encouraged me to work hard and achieve my goals.

ACKNOWLEDGEMENTS

This work has been made possible by using the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca) and Compute/Calcul Canada. This research work has been partially supported by the Natural Sciences and Engineering Research Council of Canada, NSERC, Vector Institute for Artificial Intelligence, Canada, and the University of Windsor, Office of Research Services and Innovation.

I would like to express my gratitude to my supervisor, Dr. Luis Rueda, for his assistance and encouragement during the thesis. He has been an outstanding instructor, mentor, and thesis advisor. His support, guidance, and extensive knowledge in this field have made this an inspiring experience for me. I am grateful for the opportunity to work under his supervision.

I would also like to thank my external reader Dr. Ning Zhang and internal reader Dr. Jianguo Lu for being a part of my thesis committee and for your insightful comments and suggestions. Your feedback has been really valuable to me.

Finally, I would like to give special thanks to my husband Amir Singh and the entire family for their endless love, support, and understanding. My parents instilled in me the values of working hard, believing in oneself, being determined, and achieving one's goals. Your prayers for me have kept me going this far. Thank you to all of my friends, especially Swaminathan, who encouraged me to pursue this course and has always bolstered me, Akram, who has always been an inspiration to me. Last but not least, I would like to express my appreciation to all my friends and colleagues of the School of Computer Science at the University of Windsor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Basics of Molecular Biology

Molecular biology refers to the study of biology at a molecular level. Genetics and biochemistry are areas where the field intersects with biology and chemistry. The focus of molecular biology is on understanding the relationships between the different systems of a cell, such as the interactions between DNA (deoxyribonucleic acid), RNA (ribonucleic acid), and protein biosynthesis, as well as how these interactions are regulated [21].

1.1.1 Cell

A cell is the basic unit of living organisms capable of sustaining life and reproduction in living creatures. Viruses are not cells because they are incapable of sustaining life and reproducing on their own. A nerve cell or a red blood cell, for example, are two kinds of cells. There are two types of cells: prokaryotic and eukaryotic cells. In a prokaryotic cell, there is no nucleus. Every eukaryotic cell contains a nucleus. Eukaryotes are organisms that are made up of eukaryotic cells. Protista, fungi, animals, and plants are among them. Archaebacteria and eubacteria are two types of prokaryotes. They are organisms with only one cell. Molecular biology focusses on the study of the molecular foundations of the replication, transcription, and translation of genetic material [21].

1.1.2 Deoxyribonucleic acid and Ribonucleic acid

Deoxyribonucleic acid, or DNA, is the molecule that contains the majority of genetic information in cells. A double helix structure, in which two individual DNA strands twist around each other in a right-handed spiral, is the most common form of DNA in a cell. To make the information written in the DNA usable, parts of it are transcribed into another sort of biological information called Ribonucleic acid, or RNA [15]. RNA is a molecule similar to DNA. Unlike DNA, RNA is single-stranded. RNA delivers the DNA's information out of the nucleus which is known as messenger RNA (mRNA) [14].

Nucleotides are the basic building block of nucleic acids like DNA and RNA. Each nucleotide subunit is made up of three components: a phosphate group, a deoxyribose sugar ring and a nucleobase [21]. The four types of nucleobase in DNA are: adenine (A), cytosine (C), guanine (G), and thymine (T). The nucleobases of RNA are also of these four types, with the exception that the T in RNA is substituted by the uracil (U). The structure of RNA and DNA is shown in Fig 1.10.1.

DNA is made up of two strands, each of which runs in the opposite direction. The end of the DNA strand that contains the hydroxyl group is known as 3' end of the molecule. The end that has a phosphate group is known as 5' end. Base A is always paired with base T on the other strand while base G is paired with base C. This is termed as base pairing. The base-pairing rule becomes A-U, T-A, G-C, and C-G when an RNA strand pairs with a DNA strand [14]. An example of a segment of double-strand DNA sequence is shown below:

5'- A C C G A C T T G C G A -3'
3'- T G G C T G A A C G C T -5'

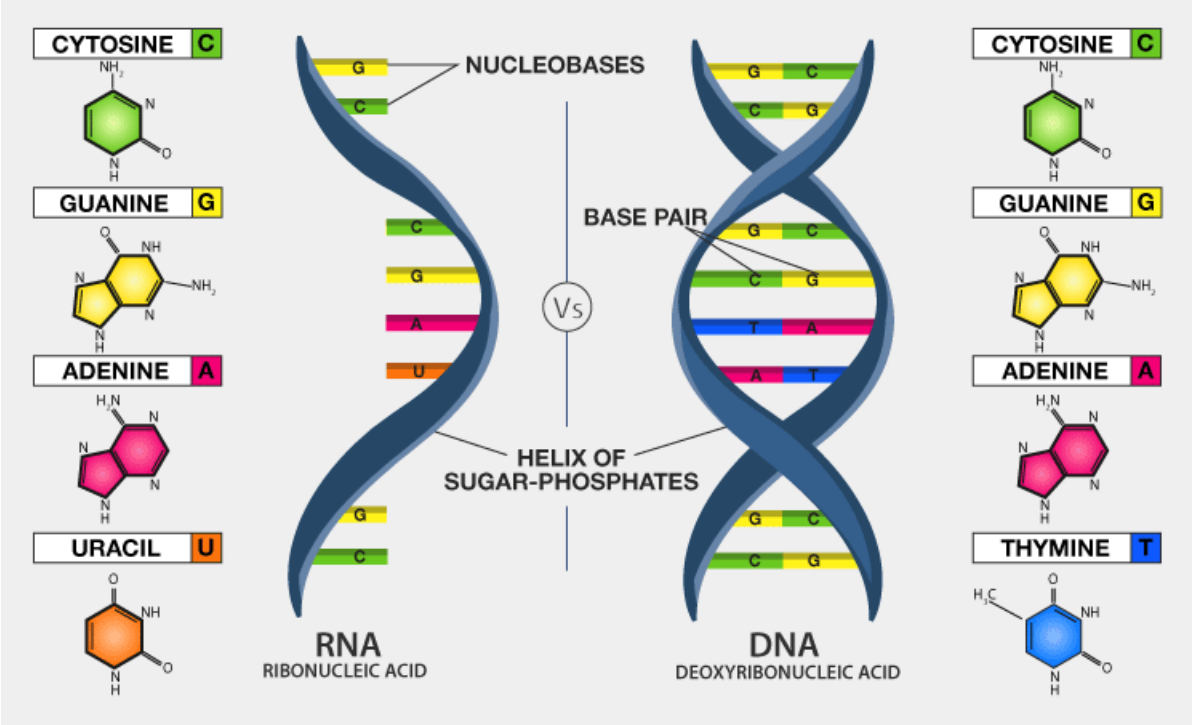

Fig. 1.1.1: Structure of RNA and DNA [7]

1.1.3 Gene

The parts of DNA that are transcribed into RNA are known as coding regions or genes [15]. In a living organism, the gene is the basic unit of heredity. All living organisms depends on genes. Genes include the information for constructing and maintaining cells, as well as passing genetic features on to offspring. A gene is a piece of nucleic acid that holds genetic information and specifies a trait [21].

1.2 Central Dogma of Molecular Biology

The central dogma in genetics outlines the typical mechanism by which information encoded in DNA sequences is first passed on to a kind of RNA known as messenger RNA (mRNA) through transcription process and subsequently to proteins via translation process (Fig 1.2.1). The complementary base pairing rule governs transcription between the DNA base and the transcribed RNA base. That is, an A in DNA is transcribed to a U in RNA, a T to an A, a G to a C, and vice versa [14].

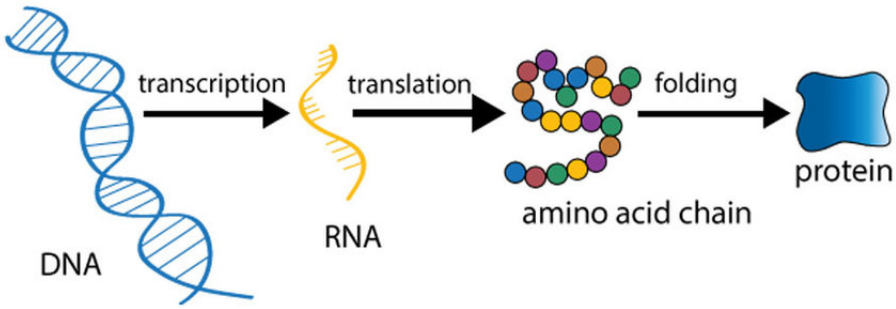

Fig. 1.2.1: Central dogma of molecular biology [1]

Each protein is made up of a linear sequence of smaller molecules known as amino acids. The constituent amino acids are connected by a "backbone" made up of a regularly repeated sequence of bonds. There are 20 different standard amino acids that utilized in production of proteins. The number of amino acids that make up a protein is commonly used to determine its size. Proteins can be anywhere between 20 and 5000 amino acids long, with the average protein being around 350 amino acids long. Proteins are synthesized in a two-step process i.e. transcription and translation. The combination of these two steps is referred to as gene expression [26]. The transcription and translation processes plays a critical role in determining what proteins are present in a cell and in what amounts. Proteins are encoded by genes, and cell function is dictated by proteins. As a result, the thousands of genes expressed in a cell determine what that cell can do. Furthermore, each step in the information flow from DNA to RNA to protein presents a possible control point for the cell to self-regulate its functions by altering the number and type of proteins it produces. The amounts and types of mRNA molecules in a cell represent the function of that cell [9].

1.3 Next-generation sequencing

With ultra-high-throughput output and a huge cost reduction, next generation sequencing (NGS) has superseded the conventional Sanger sequencing approach to become the preferred choice for large-scale, genome-wide sequencing studies. In the

biological sciences, NGS technologies have had a significant impact on structural and functional genomics research [19]. NGS applications are mostly used on DNA and RNA molecules. We can detect DNA and RNA sequences as well as define DNA–protein interactions and epigenetic DNA modifications by sequencing these (DNA and RNA) molecules. Thus, the NGS output data provide a wealth of information about the structural and functional properties of cells and tissues. Commonly used NGS applications are Expression analysis, DNA-protein interactions, DNA methylation, whole genome sequencing, whole-exome sequencing, target sequencing, and de novo sequencing [15].

1.4 RNA-Sequencing

In recent years, RNA-seq has spurred a lot of medical discovery and innovation. RNA-seq (RNA-sequencing) is a technology that uses NGS to analyze the quantity and sequences of RNA in a sample. It examines the transcriptome to determine which of the genes encoded in our DNA are active and to what extent. RNA-seq can be used to examine and discover the transcriptome, the total cellular composition of RNAs and cellular responses. Some of the most popular RNA-seq strategies include transcriptional profiling, single nucleotide polymorphism (SNP) identification,3 RNA editing, and differential gene expression analysis. It provides information to researchers about the function of genes. Sanger sequencing technology, although being innovative at the time had low throughput and was expensive, was earlier used by RNA-seq technologies. With the advent and widespread of NGS technology, we have lately been able to take full advantage of RNA-seq's potential [22].

There are various steps in an RNA-seq workflow. The first stage in the process is isolate the RNA and perform the reverse transcription which is to convert the population of RNA to be sequenced into complementary DNA (cDNA) fragments (a cDNA library). After that, the cDNA is fragmented, and adapters are attached to each fragment's end. These adapters have functional features that allow for sequenc-

ing. They allow the sequencing machines to recognize the fragments, and allows us to sequence different samples at the same time, since different samples can use different adapters. Next, the library is PCR amplified. Only the fragments with adapters are amplified; they are encriched. After amplification, size selection, clean-up and quality check is performed. This step is known as cDNA library preparation. The next stage in the workflow is cDNA sequencing. The cDNA library is then evaluated by NGS, yielding short sequences that correspond to all or part of the fragment from which it was produced. Sequencing can be done in one of two ways: single-end or paired-end. Single-end sequencing is cheaper and faster as compared to paired-end sequencing. These reads, by the end of workflow, can then be aligned to a reference genome or assembled de novo to produce genome-wide expression profile. In addition to the currently known transcripts, de novo assembly will allow for the discovery of new ones. After alignment stage, the last step is RNA-seq data analysis [22].

RNA-seq is regarded as superior compared to microarray technology. This is because of several reasons. Microarray technology depends on already known genes, whereas RNA-seq can detect transcripts from organisms with previously undetected genomic sequences. This makes it far more effective in detecting novel transcripts, SNPs (Single-nucleotide polymorphism), and other alterations. Also, RNA-seq data is quantifiable, but microarray data is only ever displayed as values relative to other signals discovered on the array. Microarrays have trouble detecting very high or very low transcription levels, which RNA-seq avoids [22].

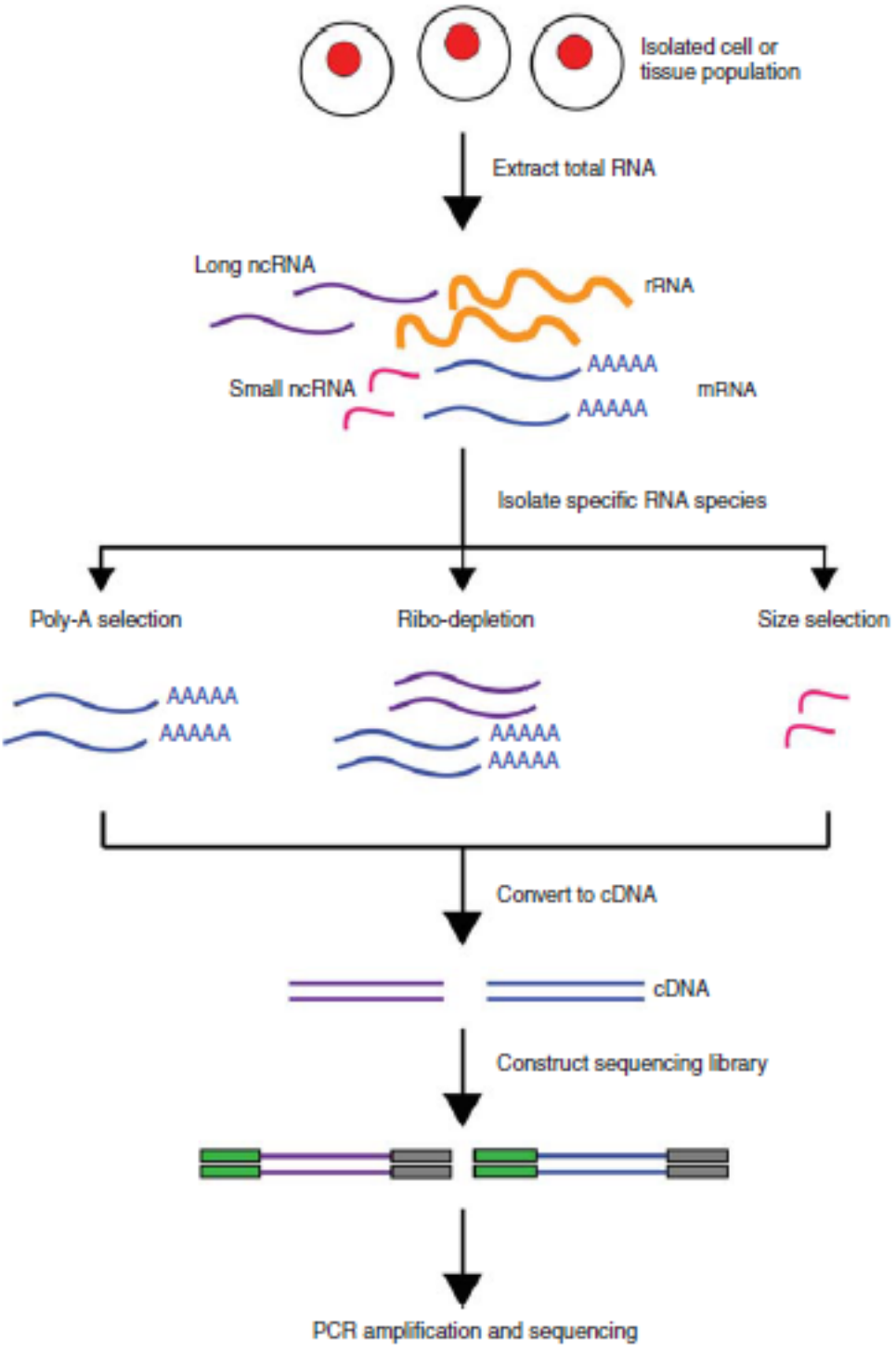

Fig. 1.4.1: Overview of RNA-seq workflow [18].

1.5 Single-cell RNA sequencing

In the last decade, bulk RNA-seq methods have been widely used to study gene expression patterns at the population level. The introduction of single-cell RNA sequencing (scRNA-seq) has opened up opportunities for studying gene expression profiles at the single-cell level. Since bulk RNA-seq generally represents the averaged gene expression across thousands of cells, scRNA-seq has become a popular choice for addressing crucial biological questions of cell heterogeneity and the development of early embryos (which only have a few cells) [6]. In comparison to existing profiling methods that examine bulk populations, these single-cell assessments will allow researchers to uncover new and perhaps unexpected biological discoveries. For example, scRNA-seq can identify rare cell populations, reveal regulatory relationships between genes, and track the trajectories of distinct cell lineages in development [13]. There has been a growing interest in doing scRNA-seq studies since the first one was published in 2009. One of the most compelling reasons is that scRNA-seq can describe RNA molecules in individual cells at a genomic scale and with high resolution. scRNA-seq can provide crucial information regarding fundamental characteristics of gene expression in addition to resolving cellular heterogeneity [12]. In recent years, scRNA-seq has been used to study a variety of species, including human tissues (both healthy and cancer), and these studies have revealed significant cell-to-cell gene expression heterogeneity [6].

A scRNA-seq workflow is depicted in Fig. 1.5.1 which includes nine basic steps. The first, and most important, step in conducting scRNA-seq is the successful isolation of live, single cells from the tissue of interest. In the next step, isolated individual cells are lysed in order to capture as many RNA molecules as possible. Poly[T]-primers are often used to selectively analyze polyadenylated mRNA molecules while avoiding capturing ribosomal RNAs. Non-polyadenylated mRNA analysis is often more complex and necessitates the use of specialist procedures. Next, a reverse transcriptase converts poly[T]-primed mRNA to complementary DNA (cDNA). Other nucleotide

sequences, such as adaptor sequences for detection on NGS platforms, unique molecular identifiers (UMIs) to mark unambiguously a single mRNA molecule, and sequences to preserve information on cellular origin, will be added to the reverse-transcription primers, depending on the scRNA-seq protocol. The minute amounts of cDNA are subsequently amplified by PCR or, in certain cases, by in vitro transcription followed by a second round of reverse transcription. Then, using library preparation methods, sequencing platforms, and genomic-alignment tools, the amplified and tagged cDNA from each cell is pooled and sequenced by NGS. Next, bioinformatic tools are used to assess quality and variability. Lastly, the analysis and interpretation of the data is done using bioinformatics and/or computational methods [12].

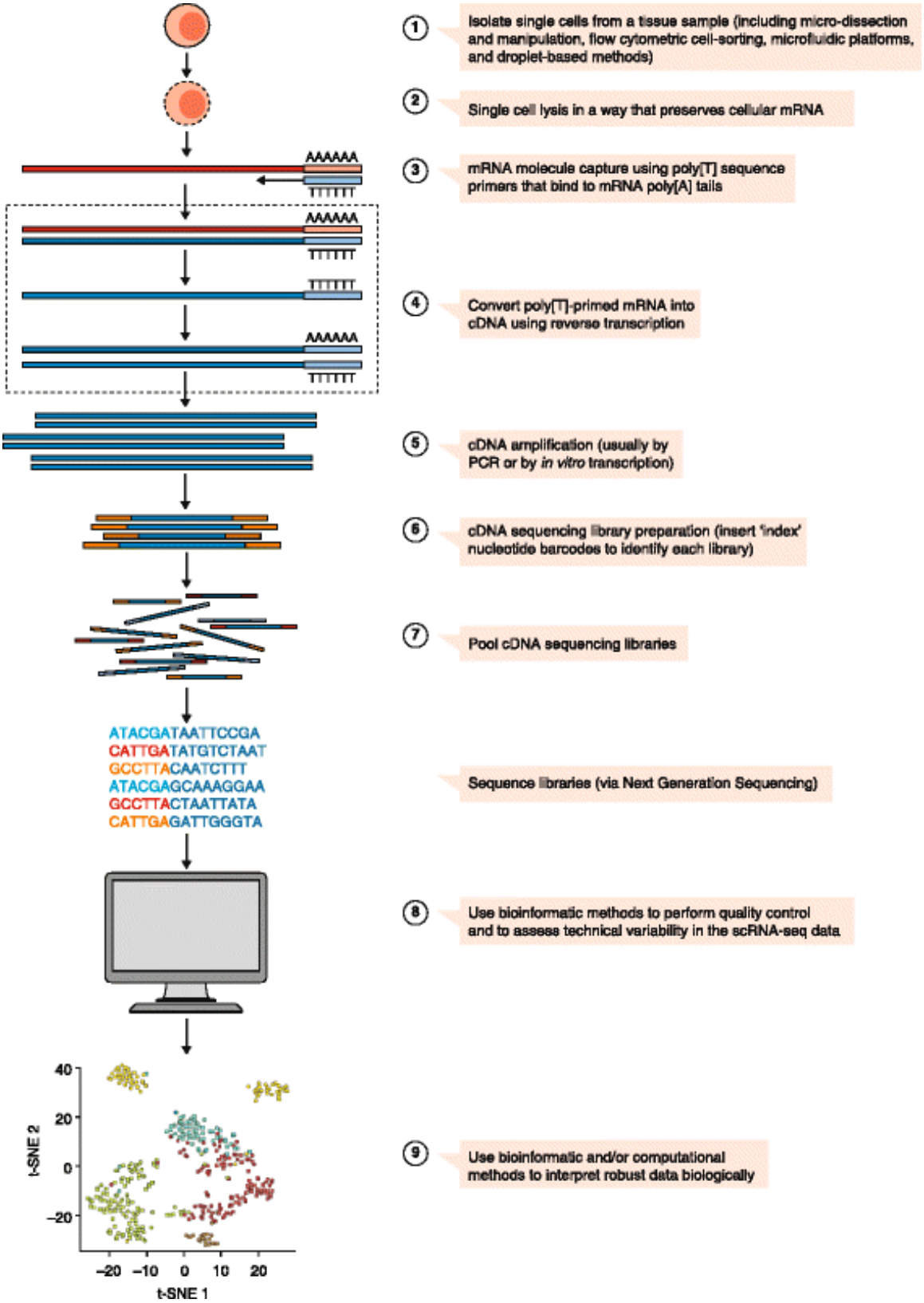

Fig. 1.5.1: Overview of scRNA-seq workflow [12].

1.6 Cell-cell communication

Cell-cell communication (CCC) or cell-cell interaction refers to the direct interactions between cell surfaces that play a crucial role in the development and function of multicellular organisms. It is an important aspect of tissue form and function which regulates individual cell functions and intercellular connections [2]. The coordination of cellular activity, which is dependent on cell–cell interactions (CCIs) across an organism's diverse cell types and tissues, is essential for multicellular life. Thus, CCC plays an important role as the ability to send and receive signals is essential for the survival of the cell and disease occurs when cells do not connect properly or decode molecular messages incorrectly [3]. Cells use ligands, which are molecules generated by sending cells to target cells that can detect them if they have the right receptor. The binding of ligands to receptors causes a response, such as changes in gene expression or the activation of cell division [4]. Cells can communicate with one another in a variety of ways, including Juxtacrine, autocrine, endocrine, and paracrine signaling [3]. The scheme of signalling is depicted in Fig. 1.6.1.

- **Juxtacrine** Juxtacrine cell–cell communication relies on gap junctions to transmit signalling molecules directly between cells, without secretion into the extracellular space. It is contact-dependent communication between cells [3].

- **Autocrine signalling** Autocrine signalling is a type of intracellular communication in which cells release ligands that are employed to trigger cellular responses via corresponding receptors on the same cell [3].

- **Endocrine signalling** Endocrine cell–cell communication refers to intercellular communication that includes the secretion of signaling molecules that travel long distances through extracellular fluids like blood plasma [3].

- **Paracrine signalling** Paracrine cell–cell communication does not require cell–cell contact; instead, signaling molecules spread from one cell to another after secretion [3]. In this type of signaling, cells are typically close to one another. It enables communication over short distances [4].

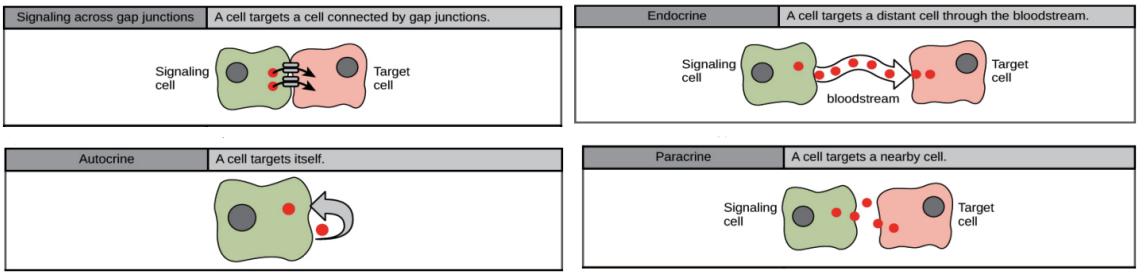

Fig. 1.6.1: Schemes of signalling [4]

Due to the availability of single-cell transcriptomics, the study of single-cell transcriptomics has shifted from focusing solely on what cells are there to focusing more on the interactions between cells [2]. Recent advances in RNA sequencing technologies have made routine analysis of intercellular signaling from bulk and single-cell gene expression data sets possible [3]. In this study, we focus on predicting CCC in scRNA-seq data.

1.7 Graphs (Networks)

Graphs or networks are a universal language for describing a set of complex systems [31]. Graphs or networks give us a way to explicitly and mathematically represent this complex information, as well as the complex interactions that exist in today's data. It gives a visual representation of data which helps us gain actionable insights and make better data driven decisions. For Example, the working of a social system can be represented by considering the interactions between the pair of people. Graphs can be used to simulate a wide range of relationships and processes in physical, biological, social, and information systems, and thus has a variety of applications including Finding communities in network i.e. friends/connection recommendation on social media, to discover uncommon patterns that aid in the prevention of fraudulent transactions, GPS/Google maps to find the fastest route to home etc. There are complex systems all around us: society is made up of around eight billion people, communication systems bind electronic devices, information and knowledge are organized and connected, thousands of genes and proteins interact to control life,

and human thoughts are hidden in billions of neurons in our brain [10]. All of these complex systems have a graph structure. We deal with undirected, attributed graphs.

*A **graph** is defined as $G = (V, E)$, where V is the finite set of vertices (or nodes) and E is the finite set of edges (or links). Also, $|V| = n$, where n is the number of vertices. $|E| = m$, where m is the number of edges.*

Figure 1.7.1 illustrates a graph G with five vertices and five edges connecting them, with $V = \{v_1, v_2, v_3, v_4, v_5\}$ and $E = \{e_{12}, e_{23}, e_{34}, e_{14}, e_{25}\}$.

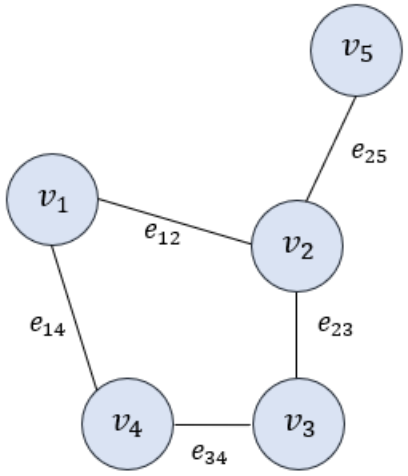

Fig. 1.7.1: Graph with five vertices and five edges connecting the vertices.

1.7.1 Adjacency Matrix

In a graph $G = (V, E)$, let $(v_i, v_j) \in V$ denote two vertices and $e_{ij} = (v_i, v_j) \in E$ denote an edge between vertices v_i and v_j. The adjacency matrix $\mathbf{A} = [a_{ij}] \in R^{n \times n}$ [20] is a $(n \times n)$ matrix, where n is the number of vertices. Each element a_{ij} represents the element at i^{th} row and j^{th} column of the adjacency matrix. We define $a_{ij} = 1$ if $e_{ij} \in E$ and $a_{ij} = 0$ if $e_{ij} \notin E$. In other words, the elements in the adjacency matrix are 1 in position where the two vertices are connected by an edge and 0 otherwise. Figure 1.7.2 shows a graph with its adjacency matrix.

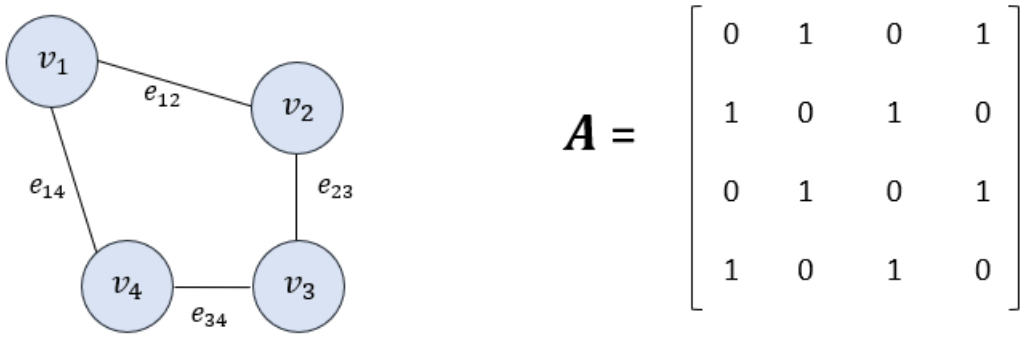

Fig. 1.7.2: Graph with four vertices and four edges and its adjacency matrix.

1.7.2 Directed Graphs

Directed graph is defined as a graph with all the edges of the graph having direction/orientation. The edges of the graph reflect a particular direction from one vertex to the next. In the graph (Figure 1.7.3), vertex v_1 connects to vertex v_2 where node v_1 is the origin and node v_2 is the destination. The direction is from v_1 to v_2.

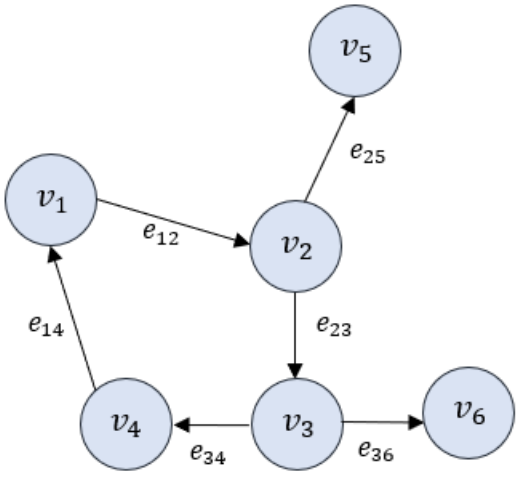

Fig. 1.7.3: Example of Directed Graph with six vertices and six edges connecting the vertices.

1.7.3 Undirected Graphs

An undirected graph has an unordered pair of vertices. In other words, the edges are not represented in any particular direction. The vertices connect by undirected arcs, which are edges without arrows. Figure 1.7.1 shows an example of undirected graph with five vertices and five edges connecting the vertices. An edge between vertex v_1 and v_2 would be identical to the edge from v_2 to v_1.

1.7.4 Attributed Graph

The graph that has attributes associated with vertices is known as attributed graph. Attributes which describes the information/properties of the vertices are represented in the form of matrix X and each row in X defines the attributes associated with that particular vertex v which is represented as x_v. Figure 1.7.4 shows an example of attributed graph with five vertices and five edges. Each vertex has attributes associated to it. For example, the vertex v_2 of graph has attributes $[x_{21}, x_{22}, x_{23}, ..., x_{2d}]$.

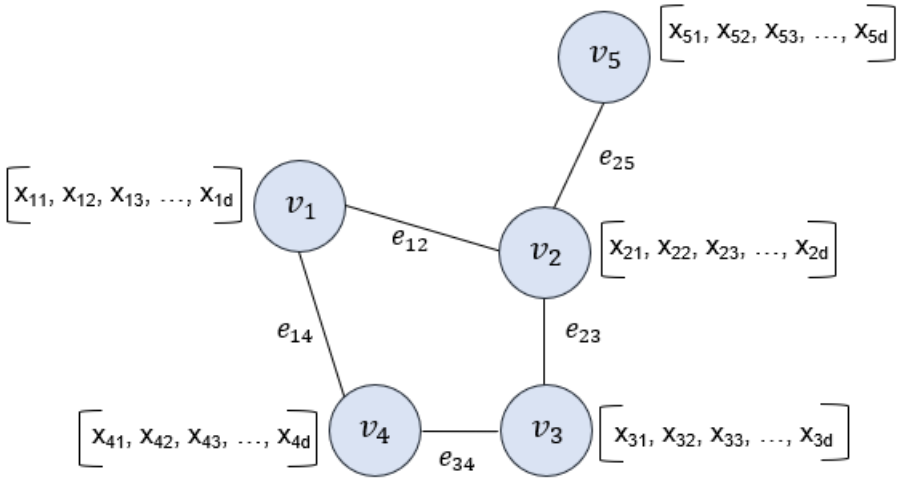

Fig. 1.7.4: Example of an attributed graph with five vertices and five edges. Each vertex has d dimensional attribute vector associated with it.

An attributed graph is defined as $G = (V, E, X)$ *where* $X \in R^{n \times d}$ *is a matrix of attributes.* $x_v \in R^d$ *represents the attribute vector of vertex* v*, where* d *is the number of attributes.*

Below is an example of an attribute matrix X. Each row in the X represents the corresponding attribute vector of a vertex of graph. For example, the second row of the attribute matrix $[x_{21}, x_{22}, x_{23}, ..., x_{2d}]$ denotes the the attribute vector of the second vertex v_2 in the attributed graph.

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1d} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2d} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3d} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nd} \end{bmatrix}$$

1.8 Types of Graph Data

Graphs are everywhere. So everything from state machines to molecular networks to social networks all can be represented with graphs. Social networks, Biomedical networks, economic networks, information networks - webs & citations, and logistic networks are some of the types of graph data. In this section, several examples of graph data have been introduced.

1.8.1 Social Network

A social graph is a graph that depicts the social relationships that exist between individuals. In the social network, nodes represent people or other entities embedded in a social context, and edges represent connections, collaboration, or influence between entities. Facebook ([27]) is one example of a social network. Figure 1.8.1 depicts an example of a social network in which users are the vertices and the link between them is the network's edge. Each user has their collection of properties, such as the photos they have uploaded, the details in their user profile, working together on a project, or studying in the same school. We may suggest new friends to a user using machine

learning algorithms based on their connections and properties such as age, interests, region, and school or work [25].

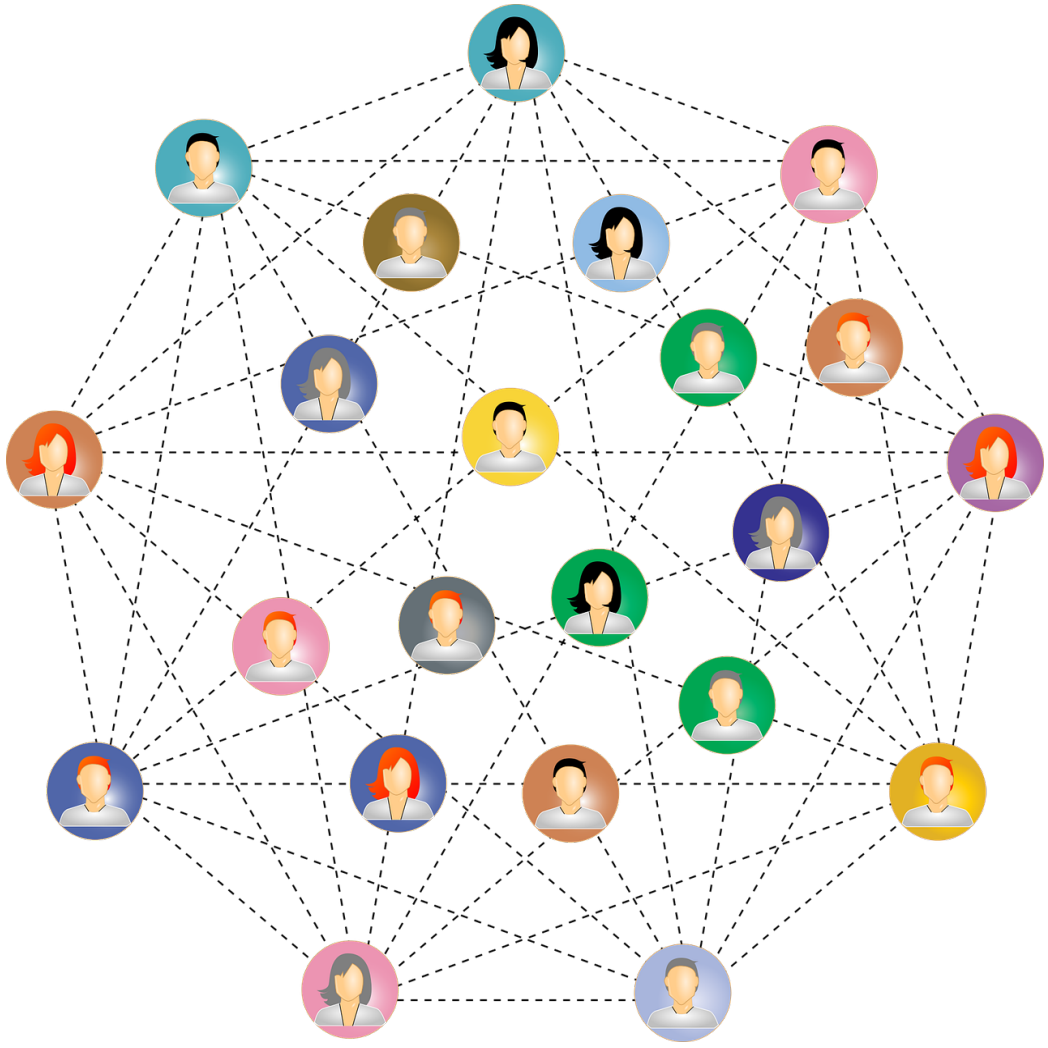

Fig. 1.8.1: Example of a social network with nodes representing people and edges representing their connections or interactions.

1.8.2 Citation Network

A citation graph (or citation network) is a graph that defines the citations inside a series of documents. The vertices of the citation network are authors and research articles, while the edges are citations, authorship, and co-authorship of the research paper. Each research paper has its own set of attributes, including the text, authors, title, publication date, and keywords. We may categorize research papers into var-

ious groups using machine learning algorithms based on the content and attributes of the research papers. CiteSeerX network ([17]) is one example of a citation network. A citation network is depicted in Figure 1.8.2. The researchers and research papers are the network's vertices, while the relations between the researchers and their publications are the network's edges [25].

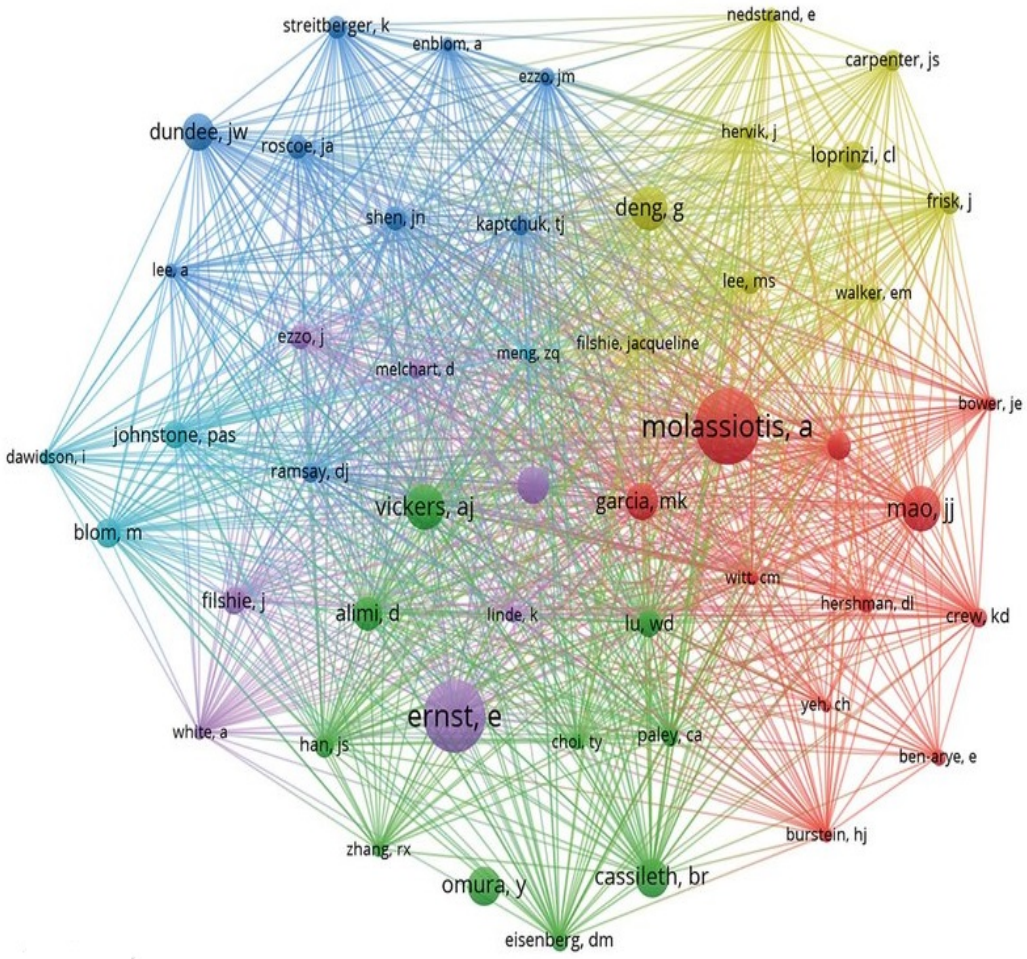

Fig. 1.8.2: Example of a co-citation network of 50 authors that were co-cited in more than 50 publications [11].

1.8.3 Chemical Network

Molecules and atoms are the entities in a chemical network. The chemical network is made up of atoms and molecules interacting with one another. Atoms and molecules,

as well as their properties such as chemical formulas, serve as the network's vertices, while connections and bonds between them serve as the network's edges. Figure 1.8.3 depicts a chemical network with atoms and molecules as vertices and chemical bonds between them as edges [25].

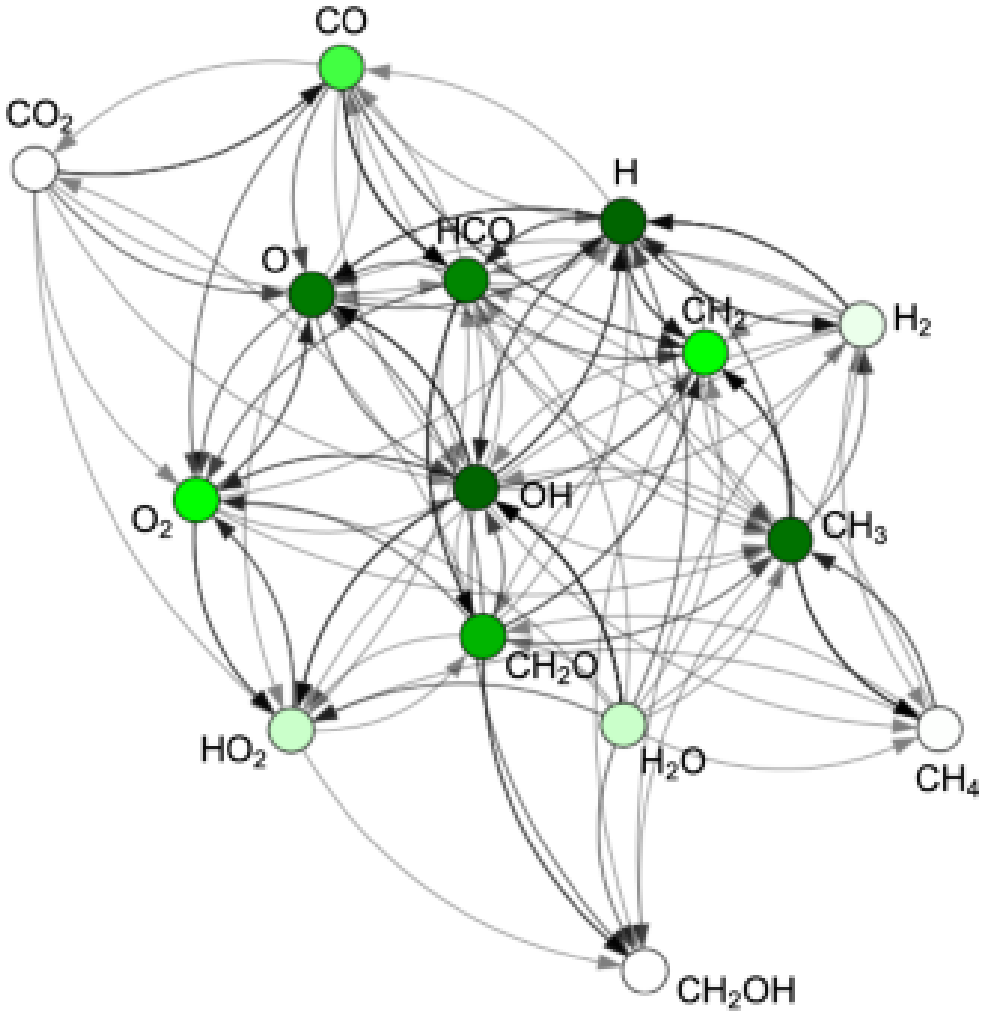

Fig. 1.8.3: Example of a chemical network with molecules as nodes and chemical reactions as edges [24].

1.8.4 Research Network

The researchers are the entities in a research network, and researchers interact with other researchers who are colleagues, research students, partners, and followers to create their network. The researchers serve as the network's vertices, with the form

of research interaction between two researchers serving as the network's edge. ResearchGate [29] is one example of a Research Network. Figure 1.8.4 depicts a research network with the professor, research students, collaborators as vertices, and connections among them as edges [25].

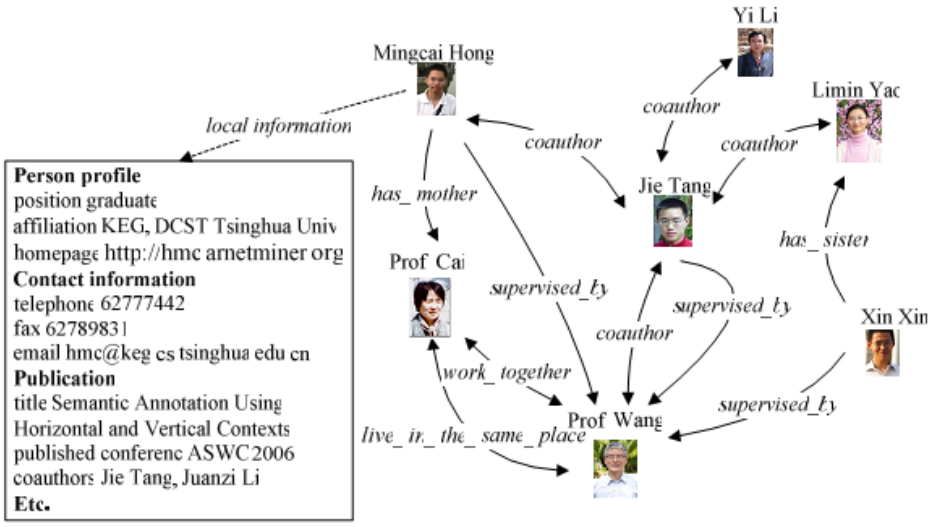

Fig. 1.8.4: Example of an academic research network with researchers as nodes and relationships among them as edges [30].

1.9 Machine Learning Tasks in Network

Machine learning is a technique for improving system performance through the use of computing algorithms that learn from experiences. Experience exists in the form of data in computer systems, and machine learning's major purpose is to construct learning algorithms that generate models from data. We feed the learning algorithms with experience data and create a model that can make predictions based on new observations [32]. Learning or training is the process of using machine learning algorithms to construct models from the data. The data used to train the model is referred to as *training data*, while the data used to test the model, i.e. make predictions, is referred to as *testing data*. The outcome information of a sample is known as *label*. Machine learning can be used for prediction, classification, clustering, recommendation, image recognition, speech recognition, and a variety of other tasks.

We divide machine learning tasks into two categories based on whether the dataset is labeled or not: *supervised learning* (for example - classification and regression) and *unsupervised learning* (for example - clustering) [32].

Supervised Learning is when you have input data x and output data y and the goal is to learn a mapping function from input data to the output data. To train the model, supervised learning requires supervision, similar to how a student learns in the presence of a teacher. Unsupervised learning is when you only have input data x but no corresponding output data. Unsupervised learning focuses on extracting structure and patterns from unstructured data. Unsupervised learning does not require supervision. Instead, it searches the data for patterns on its own. Labeled data is used to train supervised learning algorithms while unlabeled data is used to train unsupervised learning algorithms. Semi-supervised learning is a hybrid approach (i.e. combination of supervised and unsupervised approach) when you only have some labeled data.

In this work, we uses a supervised learning machine learning framework where labeled data is available. Machine learning can be used to graph datasets in a variety of ways. Classical machine learning tasks in networks are node classification, community detection, graph classification and link prediction. These are explained below. In this thesis, we focus on prediction of cell-cell interactions (i.e. link prediction task).

1.9.1 Node classification

Node classification, also known as vertex classification, is the problem of identifying the unknown labels of the nodes in a network using machine learning approaches given the labels are available for small subset of nodes [16] and [5]. Figure 1.9.1 depicts an input graph with the orange and green class labels. The goal of machine learning is to predict the class labels of the grey nodes marked with question mark.

Fig. 1.9.1: Node Classification Example with an input graph with two known orange and green class labels (Left) and the goal is to predict the labels of grey nodes as either orange or green (Right).

1.9.2 Community Detection

Detecting communities is important in fields like sociology, biology, and computer science [8]. Given the organization of vertices in clusters (i.e. communities), with multiple edges joining vertices in the same cluster and relatively few links connecting vertices from different clusters, the community detection problem is to predict the community of vertices whose community is unknown [8]. Figure 1.9.2 depicts a graph with four communities (green, red, yellow and grey), enclosed by the dashed circle.

1.9.3 Graph Classification

Given a dataset containing graphs in the form of (G, y) where G is a graph and y is its class, the graph classification problem is to predict the label of the graph for which labels are unknown [31]. Figure 1.9.3 depicts an input dataset of graphs with known graph labels. The goal of machine learning is to learn a function f that can predict the label of unknown graph.

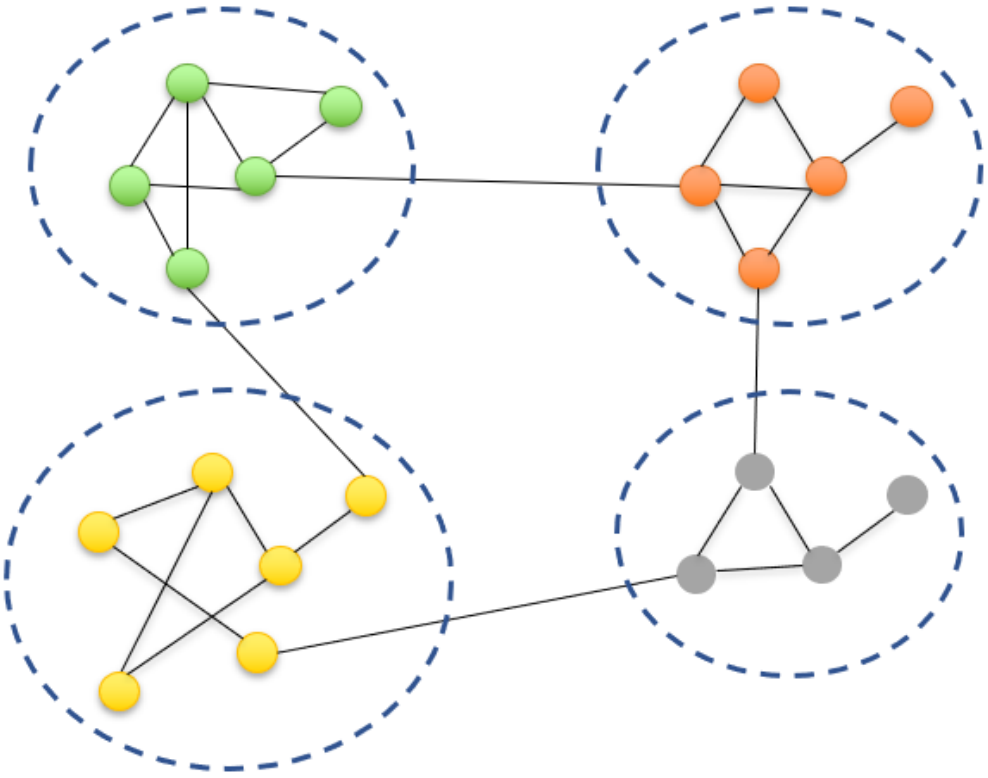

Fig. 1.9.2: Community Detection Example with a graph with four communities (green, red, yellow and grey) enclosed by the dashed circle.

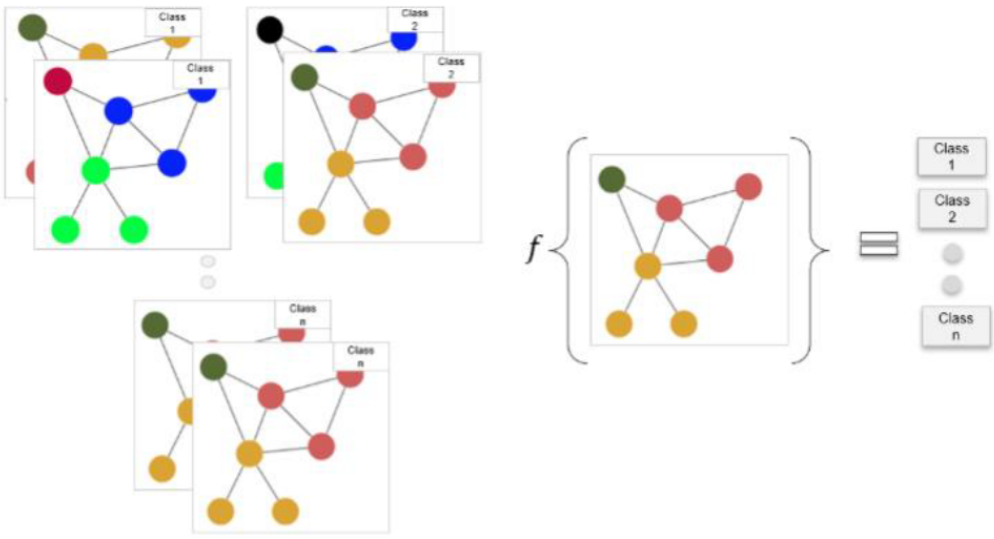

Fig. 1.9.3: Graph Classification Example with an input dataset of graphs with known graph labels and the goal is to learn a function f that can predict the label of unknown graph [25]

.

1.9.4 Link Prediction

Link prediction, also known as edge prediction, is the problem of predicting the missing links between vertices in a network using machine learning approaches given the edges between some vertices [28]. Figure 1.9.4 depicts an input graph with certain known and unknown edges. In Figure 1.9.4, unknown edges are denoted by question marks. The goal of machine learning is to predict whether or not there is an edge between the vertices.

Fig. 1.9.4: Link Prediction Example with an input graph with some known and unknown edges; unknown edges are denoted by question marks (Left) and the goal is to predict the existence of edge between two vertices (Right).

1.10 Performance Metrics

To evaluate our method, we have used the most widely used evaluation metrics, including accuracy, precision, recall, and F1-score. Equation 1 refers to accuracy, which accounts for the percentage of correctly classified test observations. Precision is another measure that calculates the number of positive observations that are correctly predicted as positive from the total number of predicted positive observations. Furthermore, recall measures the number of positive observations that are correctly predicted as positive from a total number of original positive observations. Also, the F1-score signifies average precision and recall. Precision, recall and F1-score are calculated using the following equations:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - score = 2.\frac{(Precision.Recall)}{(Precision + Recall)} \tag{4}$$

where TP, TN, FP, and FN stands for True Positive (model correctly predicts the positive interactions), True Negative (model correctly predicts the negative interactions), False Positive (model incorrectly predicts the positive interactions), and False Negative (model incorrectly predicts the negative interactions) respectively. Here, positive means interacting cells and negative means non-interacting cells.

Apart from these performance measures, the model's performance is evaluated using AUROC (Area Under the Receiver Operating Characteristics). The AUC (Area Under The Curve) ROC (Receiver Operating Characteristics) curve is used to visualize the results. The ROC curve shows the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR). TPR or Recall is the proportion of observations that were correctly predicted to be positive out of all positive observations. FPR is the proportion of observations that are incorrectly predicted to be positive out of all negative observations. The top left corner of the plot is the ideal point for ROC, with 0% FPR and 100% TPR. AUC indicates how well the model can differentiate between interacting and non-interacting cells. Higher the AUC, the better the model is at predicting 0 as $0s$ and $1s$ as $1s$ where 0 means no interactions between cells and 1 means there is an interaction between cells.

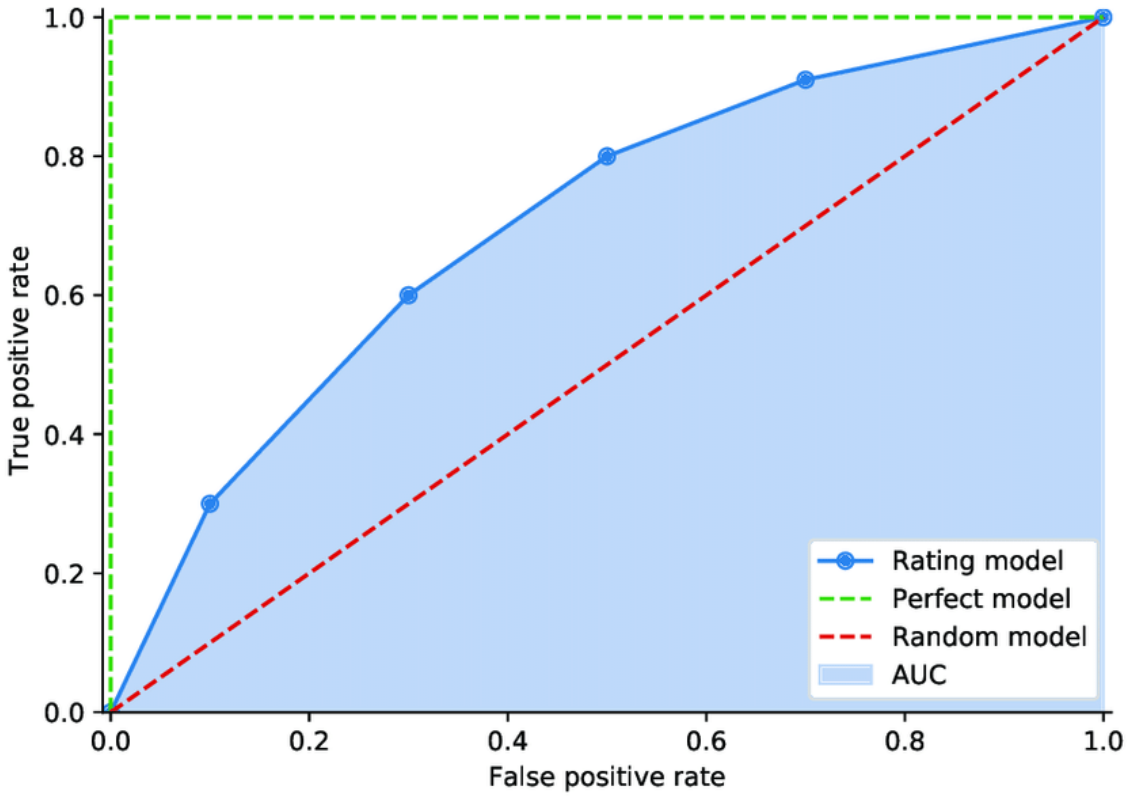

Fig. 1.10.1: Receiver Operating Characteristic (ROC) curve example [23]

1.11 Motivation

Data analysis in the form of graphs is gaining a lot of attention. It has become one of the popular research topics and has gained popularity in a range of domains, ranging from biomedical networks networks to social networks. Link prediction is one of the most important research topics in the field of graphs (or networks). Cell-cell communication play a crucial role in the development and function of multicellular organisms. Recent developments in single-cell RNA sequencing technologies have made routine investigations of intercellular signaling networks much easier. We wanted to do more research in this area of cell-cell communication in single-cell RNA sequencing data. This leads to first converting the scRNA-seq data to graph format and then predicting the cell-cell communication. With a more in-depth analysis of graphs, we discovered that rather than studying the entire structure of a graph, it is important

to create a new graph substructure that captures the high-level attributes by analyzing the attributes of a node and its neighbors. Convolutional Neural Networks (CNNs), for example, are good at capturing and representing data by aggregating the attributes of its neighbors. CNNs perform well on data that is grid-structured, such as images, sequences or text data, and has a well-defined spatial ordering. Graphs, on the other hand, have an arbitrary size and complex topological structure, no fixed ordering, and are also dynamic. Because of the complexity of graph-structured data, replicating the design of a CNN that works effectively with fixed grid-structured data to graph-structured data is a challenge. Our motivation for this thesis is to address the challenging task of handling graph-structured data in the same way that fixed grid-structured data is treated. This problem is addressed using Graph Convolutional Networks (GCN), a class of neural networks explicitly designed for in-depth analysis of graph-structured data [5]. GCN [16] is used to generalize convolution from traditional data (images or grids) to graph data. Given the complicated, high-dimensional single-cell RNA-seq data, the study aims to provide a novel framework for predicting cell-cell communication using GCN.

1.12 Problem Statement

Given complex, high-dimensional scRNA-seq data, we aim to predict cell-cell interactions by creating a pipeline that analyzes single-cell data and converts it to a graph format, performing the prediction using GNNs. We consider the gene expression profile from scRNA-seq data by converting it to an undirected attributed graph, G, in which cells and cell-cell interactions are represented by nodes and edges respectively. More formally, given an undirected attributed graph $G = (V, E, \mathbf{X})$ at a particular time t, where V is a finite set of nodes (cells), E is a finite set of edges (cell-cell interactions), in which $e_{ij} = (v_i, v_j) \in E$ and x_{v_i} is the attribute vector associated with the node $v_i \in$V.*Also*, A=$(\mathrm{a}_{ij})_{N\times N}$ represents the adjacency matrix of graph G, where $a_{ij} = 1$ if $e_{ij} \in E$ and $a_{ij} = 0$ otherwise, and N is the number of nodes. We aim to predict the likelihood of connection between v_i and v_j in the near future. In

other words, link prediction can have a temporal aspect where the goal is to forecast the links at time t' (future) based on the collection of links at present time t.

1.13 Proposed Method

This thesis proposes a novel method, SEGCECO : Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication, for identifying cell-cell communications in single-cell RNA-seq data via a gene expression attributed graph convolutional network. The pipeline consists of three steps: Preprocessing step, Cell-cell communication network (CCN) creation, and Applying the GCN. Before applying GCN, the primary step is to preprocess the data to reduce the effects of noise in the samples. This step includes basic filtering, normalization, log transformation and scaling. Once the data is preprocessed, we create the CCN with nodes representing cells in the CCN and edges representing cell-cell interactions. The last module of the pipeline the proposed GCN for prediction, SEGCECO, which takes processed scRNA-seq data and the CCN to create an attributed graph dataset, and then predict the output. The proposed method is applied to six single-cell RNA-seq datasets extracted from the human pancreas and mouse pancreas tissue. The performance of the method is compared in terms of accuracy and AUC/ROC with other latent feature-based approaches, as well as the state-of-the-art methods for cell-cell interaction prediction. We experimentally demonstrate that our model outperforms other methods.

1.13.1 Contributions

- We propose a new pipeline by integrating methodologies from state-of-the-art studies for cell-cell interaction prediction in scRNA-seq data.
- We introduce a statistically significant pooling layer that employs information gain as an approach for coarsening graph attributes from the scRNA-seq data, while preserving the global structure of the input graph.
- In the Node Information or Attribute matrix, we include explicit features from

the single-cell gene expression matrix.

- We apply the proposed method on different datasets and obtain higher performance compared to the state-of-the-art approaches.
- We have developed an open-source Github project for the proposed pipeline.

References

[1] Ayoob Alfalahi. *Hello ! Is it possible to use genomic DNA(gDNA) to check gene expression instead of using cDNA ?* Apr. 2018.

[2] Axel A Almet et al. "The landscape of cell–cell communication through single-cell transcriptomics". In: *Current opinion in systems biology* 26 (2021), pp. 12–23.

[3] Erick Armingol et al. "Deciphering cell–cell interactions and communication from gene expression". In: *Nature Reviews Genetics* 22.2 (2021), pp. 71–88.

[4] Weronika Bartosik and Julita Kulbacka. "The role of cell-cell communication in physiology and pathology". In: ().

[5] Joan Bruna et al. "Spectral networks and locally connected networks on graphs". In: *arXiv preprint arXiv:1312.6203* (2013).

[6] Geng Chen, Baitang Ning, and Tieliu Shi. "Single-cell RNA-seq technologies and related computational data analysis". In: *Frontiers in genetics* 10 (2019), p. 317.

[7] *Differences Between RNA and DNA*. en. URL: `https://byjus.com/biology/difference-between-dna-and-rna/` (visited on 03/13/2022).

[8] Santo Fortunato. "Community detection in graphs". In: *Physics reports* 486.3-5 (2010), pp. 75–174.

[9] *Gene Expresssion: Learn Science at Scitable - Nature*. en. URL: `https://www.nature.com/scitable/topicpage/gene-expression-14121669/` (visited on 03/13/2022).

[10] Jonathan L Gross and Jay Yellen. *Handbook of graph theory*. CRC press, 2003.

[11] Jing Guo et al. "Research trends of acupuncture therapy on cancer over the past two decades: a bibliometric analysis". In: *Integrative cancer therapies* 19 (2020), p. 1534735420959442.

[12] Ashraful Haque et al. "A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications". In: *Genome medicine* 9.1 (2017), pp. 1–12.

[13] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. "Single-cell RNA sequencing technologies and bioinformatics pipelines". In: *Experimental & molecular medicine* 50.8 (2018), pp. 1–14.

[14] Rui Jiang, Xuegong Zhang, and Michael Q Zhang. *Basics of bioinformatics: Lecture notes of the graduate summer school on bioinformatics of China*. Springer Science & Business Media, 2013.

[15] Melanie Kappelmann-Fenzl. *Next Generation Sequencing and Data Analysis*. Springer, 2021.

[16] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[17] Ajith Kodakateri Pudhiyaveetil et al. "Conceptual recommender system for CiteSeerX". In: *Proceedings of the third ACM conference on Recommender systems*. 2009, pp. 241–244.

[18] Kimberly R Kukurba and Stephen B Montgomery. "RNA sequencing and analysis". In: *Cold Spring Harbor Protocols* 2015.11 (2015), pdb–top084970.

[19] Jerzy Kulski. *Next generation sequencing: advances, applications and challenges*. BoD–Books on Demand, 2016.

[20] Qimai Li, Zhichao Han, and Xiao-Ming Wu. "Deeper insights into graph convolutional networks for semi-supervised learning". In: *Thirty-Second AAAI conference on artificial intelligence*. 2018.

[21] Yinghui Li and Dingsheng Zhao. "Basics of Molecular Biology". In: *Molecular Imaging*. Springer, 2013, pp. 541–601.

[22] Ruairi J Mackenzie. "RNA-seq: Basics, Applications and Protocol". In: *Retrieved from Technology Networks: https://www. technologynetworks. com/genomics/articles/rnaseqbasics-applications-and-protocol-299461* (2018).

[23] Andro Merćep et al. "Deep Neural Networks for Behavioral Credit Rating". In: *Entropy* 23.1 (2020), p. 27.

[24] Sina Stocker et al. "Machine learning in chemical reaction space". In: *Nature communications* 11.1 (2020), pp. 1–11.

[25] Susha Pozhampallan Suresh. "Attributed Graph Classification via Deep Graph Convolutional Neural Networks". PhD thesis. University of Windsor (Canada), 2019.

[26] Martin Tompa. "Basics of Molecular Biology". In: (2003).

[27] Johan Ugander et al. "The anatomy of the facebook social graph". In: *arXiv preprint arXiv:1111.4503* (2011).

[28] Peng Wang et al. "Link prediction in social networks: the state-of-the-art". In: *Science China Information Sciences* 58.1 (2015), pp. 1–38.

[29] Min-Chun Yu et al. "ResearchGate: An effective altmetric indicator for active researchers?" In: *Computers in human behavior* 55 (2016), pp. 1001–1006.

[30] Jing Zhang, Jie Tang, and Juanzi Li. "Expert finding in a social network". In: *International conference on database systems for advanced applications*. Springer. 2007, pp. 1066–1069.

[31] Muhan Zhang et al. "An end-to-end deep learning architecture for graph classification". In: *Thirty-second AAAI conference on artificial intelligence*. 2018.

[32] Zhi-Hua Zhou. *Machine Learning*. Springer Science & Business Media, 2021.

CHAPTER 2

SEGCECO: Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication

2.1 Introduction

In the graph domain, link prediction is the problem of predicting the existence of a connection between two entities in a network. Given a network with various nodes connected to one another, we want to predict if two nodes are connected or are likely to connect in the future. With graph neural networks (GNN), we use not only network structural information, such as connections between nodes, but also individual node characteristics including the feature set of the node. Predicting friendship links among users in a social network, predicting co-authorship links in a citation network, and predicting interactions between genes and proteins in a biological network are some examples of link prediction.

On the other hand, cell-cell interactions regulate organism development by cell functions. A disease may occur when cells do not interact properly or decode molecular messages improperly. Thus, identifying and quantifying inter-cellular signaling pathways has become a common analysis carried out across a variety of fields [4].

With the rapid advancement of single-cell RNA sequencing technologies, researchers are becoming more interested in inferring cell-cell communication from single-cell (scRNA-seq) data. There are a variety of computational tools and resources including ProximID [6], CellChat [13], CellTalker [8], iTalk [29], SingleCellSignalR [7], CellPhoneDB [10], SpotSC [27], and scTensor [25], among others, which are available to predict cell-cell communication (CCC) using gene expression profile obtained from scRNA-seq data.

Generally, in scRNA-seq data analysis, cells are clustered based on their gene expression profiles, and cell types are determined and assigned to clusters based on the known marker genes. CCC tools mostly predict the inter-cellular communications, on the other hand, based on ligand-receptor interactions between pairs of clusters, i.e., cell types, in which one cluster is the source and the other is the target. The majority of the tools are made up of two main components: 1) a prior knowledge resource of intercellular interactions and 2) a method for estimating CCC based on known interactions and the present dataset. Each tool uses different methods, such as permutation of cluster labels, differential combinations, regularizations, and scaling, depending on the input datasets. These approaches result in a varied scoring system which makes it difficult to compare and evaluate the performance of CCC methods. Thus, selecting the appropriate tool to produce the best results is challenging [9]. A recent review study [4] discusses several existing tools for measuring cell-cell communication.

In this work, to predict cell-cell communication, we resort to various approaches that have been successfully used for other existing link prediction problems, such as prediction of social connections between users in social networks [16]. Traditional approaches include heuristic methods such as common neighbors (CN) [18], Adamic Adar (AA) [1], and Resource Allocation (RA) [36]. Heuristic link prediction methods use network structure, i.e. network topology information, in the prediction

process. Existing algorithms can be classified based on the maximum hop of neighbors required to calculate the score [32]. Common neighbors (CN), for example, are **first-order heuristics** that involve the target nodes' one-hop neighbors. Also, some supervised approaches are used for connection prediction, including support vector machine (SVM), baggings, and naives bayes, which are used to model the problem as a binary classification in which extraction of edge features is fundamental.

Moreover, recent methods are mostly built on top of node embedding methods (e.g., DeepWalk [19], node2vec [11], and structural deep network embedding [26]), with the edge representation constructed from the interaction between corresponding node embeddings.

We discovered that some methods perform well on certain types of networks. For instance, every heuristic technique is based on some assumptions and works based on the extracted pattern from the network topology, which is why there is no single heuristic method that works well for all types of networks. Thus, this is a significant drawback in heuristic approaches. The same can be said about latent approaches, which achieve high accuracy in some types of networks but low accuracy in others. Thus, deciding on the best link prediction approach is usually a trial-and-error process.

On the other hand, Weisfeiler-Lehman Neural Machine (WLNM) [33] is considered as a state-of-the-art among link prediction methods based on its performance. It is a new approach based on the subgraph extraction around both target nodes u and v. The local enclosing subgraph for a node pair (u, v) is the subgraph induced from the network by the union of u and v's neighbors up to h hops. The hop is the maximum distance that node features can travel. This approach gives higher accuracy than heuristic and latent methods but requires additional computation time and memory.

In addition, SEAL (Learning from Subgraphs, Embeddings, and Attributes for

Link Prediction) [32] is also a subgraphing method that addresses a number of weaknesses that WLNM has. To begin with, it enables learning not only from subgraph structures but also from latent and explicit node attributes, allowing it to incorporate a variety of information. Secondly, the fully-connected neural network in WLNM is replaced by a GNN that enables graph feature learning improvement. SEAL derived γ decaying theory and proved that a small number of hops is enough to extract high-order heuristics and outperform WLNM. As a result, we choose SEAL as the baseline for predicting links between cells in our proposed framework, SEGCECO. It is a novel method that predicts cell-cell communication in scRNA-seq data via a gene expression attributed graph convolutional network. To our knowledge, this is the first time that graph-based methods are used for prediction of cell-cell communication prediction.

Also, to obtain more precise results, nodes in cell-cell communicating networks (CCN) represent the cells instead of groups of cells, i.e., cell types in our pipeline. Thus, the edges denote the connections (ligand-receptor interactions) between individual cells.

Our study aims to discover cell interactions, with nodes representing cells in the CCN and edges representing cell-cell interactions. Thus, we use similarity matrix-based optimization for scRNA-seq data analysis tool (SpotSC) [27] to perform such a task. Once the CCN network is constructed, our main goal is to predict links among the cells.

2.2 PRELIMINARIES

2.2.1 k-order proximity or k-hop

Given a node $v \in V$ the $k-$ order proximity of v is defined as the set of q vertices at an edge distance less than or equal to k from v and is denoted by $N_k(v)$ [12].

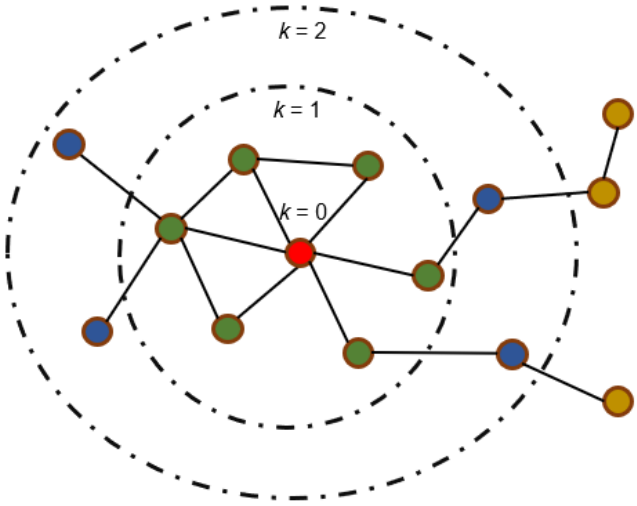

Fig. 2.2.1: $k-$hop proximity of target node marked in red and the neighbors of the target node in the $k-$hop neighborhood within $k = 0, 1,$ and 2.

It is also known as neighborhood of radius k or $k-$hop neighborhood or $k-$order neighborhood.

2.2.2 Subgraph

Given a set of nodes V, the subgraph formed by S is a graph that has s_v as its set of vertices such that $s_v \in V$, and contains every edge of a graph G whose endpoints are in S.

2.2.3 Neighborhood Subgraph

The neighborhood subgraph of radius k of the target node $v \in V$ is the subgraph induced by the neighborhood of radius k of v, and v itself and is denoted by ${S^k}_v$ [32].

Fig. 2.2.1 depicts $k-$order proximity of target node marked in red and the neighbors of the target node in the $k-$hop neighborhood within radius $k = 0, 1, and\ 2$. The subgraph S_v^1 is the graph with target node (marked in red) and its 1-hop ($k = 1$) neighborhood shown in green; as well as the edges connecting them. Similarly, The subgraph S_v^2 is the graph with target node (marked in red) and its 2-hop ($k = 2$)

neighborhood shown in blue; as well as the edges connecting them.

2.2.4 Latent Feature Methods

Given a network G with a finite set of nodes (or vertices) V and finite set of edges E, latent features are the features or representations of nodes V computed using matrix factorization. The matrix can be the adjacency matrix or the Laplacian matrix derived from the network G. For each node, low dimensional embedding is learned by factorization [31]. Node2vec [11], LINE [22], and DeepWalk [19] are examples of network embedding algorithms that learn low-dimensional embedding for nodes. In [20], these network embedding methods were found to implicitly factorize some network matrix representation. Thus, we use them as latent feature methods for learning latent features through factorizing some matrices. Some of the latent feature methods are summarized as follows:

- **Node2Vec:** The Node2vec [11] model for graph learning is an application of the Word2vec paradigm. The latter is a framework for word embedding used to learn continuous feature representations of nodes in networks. The skip-gram model is used to learn continuous feature representations for words. Its goal is to optimize a neighborhood-preserving likelihood objective in order to learn these representations. As an extension of the skip-gram architecture of networks, Node2vec is an embedding approach that works on neighbor nodes and generates low dimensional embeddings, by converting graphs (or networks) into numerical representations. A second-order random walk approach is used to generate the numerical representation of the nodes in the graph. The idea behind Node2Vec is to use flexible, biased random walks that can trade off between local and global network views. This approach returns feature representations that maximize the likelihood of preserving network neighborhoods of nodes in a d-dimensional feature space [11].

- **DeepWalk:** DeepWalk [19] learns d- dimensional latent feature representa-

tions using local information obtained from uniform random walks. To capture network topology information, Deepwalk introduced an unsupervised strategy that learns features that capture the graph structure independently of the labels' distribution, rather than mixing the label space as part of the feature space [19].

- **LINE:** LINE [22] is a network embedding model designed for embedding very large-scale information networks, which contain millions of nodes and billions of edges. This method generates low-level embeddings by preserving both first-order and second-order proximity of nodes. Furthermore, this method incorporates a novel edge-sampling technique that improves the efficiency of the model [22].

- **Spectral Clustering:** Spectral Clustering, SC, is a matrix factorization [31] technique that performs an eigen decomposition of graph G, more specifically, the normalized Laplacian matrix L, and takes top k eigen vectors as the feature representation of nodes, i.e., node embedding vectors, Z. The edge score is calculated as the sigmoid function, $Z \times Z^T$.

- **VGAE (Variational Graph Autoencoder):** Graph Autoencoders [28] are based on graph neural networks that use matrix factorization [31] to map graph input onto a low-dimensional space. Variational Graph Autoencoder [15] is a framework that uses the idea of Variational Autoencoders (VAE) to improve prediction performance on graph-structured data. This model makes use of latent variables and can learn interpretable graph latent embedding for graphs. The model incorporates the node features by using a graph convolutional network (GCN) [14] encoder and a simple inner product decoder.

2.2.5 Subgraph-based Methods

- **WLNM (Weisfeiler-Lehman Neural Machine):** WLNM is a subgraph-based link prediction method that extracts the enclosing subgraphs around the target nodes to learn graph structure features for link prediction. The number of nodes in the subgraph, which is denoted by the user-defined integer K, is explicitly set. The Palette-WL algorithm, a variant of WL that is fast and order-preserving, is used to label nodes. The enclosing subgraph is then represented as an adjacency matrix by WLNM. A fully-connected neural network is trained on these adjacency matrices, together with their labels, to learn the existence of links. WLNM has three steps: enclosing subgraph extraction, subgraph pattern encoding, and neural network training. WLNM has several drawbacks and limitations [33], which is resolved by our proposed method. First, fully-connected neural network in WLNM is replaced by GNN, which improves graph feature learning capabilities. Second, SEGCECO enables to learn not just from subgraph structures, but also from latent and explicit node properties, allowing to extract quite relevant of information.

- **SEAL (Learning from Subgraphs, Embeddings and Attributes for Link Prediction:** SEAL framework for link prediction learns general graph structure features from local subgraphs rather than complete networks. The method takes as input the enclosing subgraphs around the links and returns the likelihood that the links exist. SEAL consists of three steps: enclosing subgraph extraction, node information matrix creation, and GNN learning. The default GNN used in SEAL is DGCNN (Deep Graph Convolutional Neural Network) [Section 2.3.2.4] [32].

2.2.6 SoptSC: Similarity-matrix based optimization for single-cell data analysis

SoptSC successfully performs multiple inference tasks such as unsupervised clustering, pseudotemporal ordering, lineage inference, and marker gene identification based on a cell–cell similarity matrix. The cell-cell similarity matrix S is learned from original scRNA-seq data matrix, i.e., gene expression matrix X of size $m \times n$ with m genes and n cells, using a low-rank representation model [37]. The element S_{ij} $(=S_{ji})$ of similarity matrix S measures the degree of similarity between cell i and cell j [27]. Also, a cell-cell communication graph G is constructed using adjacency matrix A, which is derived from similarity matrix S, where $A_{ij} = 1$ if $S_{ij} > 0$, or $A_{ij} = 0$ otherwise.

SoptSC is an R package available at: `https://mkarikom.github.io/RSoptSC`. In this work, we constructed the cell-cell communication network using this method.

2.2.7 Information Gain

Information gain (IG), as a feature selection method, computes the reduction in entropy by splitting the dataset based on a given value of a random variable and measures how important or relevant the feature is. This is done by estimating the information gain from each variable and choosing the one with the maximum value. Based on Equation (1), the largest information gain is equal to the smallest entropy. IG is calculated by subtracting the weighted entropy values from the original entropy values by following Equation (2). In other words, IG measures how changes to the dataset affect the distribution of the classes or target variables.

$$H(X) = -\sum p(X) \log p(X)\,, \tag{1}$$

where for dataset $\mathbf{X} = \{x_i\}$, $H(X)$ is the probability of randomly picking an element of the class.

$$I(X, a) = H(X) - H(X|a)\,, \tag{2}$$

where *I(X,a)* represents the information gain in dataset $\mathbf{X} = \{x_i\}$ for variable a, $H(X)$ is the original entropy of X and $H(X|a)$ is the conditional entropy for the given variable a.

2.3 Materials and Methods

Our proposed method consist of three main steps: 1) Preprocessing step (Fig. 2.3.1), 2) Cell-cell communication network (CCN) creation (Fig. 2.3.1), and 3) Applying the GCN (Fig. 2.3.1). Before applying the GCN, the primary step is to preprocess the data for downstream analysis (Section 2.3.1). Once the data is preprocessed, a CCN is constructed using SoptSC (Section 2.2.6) in Step 2 (Fig. 2.3.1). The last module of the pipeline the proposed GCN for prediction, SEGCECO, which takes processed scRNA-seq data and the CCN to create an attributed graph dataset, and then predict the output. The preprocessing and SEGCECO framework are explained in the next sections.

2.3.1 Data Preprocessing

Prior to scRNA-seq data analysis, a critical step is to preprocess the data to reduce the effects of noise in the samples. To this end, we followed a standard preprocessing pipeline in scRNA-seq data analysis [17]. This step includes basic filtering, normalization, log transformation and scaling, as shown in the first step of the pipeline depicted in Fig. 2.3.1. Low-quality cells would hamper downstream analysis. These cells may have been damaged or dead during processing, and are represented by the low number of expressed genes. Based on the pipeline [17], cells with less than 200 expressed genes, and genes expressed in less than three cells are filtered out. For example, in BHuman1, we filtered out 5,387 low-expressed genes that are detected in

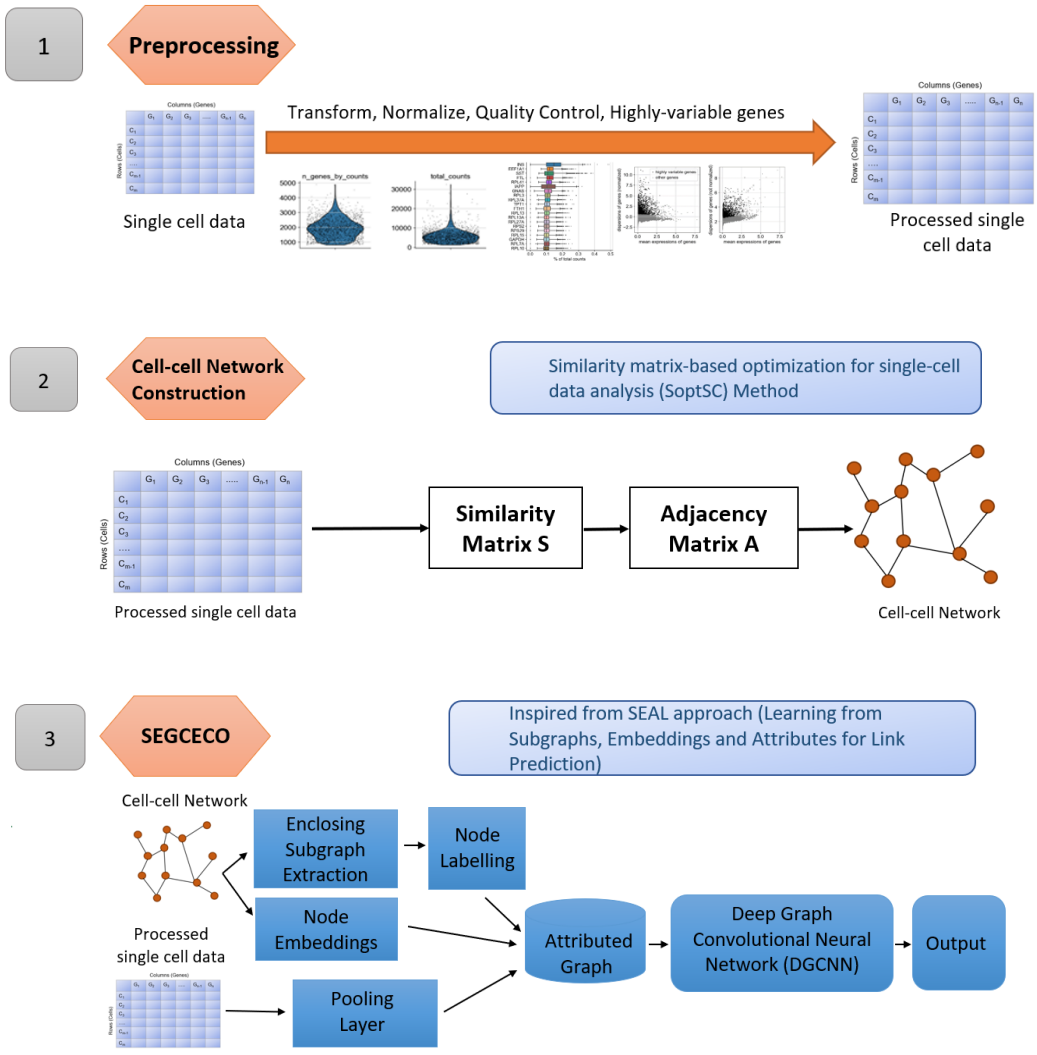

Fig. 2.3.1: Pipeline of the proposed framework for prediction of cell-cell communication.

less than three cells and kept 14,739 genes. We further investigated the distribution of the data, (Fig. 2.3.2, as a data-specific quality-control step to filter low-quality cells. The number of genes expressed in the count matrix is typically between 500 and 4,000 genes, with a dense distribution of the number of expressed genes over the total count per cell for less than 4,000 genes. As such, we filtered out seven cells to remove low-quality ones. This step is performed to remove low quality cells and poorly expressed genes.

Normalization is performed to balance the data by bringing it to a common scale without changing any values or losing any information. The top genes in the dataset are visualized before and after normalization in Fig. 2.3.3 and 2.3.4, respectively. The Counts Per Million (CPM) normalization method is used to normalize the data. Once normalization is performed, data matrices are $log(x + 1)$ transformed to mitigates the mean-variance relationship in single-cell data. The differences in log-values represent log-fold variations in expression which is the standard approach to measure changes in expression. With the use of log transformation on the cells, it compresses the variation into a less extreme range making relative differences between cells easier to observe [23]. Finally, log transformation minimizes the skewness of the data, allowing many downstream analysis tools to approximate the assumption that the data is normally distributed [17].

After per-gene quantification, we selected a subset of highly variable genes to use in downstream analyses as they are informative of the variability in the data. To achieve this, we chose a commonly used technique in [2] and defined the set of highly variable genes given a normalized dispersion higher than 0.5 after normalization, yielding 2,546 genes.

For preprocessing, we used Scanpy [30], a specifically designed package to analyze scRNA-seq datasets. Scanpy includes methods for preprocessing, visualization, clustering, pseudotime and trajectory inference, differential expression testing, and

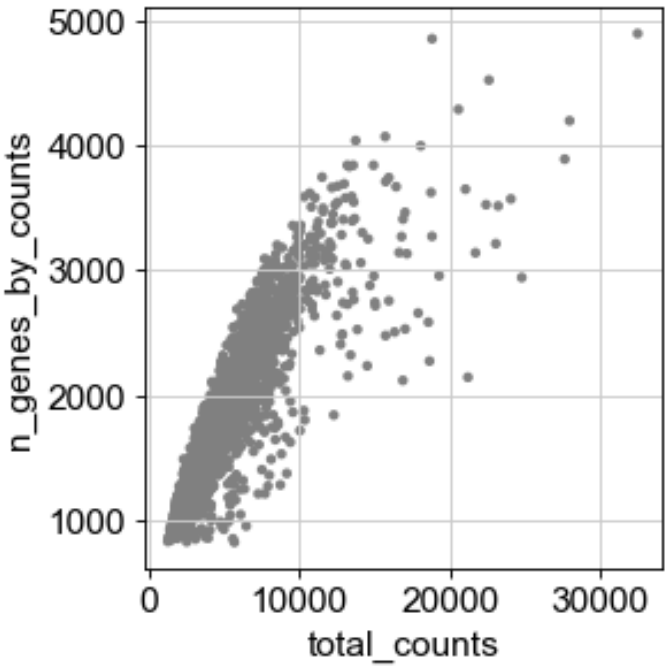

Fig. 2.3.2: Distribution of the data (BHuman1).

simulation of gene regulatory networks. The frameworks like Seurat [21], Monocle [24] and Cell ranger [35] do not scale to huge data sets with up to and above one million cells. Therefore, a framework Scanpy addresses these issues while still allowing similar analyses. It is an easy interface with advanced machine-learning packages [30].

Scanpy incorporates scalable canonical analysis methodologies. For example, it provides preprocessing comparable to Seurat [21] and Cell ranger [35], visualization through TSNE [3], and much more. It has been tested against other packages for single-cell analysis. The result shows scanpy provides tools with speedups that allow for interactive analysis of data sets with more than one million cells and run times of the order of seconds for roughly 100,000 cells [30]. Due to the abovementioned reasons, we used Scanpy for data preprocessing.

2.3.2 SEGCECO Framework

SEGCECO framework consists of four main phases: 1) Feature (gene) selection in the pooling layer, 2) Subgraph extraction, 3) Node information matrix construction,

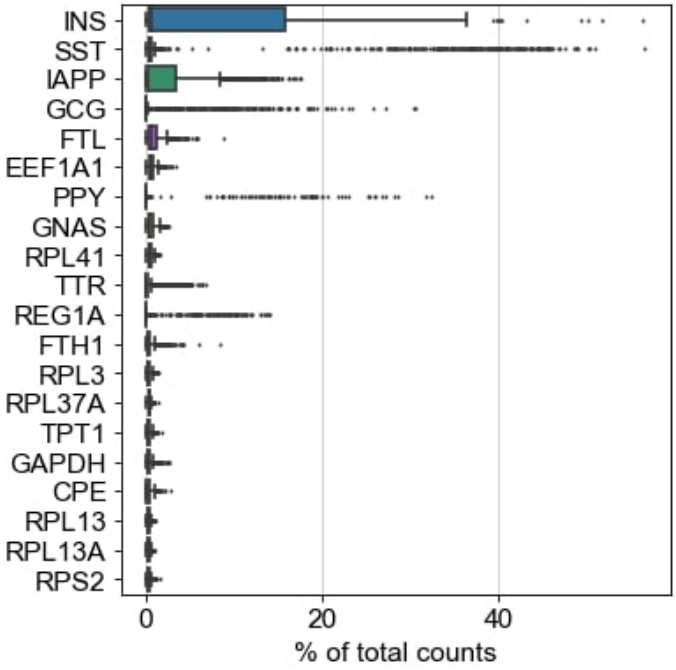

Fig. 2.3.3: Highly variable genes before normalization (BHuman1).

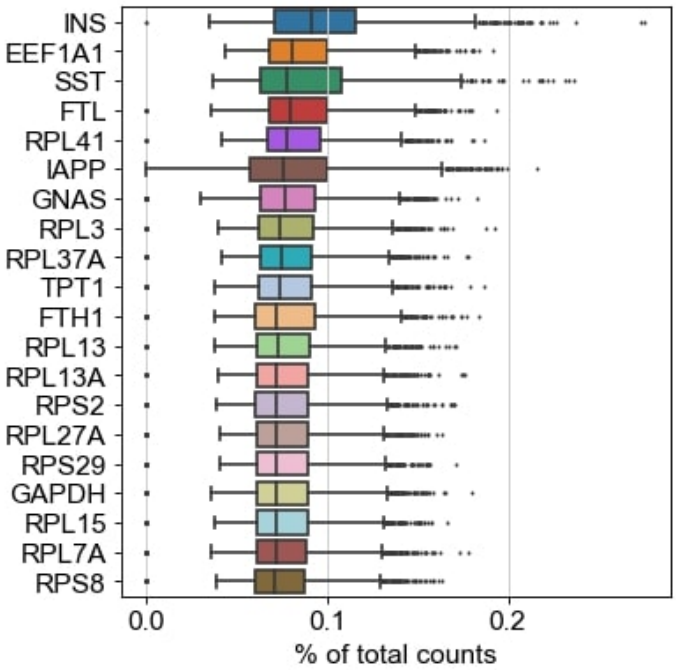

Fig. 2.3.4: Highly variable genes after normalization (BHuman1).

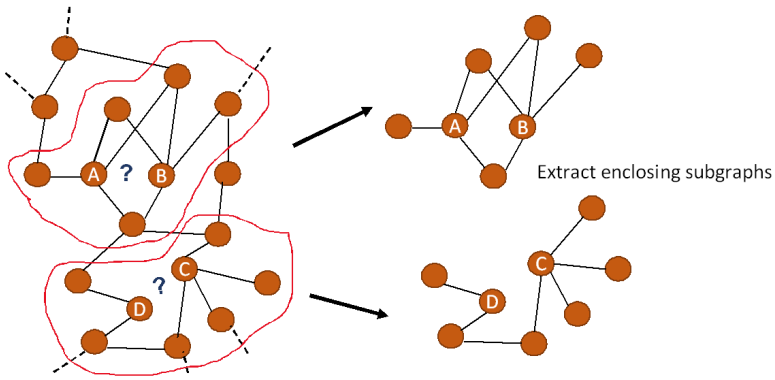

Fig. 2.3.5: 1-hop enclosing subgraphs for target nodes (A,B) and (C,D).

and 4) DGCNN learning. All these phases are explained in the next few sections.

2.3.2.1 Gene Selection in Pooling Layer

Downsampling is crucial in graph analysis, which is included in the pooling layer of the SEGCECO framework. Downsampling here means reducing the number of features (i.e., genes) of the nodes. The pooling layer consists of selecting genes (with a threshold τ) by IG (Section 2.2.7) feature selection. This step provides the node attribute information (side information) of each individual node, i.e., explicit features.

2.3.2.2 Enclosing Subgraph Extraction

Enclosing subgraph extraction involves extracting the local enclosing subgraphs around the target nodes u and v. The subgraph induced from the network by the union of u and v's neighbors up to $k-$ hops is called the enclosing subgraph for a node pair (u, v). In the next phase, the enclosing subgraph is extracted from the training data, which contains both positive (existent) and negative (non-existent) sets of sampled links, based on h-hop neighbors for the target nodes u and v. Figure 2.3.5 depicts an example of the 1-hop enclosing subgraphs for target nodes (A, B) and (C, D).

2.3.2.3 Node Information Matrix Construction

In the node information matrix, each row corresponds to the node's feature vector, which is represented as X. In the SEAL [32] approach, there are three components

in the node information matrix:

- **Node Labeling:** The Double-Radius Node Labeling (DRNL) algorithm (explained in [32]) is used to label the nodes based on their structure. The main purpose of this step is to label every node in the enclosing subgraph in order to distinguish the target nodes between which a link should be expected.

  Labels are assigned to nodes in such a way that the target nodes u and v are labeled 1. Second, the radius of node i with respect to two target nodes, namely $(d(i,u), d(i,v))$, can be used to define its position. Thus, nodes on the same orbit are given the same label. In other words, larger labels are assigned to nodes that have larger radius with respect to target nodes. This algorithm can be better understood by following the diagram of Fig. 2.3.6, which satisfies the following conditions:

  1. if $d(i,x)+d(i,y) \neq d(j,x)+d(j,y)$, then $d(i,x)+d(i,y) < d(j,x)+d(j,y) \Leftrightarrow f_l(i) < f_l(j)$;
  2. if $d(i,x) + d(i,y) = d(j,x) + d(j,y)$, then $d(i,x)d(i,y) < d(j,x)d(j,y) \Leftrightarrow f_l(i) < f_l(j)$.

  where $f_l(i)$ is the label assigned to node i and $(d(i,x), d(i,y))$ is the double radius.

- **Node Embedding:** Trick **negative injection**, as explained in [32], is used to generate the node embeddings. A trick consists of adding the negative (non-existent) set of sampled links, E_n, to the positive (existent) set of sampled links, E, and generate the embeddings on $G' = (V, E \cup E_n)$. The node embedding method used in our method is Node2vec [11].

- **Node Attributes:** Both latent and explicit features of each node are included in the node information matrix X for its corresponding row in X. Latent feature

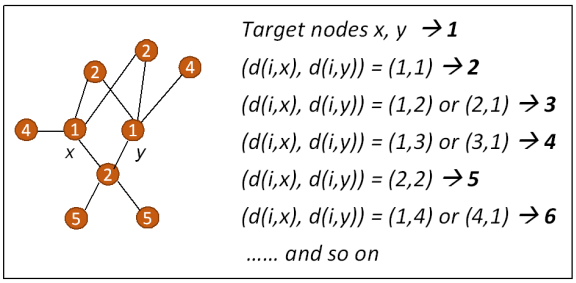

Fig. 2.3.6: Node labeling approach.

methods learn low-dimensional latent representation or embedding for each node using matrix factorization [31]. An adjacency or Laplacian matrix derived from the graph can be used. Node embedding techniques such as Node2vec [11], LINE [22] and DeepWalk [19] are some of the examples of latent feature methods which we have used in this study. The explicit features contain side information about the individual nodes, available in the form of node attributes. The SEAL method [32] shows significant improvement in performance when combine both latent and explicit features.

2.3.2.4 Deep Graph Convolutional Neural Networks

Deep Graph Convolutional Neural Networks (DGCNN) [34] is a deep learning architecture used for graph classification. DGCNN is divided into three main parts:

- **Graph Convolutions:** Localised graph convolutions are used to extract the hidden feature information of nodes from the graph and produce a consistent node ordering. DGCNN consists of four graph convolutional layers in which the output of each convolutional layer is passed to a hyperbolic tangent (tanh) non-activation function. A DGCNN architecture of four convolutional layers is shown in Fig. 2.3.7. To extract multi-hop node features, DGCNN stacks multiple graph convolution layers, and concatenates each layer's node states to the final node states [32].

- **SortPooling Layer:** In the SortPooling layer, the unordered node attributes of the graph from spatial graph convolutions layer are fed as the input. The main purpose of this layer is to sort the feature descriptors, each of which represents a node. Rather than summing up these node features, SortPooling layer arranges them in a consistent order and outputs a sorted graph representation with a given size. Then, it can be read and trained by standard convolutional neural networks. Nodes are sorted using graph labeling methods, based on their structural roles, in descending order using the last layer's output, Z^h.

  Once the feature of the nodes are sorted, the next step is to unify the sizes of the output tensor. The main intention behind it is to unify the graph sizes to k by deleting the last $n-k$ rows if $n > k$, or adding $k-n$ zero rows otherwise. The output of SortPooling Layer is shown in Fig. 2.3.8

- **Traditional Convolutional and Dense Layers:** These layers are used to make a prediction based on the sorted graph representations generated by the SortPooling layer.

The architecture of DGCNN is shown in Fig. 2.3.9. Given the adjacency matrix $A \in \{1,0\}^{nXn}$ of graph G with n number of nodes and each node containing the c dimensional feature vector as well as the node information matrix $X \in R^{nXc}$ of an enclosing subgraph with each row representing the node, DGCNN employs the following convolution layer:

$$Z = f(\tilde{D^{-}}1\tilde{A}XW)\,. \tag{3}$$

where $\tilde{A} = A + I$, I is the identity matrix, $\tilde{D}$ is the diagonal degree matrix with $\tilde{D_{i,i}} = \sum_j \tilde{A_{i,j}}$, W is a trainable graph convolutional parameters, f is a non-linear activation function, and $Z \in R^{nXc'}$ is the output activation matrix.

The graph convolution incorporates each nodes' hidden representation by aggregating attribute information from its neighbors. The graph convolution can be split

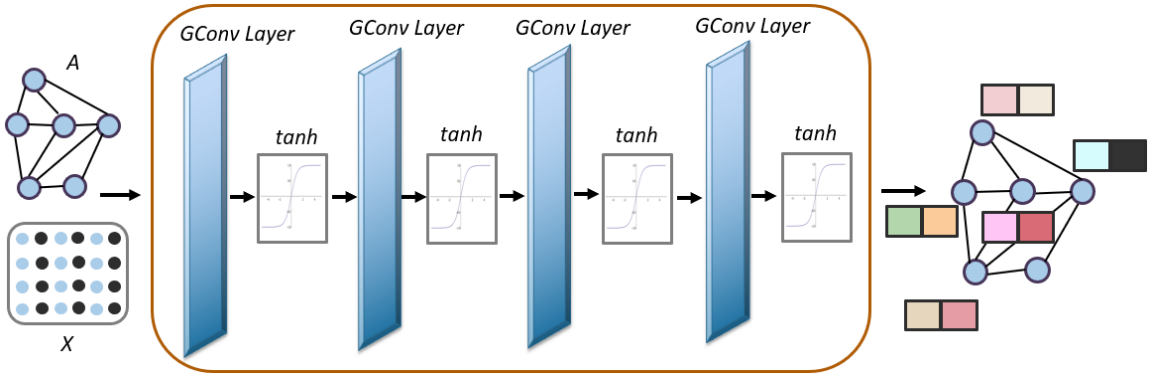

Fig. 2.3.7: Schematic view of DGCNN architecture with four graph convolutional layers used in this work.

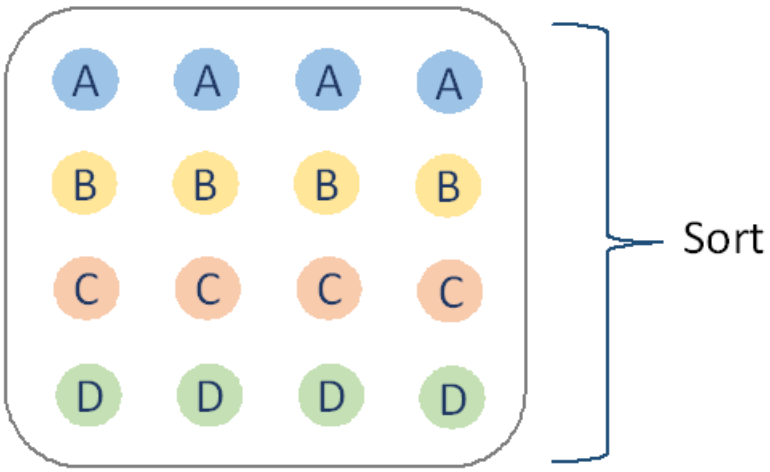

Fig. 2.3.8: Overview of the SortPooling layer's output.

into four different steps:

1. Linear feature transformation is applied to the node information matrix X, by multiplying it by W.

2. Node information is propagated to neighboring nodes as well as the node itself by $\tilde{A}XW$.

3. Each row is normalized by multiplying it by D^{-1}.

4. Non-linear activation function is applied to obtain the output.

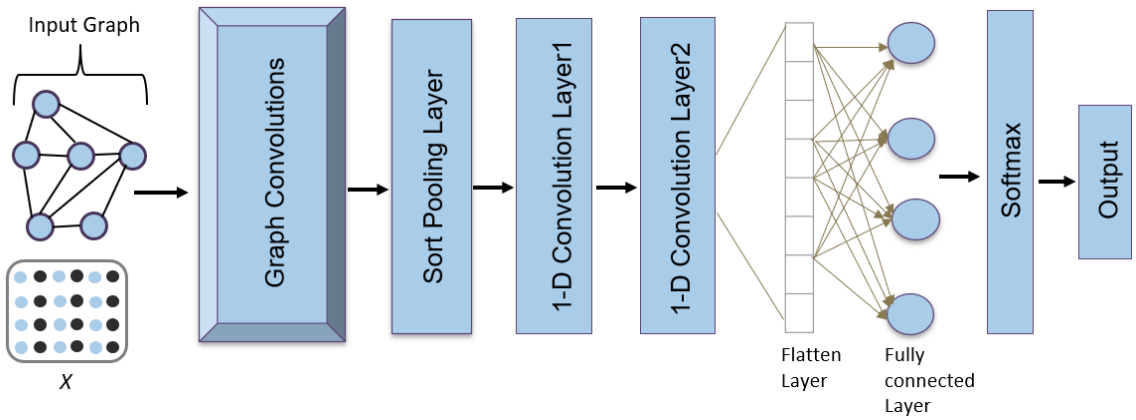

Fig. 2.3.9: Overview of the DGCNN architecture.

2.3.3 Datasets

The datasets used in this study are publicly available annotated scRNA-seq data from human and mouse pancreas tissue, drawn from the NCBI's Gene Expression Omnibus, accession number GSE84133 [5]. The datasets were generated by following the inDrop method under Illumina HiSeq 2500 to determine the transcriptomes of over 12,000 individual pancreatic cells from four human donors and two mice strains. We used the data taken from human donors with the accession numbers GSM2230757, GSM2230758, GSM2230759, and GSM2230760, as well as mice strains with the accession numbers GSM2230761 and GSM2230762. Pancreatic cells are divided into 14 different clusters of previously characterized cell types: all endocrine cell types, including rare ghrelin-expressing epsilon-cells, exocrine cell types, vascular cells, Schwann cells, quiescent and activated pancreatic stellate cells, and four types of immune cells. Table 2.3.1 depicts the details of datasets including tissue, the accession number, the number of cell types, the number of cells, and the number of genes.

2.3.4 Performance Evaluation

The node embedding methods, i.e., Node2Vec, LINE and SC, give the feature representation of nodes in a graph. Thus, an additional step is required to learn the features of the edges from node embedding in order to predict links as a binary classification problem. To evaluate our method, we use the binary operator over the

Table 2.3.1: Details of the datasets used in this work including tissue, the accession number, the number of cell types, the number of cells, and the number of genes.

Dataset	Tissue	Accession	# Cells	# Genes
Baron-human1 (BHuman1)	Human-Pancreas	GSM2230757	1,937	20,125
Baron-human2 (BHuman2)	Human-Pancreas	GSM2230758	1,724	20,125
Baron-human3 (BHuman3)	Human-Pancreas	GSM2230759	3,605	20,125
Baron-human4 (BHuman4)	Human-Pancreas	GSM2230760	1,303	20,125
Baron-mouse1 (BMouse1)	Mouse-Pancreas	GSM2230761	822	14,878
Baron-mouse2 (BMouse2)	Mouse-Pancreas	GSM2230762	1,064	14,878

corresponding feature vectors of nodes u and v, i.e., $f(u)$ and $f(v)$, to generate the edge embedding $g(u, v)$ for edge $e = (u, v)$, as used in [11].

- **Average:**

$$fx(u) \boxplus f(v) = \frac{f(u) + f(v)}{2} \,. \tag{4}$$

- **Hadamard:**

$$f(u) \boxdot f(v) = f(u) * f(v) \,. \tag{5}$$

- **Weighted-L1:**

$$||f(u) \cdot f(v)||_{\overline{1}} = |f(u) - f(v)| \,. \tag{6}$$

- **Weighted-L2:**

$$||f(u) \cdot f(v)||_{\overline{2}} = |f(u) - f(v)|^2 \,. \tag{7}$$

2.4 Experimental Results

For performance comparison, we included WLNM, GAE and VGAE, as well as four state-of-the-art latent feature methods including Node2Vec, LINE, Deepwalk and SC. These methods are explained in Section 2.2.4 and 2.2.5. We used Area Under Curve (AUC), accuracy, precision, recall, F1-score and receiver operating characteristic curve (ROC curve) as evaluation metrics. To calculate the evaluation metrics, we used training and testing data which consists of both positive (existent) and negative (non-existent) links. As a negative set, we randomly chose an equal number of unconnected pairs of nodes from the network with no edge connection between them. We arbitrarily remove 10% of links as testing data and the remaining 90% are used as training data. The statistical information of the network extracted from datasets (discussed in Section 2.3.3) is shown in Table 2.4.1.

Table 2.4.1: Statistical information from the Network of Datasets.

Dataset	**# Nodes**	**# Edges**	**Average Node Degree**
Baron-human1 (BHuman1)	1,930	33,941	35.172
Baron-human2 (BHuman2)	1,724	30,223	35.0615
Baron-human3 (BHuman3)	3,597	62,850	34.9458
Baron-human4 (BHuman4)	1,282	22,729	35.4587
Baron-mouse1 (BMouse1)	821	14,774	35.9903
Baron-mouse2 (BMouse2)	1,061	18,791	35.4213

2.4.1 Implementation Details

We evaluated the **latent feature methods** supplied in this study using the authors' original code for Node2vec, Deepwalk, and LINE.

Also, we used SC for evaluation. The node embeddings generated from these methods are used to generate the link embeddings as explained in Section 2.2.4.

Hyperparamaters for Node2vec							
Dimension	walk length	No. of walks	window size	p	q	iteration	worker
16	80	10	10	1	1	1	8
Hyperparamaters for DeepWalk							
Representation-size	walk length	No. of walks	window size	workers			
16	40	10	5	1			
Hyperparamaters for LINE							
size	binary	order	negative	rho	threads		
16	0	2	5	0.025	1		
Hyperparameters for GAE & VGAE.							
learning_rate	epochs	Hidden Layer1 Units #	Hidden Layer2 Units #	features			
0.01	200	32	16	0			
Hyperparameters for DGCNN							
GConv1	GConv2	GConv3	GConv4	k	Conv 1D_1 Output	Conv 1D_2 Output	
32	32	32	1	60	16	32	
Hyperparameters for SEGCECO							
learning_rate	epochs	dimension	hop	Pooling Layer (τ)			
0.00001	100	16	1,2	300			

Table 2.4.2: Summary of hyperparameters used by methods

Then, we used logistic regression as the classifier to predict the links. Ten-fold cross-validation is used to test and train the model.

To evaluate the performance of other methods including GAE, VGAE, WLNM, we used the default settings.

To implement the core of our method, SEGCECO, we used the base implementation of the SEAL method. Then, to evaluate the performance of the results, we used 90%-10% of data as training and testing set respectively. To generate FPR/TPR distribution of the datasets, we have taken the mean of the corresponding values.

2.4.2 Hyperparameter Tuning

We used different hyperparameters for each method as described in Table 2.4.2.

2.4.3 Discussion

Overall, compared to other methods, SEGCECO achieves improvement in performance in terms of AUC. Table 2.4.3 shows the performance (AUC) of SEGCECO and latent methods. For all the datasets, Node2vec outperformed all other approaches for three of the four operators. It means Node2vec excels in generating low-dimensional

embeddings of nodes in networks and achieving a neighborhood-preserving objective. Thus, we chose Node2vec as the node embedding method in our framework.

Table 2.4.4 depicts the performance (AUC) of SEGCECO with other GNN-based methods, such as GAE, VGAE, WLNM and SEGCECO. Among them, SEGCECO performs best with approximately 0.99 AUC. We anticipate that the improved performance of SEGCECO is due to the proposed pooling layer in the framework, which uses IG as the feature selection method to select the top τ attributes (i.e., genes) as explicit features in the node information matrix, X, resulting in better prediction.

Moreover, Fig. 2.4.1 plots the ROC curve for DeepWalk, Node2vec, LINE, SC, GAE, VGAE, WLNM, and SEGCECO on BHuman1 dataset. It is noticeable that SEGCECO surpasses other approaches since the curve is closer to the top-left corner, indicating better performance. Here, positive means interacting cells and negative means non-interacting cells. By observing Fig. 2.4.2, reveals that SEGCECO obtained the lowest FPR of 0.0135 among all the approaches, implying that there is a very lower probability that SEGCECO will predict non-interacting cells as interacting cells. This, in other words, means when the cells do not have interactions, the chances of inaccurate predictions, i.e., the cells interacts, are minimal. Furthermore, the SEGCECO performs best in predicting actual interactions, that is, when there exist interactions between cells, the method predicts the same. The same behavior is detected in other datasets as well. The ROC curves, FPR and TPR distribution on other datasets can be found in the Section Appendix A (Figs. A.0.1-A.0.10). Thus, it can be concluded that SEGCECO yields the best results for all datasets when it regards to distinguishing between interacting and non-interacting cells and making predictions.

Accuracy, Precision, Recall, and F1-score are the commonly used evaluation metrics to illustrate the performance of the the model. Recall evaluates the model for correctly identifying the cell-cell communication. Precision shows the percentage of

Table 2.4.3: Comparison of SEGCECO with latent methods for all datasets used in this study.

Operator	Method	BHuman1	BHuman2	BHuman3	BHuman4	BMouse1	BMouse2
Average	Node2vec	0.4999	0.5162	0.5035	0.5177	0.5049	0.5127
	LINE	0.5061	0.5034	0.5053	0.4968	0.5142	0.5093
	DeepWalk	0.5029	0.5052	0.5000	0.5148	0.5099	0.5130
	SC	0.4728	0.5464	0.5361	0.5310	0.5043	0.5312
Hadamard	Node2vec	0.9748	0.9766	0.9833	0.9711	0.9564	0.9726
	LINE	0.7077	0.7908	0.5696	0.8279	0.8474	0.8494
	DeepWalk	0.9560	0.9634	0.9514	0.9615	0.9558	0.9635
	SC	0.9392	0.9625	0.9501	0.9623	0.9589	0.9648
Weighted L1	Node2vec	0.9887	0.9885	0.9917	0.9851	0.9798	0.9846
	LINE	0.7204	0.7474	0.5528	0.8421	0.8940	0.8848
	DeepWalk	0.9867	0.9857	0.9859	0.9820	0.9813	0.9812
	SC	0.9743	0.9757	0.9696	0.9716	0.9690	0.9694
Weighted L2	Node2vec	0.9896	0.9895	0.9919	0.9862	0.9802	0.9846
	LINE	0.7243	0.7474	0.5603	0.8487	0.8989	0.8865
	DeepWalk	0.9869	0.9866	0.9857	0.9825	0.9823	0.9822
	SC	0.9748	0.9752	0.9687	0.9736	0.9752	0.9729
	SEGCECO	0.9985	0.9980	0.9989	0.9982	0.9975	0.9972

predictions accurately made by the model. Table 2.4.5 shows the AUC, accuracy, precision, recall, and F1-score of link prediction using SEGCECO framework on different datasets. The SEGCECO shows a performance of around 99% for all measures, indicating that our model can accurately predict cell-cell interactions and discriminate interacting cells from non-interacting cells.

Based on the findings of the above-mentioned comparison, we conclude that SEGCECO surpassed all other approaches with 99% AUC, accuracy, and other performance measures across all datasets.

Table 2.4.4: Comparison of SEGCECO with other methods for the datasets.

Datset	**GAE**	**VGAE**	**WLNM**	**SEGCECO**
BHuman1	0.9835	0.9852	0.9832	0.9985
BHuman2	0.9859	0.9805	0.9839	0.9980
BHuman3	0.9876	0.9869	0.9889	0.9989
BHuman4	0.9838	0.9764	0.9773	0.9982
BMouse1	0.9841	0.9764	0.9673	0.9975
BMouse2	0.9838	0.9829	0.9744	0.9972

Table 2.4.5: Performance metrics of SEGCECO for the datasets.

Datset	**AUC**	**Accuracy**	**Precision**	**Recall**	**F1-score**
BHuman1	0.9985	0.9928	0.9872	0.9987	0.9929
BHuman2	0.9980	0.9903	0.9915	0.9891	0.9903
BHuman3	0.9989	0.9923	0.9925	0.9921	0.9923
BHuman4	0.9982	0.9886	0.9862	0.9913	0.9887
BMouse1	0.9975	0.9854	0.9800	0.9908	0.9854
BMouse2	0.9972	0.9878	0.9800	0.9954	0.9876

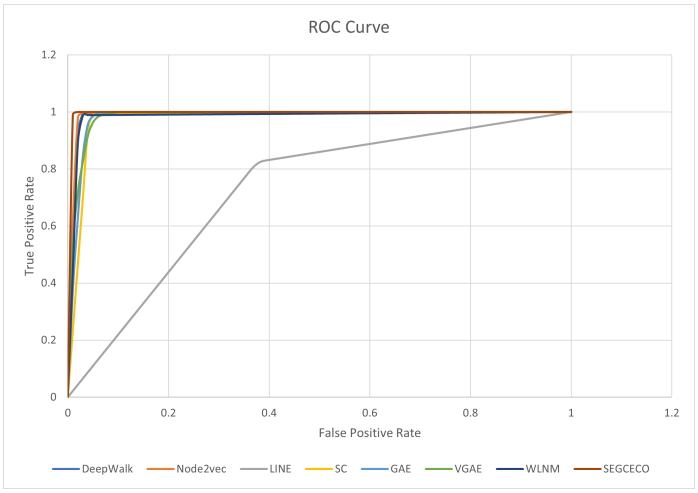

Fig. 2.4.1: ROC Curve for BHuman1 dataset

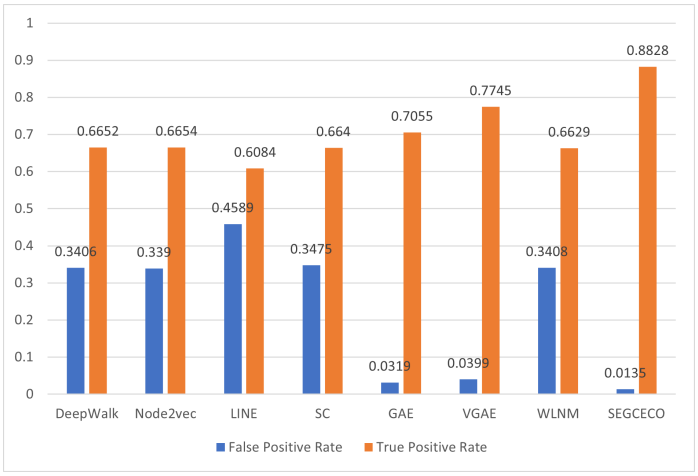

Fig. 2.4.2: False Positive Rate and True Positive Rate distribution of BHuman1 dataset

References

[1] Lada A Adamic and Eytan Adar. "Friends and neighbors on the web". In: *Social networks* 25.3 (2003), pp. 211–230.

[2] Robert A Amezquita et al. "Orchestrating single-cell analysis with Bioconductor". In: *Nature methods* 17.2 (2020), pp. 137–145.

[3] El-ad David Amir et al. "viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia". In: *Nature biotechnology* 31.6 (2013), pp. 545–552.

[4] Erick Armingol et al. "Deciphering cell–cell interactions and communication from gene expression". In: *Nature Reviews Genetics* 22.2 (2021), pp. 71–88.

[5] Maayan Baron et al. "A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure". In: *Cell systems* 3.4 (2016), pp. 346–360.

[6] Jean-Charles Boisset et al. "Mapping the physical network of cellular interactions". In: *Nature methods* 15.7 (2018), pp. 547–553.

[7] Simon Cabello-Aguilar et al. "SingleCellSignalR: inference of intercellular networks from single-cell transcriptomics". In: *Nucleic acids research* 48.10 (2020), e55–e55.

[8] Anthony R Cillo et al. "Immune landscape of viral-and carcinogen-driven head and neck cancer". In: *Immunity* 52.1 (2020), pp. 183–199.

[9] Daniel Dimitrov et al. "Comparison of resources and methods to infer cell-cell communication from single-cell RNA data". In: *BioRxiv* (2021).

[10] Mirjana Efremova et al. "CellPhoneDB: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes". In: *Nature protocols* 15.4 (2020), pp. 1484–1506.

[11] Aditya Grover and Jure Leskovec. "node2vec: Scalable feature learning for networks". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864.

[12] William L. Hamilton. "Graph Representation Learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14.3 (), pp. 1–159.

[13] Suoqin Jin et al. "Inference and analysis of cell-cell communication using CellChat". In: *Nature communications* 12.1 (2021), pp. 1–20.

[14] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[15] Thomas N Kipf and Max Welling. "Variational graph auto-encoders". In: *arXiv preprint arXiv:1611.07308* (2016).

[16] David Liben-Nowell and Jon Kleinberg. "The link-prediction problem for social networks". In: *Journal of the American society for information science and technology* 58.7 (2007), pp. 1019–1031.

[17] Malte D Luecken and Fabian J Theis. "Current best practices in single-cell RNA-seq analysis: a tutorial". In: *Molecular systems biology* 15.6 (2019), e8746.

[18] Mark EJ Newman. "Clustering and preferential attachment in growing networks". In: *Physical review E* 64.2 (2001), p. 025102.

[19] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710.

[20] Jiezhong Qiu et al. "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec". In: *Proceedings of the eleventh ACM international conference on web search and data mining*. 2018, pp. 459–467.

[21] Rahul Satija et al. "Spatial reconstruction of single-cell gene expression data". In: *Nature biotechnology* 33.5 (2015), pp. 495–502.

[22] Jian Tang et al. "Line: Large-scale information network embedding". In: *Proceedings of the 24th international conference on world wide web*. 2015, pp. 1067–1077.

[23] Mehmat Tekman et al. "Analysis of plant scRNA-Seq Data with Scanpy". In: (). URL: Retrieved%20from%20Galaxy%20Training:%20https://training.galaxyproject.org/training-material/topics/transcriptomics/tutorials/scrna-plant/tutorial.html.

[24] Cole Trapnell et al. "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells". In: *Nature biotechnology* 32.4 (2014), pp. 381–386.

[25] Koki Tsuyuzaki, Manabu Ishii, and Itoshi Nikaido. "Uncovering hypergraphs of cell-cell interaction from single cell RNA-sequencing data". In: *BioRxiv* (2019), p. 566182.

[26] Daixin Wang, Peng Cui, and Wenwu Zhu. "Structural deep network embedding". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 1225–1234.

[27] Shuxiong Wang et al. "Cell lineage and communication network inference via optimization for single-cell transcriptomics". In: *Nucleic acids research* 47.11 (2019), e66–e66.

[28] Yingfeng Wang et al. "A simple training strategy for graph autoencoder". In: *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*. 2020, pp. 341–345.

[29] Yuanxin Wang et al. "iTALK: an R package to characterize and illustrate intercellular communication". In: *BioRxiv* (2019), p. 507871.

[30] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome biology* 19.1 (2018), pp. 1–5.

[31] Daokun Zhang et al. "Network representation learning: A survey". In: *IEEE transactions on Big Data* 6.1 (2018), pp. 3–28.

[32] Muhan Zhang and Yixin Chen. "Link prediction based on graph neural networks". In: *Advances in neural information processing systems* 31 (2018).

[33] Muhan Zhang and Yixin Chen. "Weisfeiler-lehman neural machine for link prediction". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 575–583.

[34] Muhan Zhang et al. "An end-to-end deep learning architecture for graph classification". In: *Thirty-second AAAI conference on artificial intelligence*. 2018.

[35] Grace XY Zheng et al. "Massively parallel digital transcriptional profiling of single cells". In: *Nature communications* 8.1 (2017), pp. 1–12.

[36] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. "Predicting missing links via local information". In: *The European Physical Journal B* 71.4 (2009), pp. 623–630.

[37] Liansheng Zhuang et al. "Locality-preserving low-rank representation for graph construction from nonlinear manifolds". In: *Neurocomputing* 175 (2016), pp. 715–722.

CHAPTER 3

Conclusion and Future Work

3.1 Conclusion

Data analysis in the form of graphs is gaining a lot of attention. It has become one of the popular research topics. Link prediction is one of the most important research topics in the field of graphs (or networks). However, although much effort has been made, there is still a lot to do in this field. Cell–cell interaction refers to the direct interactions between cell surfaces that play a crucial role in the development and function of multicellular organisms.

In this study, we propose a pipeline for performing cell-cell interactions prediction in scRNA-seq data using GCN. This article demonstrates how scRNA-seq data in the form of a gene expression matrix is transformed into a graph representation, i.e., a cell-cell communication network (CCN), in order to predict cell-cell interactions in scRNA-seq datasets. To our knowledge, this is the first time that the cell-cell interaction problem is being solved using GCN.

SEGCECO works with undirected, attributed graphs created from individual cell gene expression profiles. The architecture of SEGCECO includes a pooling layer that coarsens the graph attributes from the scRNA-seq data while preserving the global structure of the input graph using the Information Gain method. The pooling layer is followed by the enclosing subgraph extraction, node information matrix construction, and finally GCN that convolves over graphs to encode the representation of both local

and global attributes. In our experiments, SEGCECO has been shown to outperform previous state-of-the-art techniques. We evaluated SEGCECO using AUC, accuracy, precision, recall, and F1-score evaluation metrics. SEGCECO shows a performance of approximately 99% for all performance measures across the datasets. We empirically proved that SEGCECO yields better results in terms of AUC relative to the previously proposed latent and subgraph-based methods. Thus, we conclude that SEGCECO outperforms other approaches in predicting cell-cell predictions and distinguishing interacting from non-interacting cells.

3.1.1 Contributions

The main contributions of this thesis can be summarized as follows:

- We propose a new pipeline by integrating methodologies from state-of-the-art studies for cell-cell interaction prediction in scRNA-seq data.
- We introduce a statistically significant pooling layer that employs information gain as an approach for coarsening graph attributes from the scRNA-seq data, while preserving the global structure of the input graph.
- In the Node Information or Attribute matrix, we include explicit features from the single-cell gene expression matrix.
- We apply the proposed method on different datasets and obtain higher performance compared to the state-of-the-art approaches.
- We have developed an open-source Github project `https://github.com/sheenahora/SEGCECO` for the proposed pipeline.

3.2 Future Work

Below are some tasks that researchers would like to perform in the future:

- SEGCECO also opens up new research opportunities to work with networks in which there is a special structure, for example, heterogeneous CCN, networks with explicit domain features for nodes and edges, directed or multi-modal graphs.

- The model we proposed uses the Information Gain feature selection method to select features (i.e. genes) in the Pooling Layer. To conduct further experiments in the future, we can use different feature selection algorithms and select more features in the pooling layer.

- In addition to the application of SEGCECO on cell-cell prediction (i.e. link prediction), we could apply the proposed method on node classification, node clustering, graph partitioning, and graph classification.

- We would also foresee applying SEGCECO on domains such as disease-gene or drug-target associations, knowledge graph completion, and recommendation systems, among others.

APPENDICES

APPENDIX A

ROC curves as well as FPR and TPR Distribution

The ROC curves as well as FPR and TPR distribution of BHuman2, BHuman3, BHuman4, BMouse1 and BMouse2 datasets are depicted in Fig. A.0.1 - Fig. A.0.10.

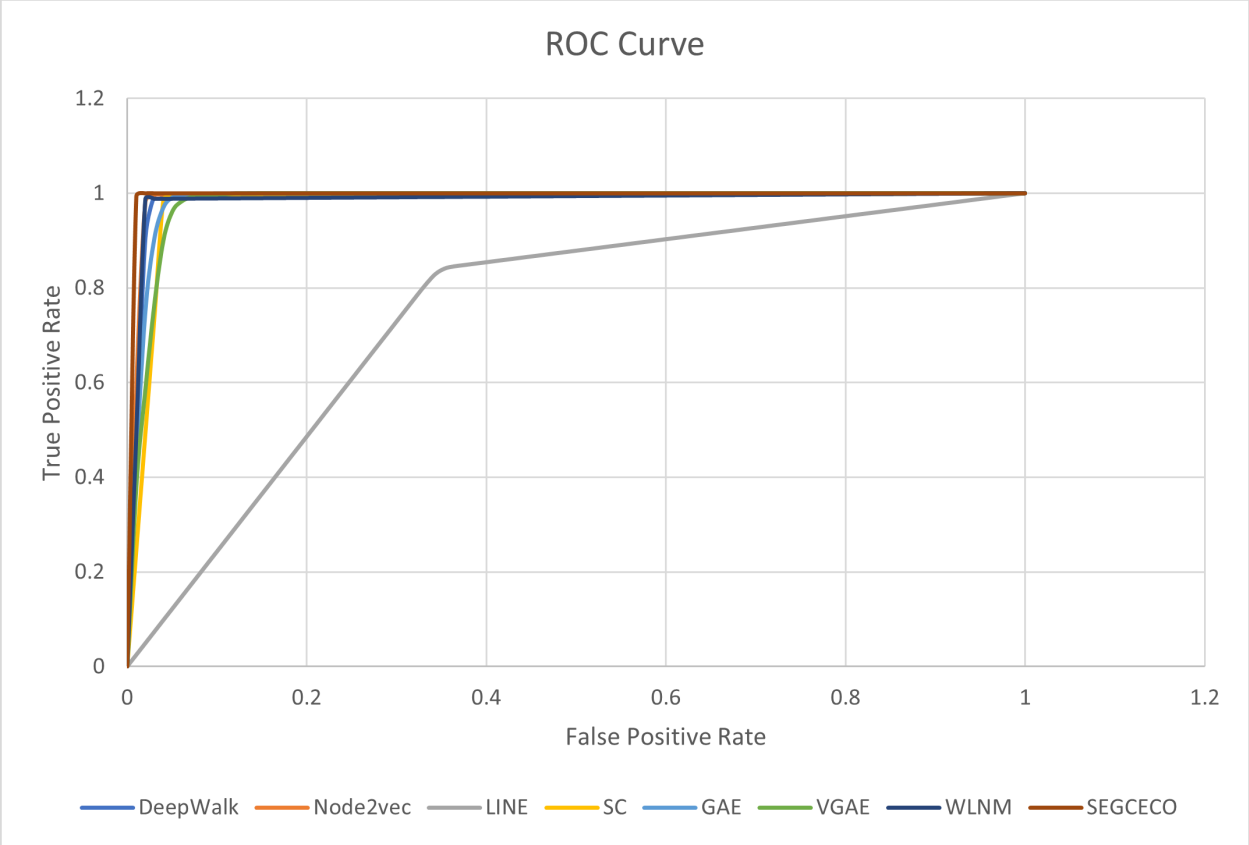

Fig. A.0.1: ROC Curve for BHuman2 dataset

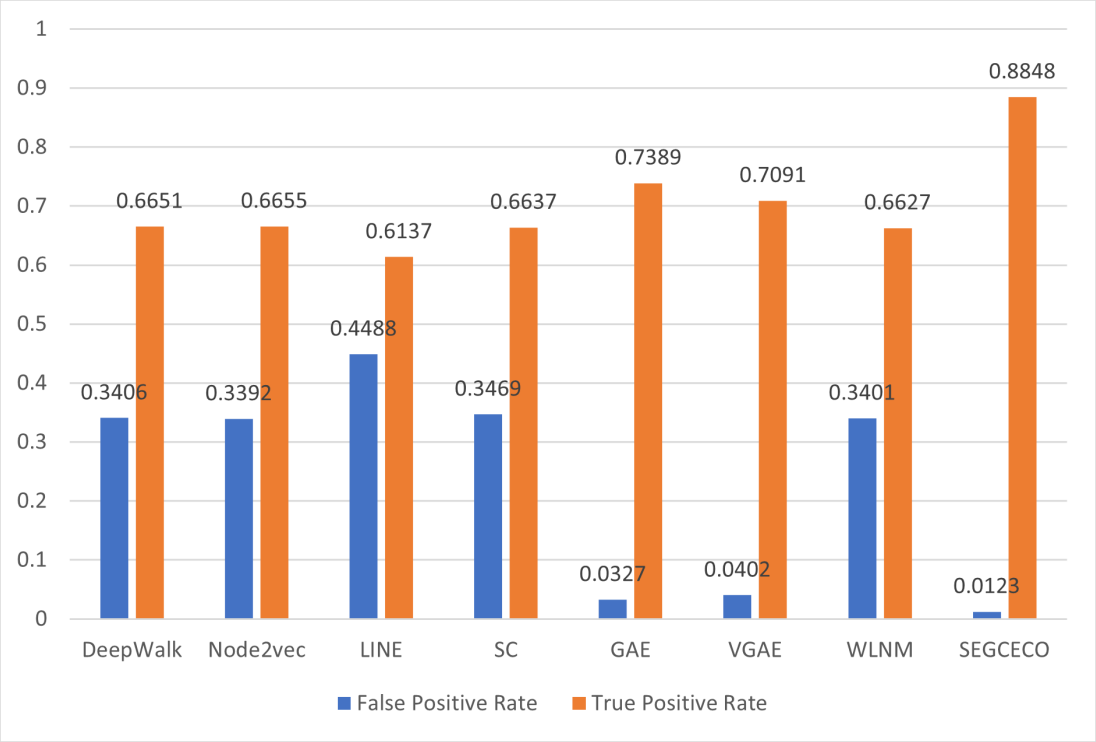

Fig. A.0.2: False Positive Rate and True Positive Rate distribution of BHuman2 dataset

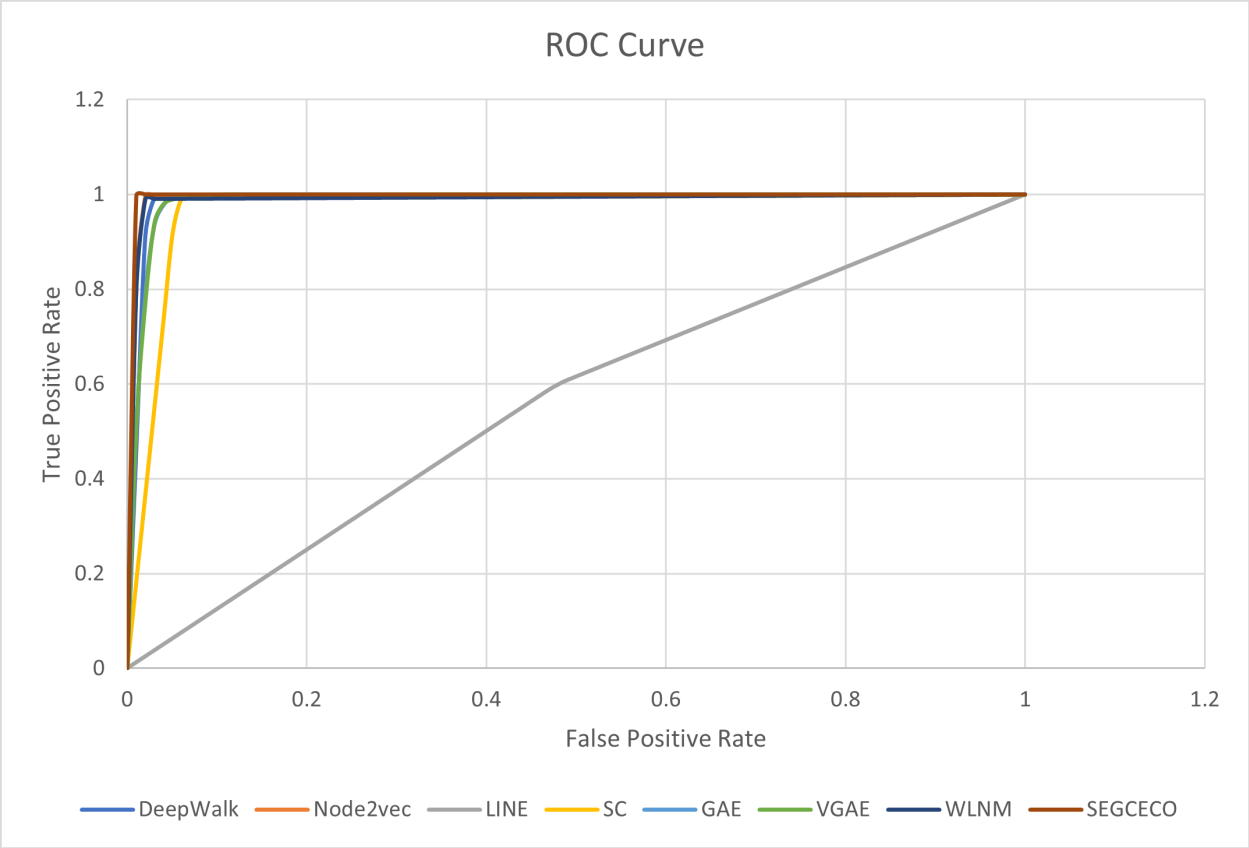

Fig. A.0.3: ROC Curve for BHuman3 dataset

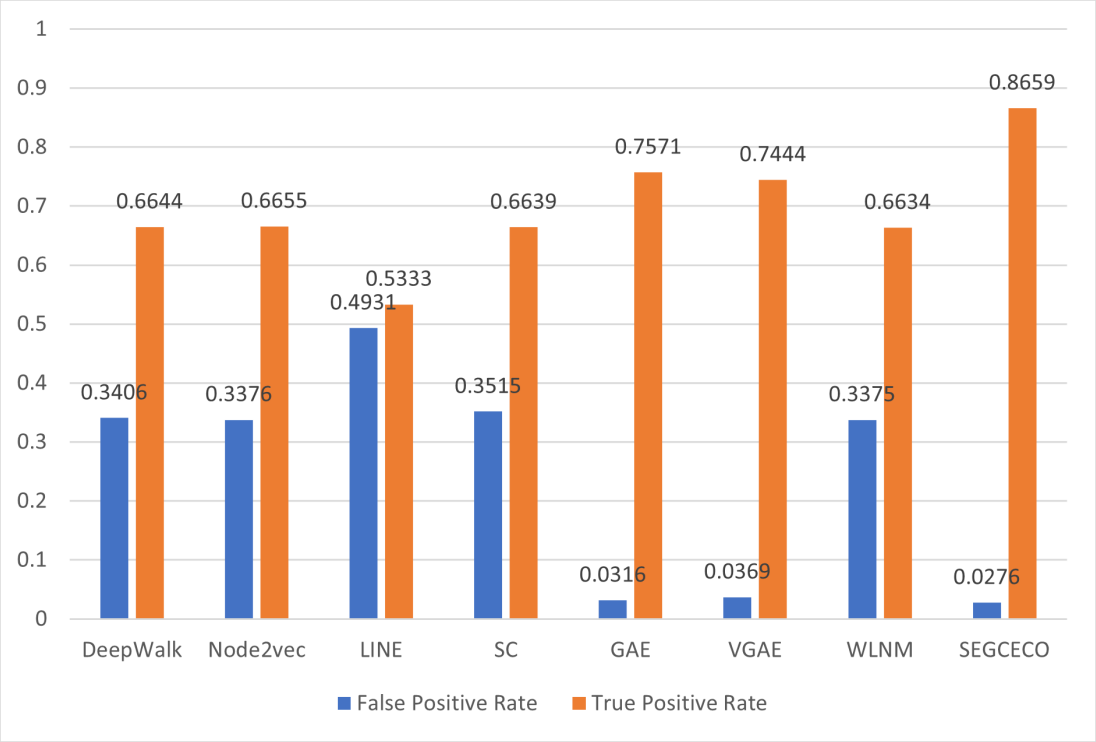

Fig. A.0.4: False Positive Rate and True Positive Rate distribution of BHuman3 dataset

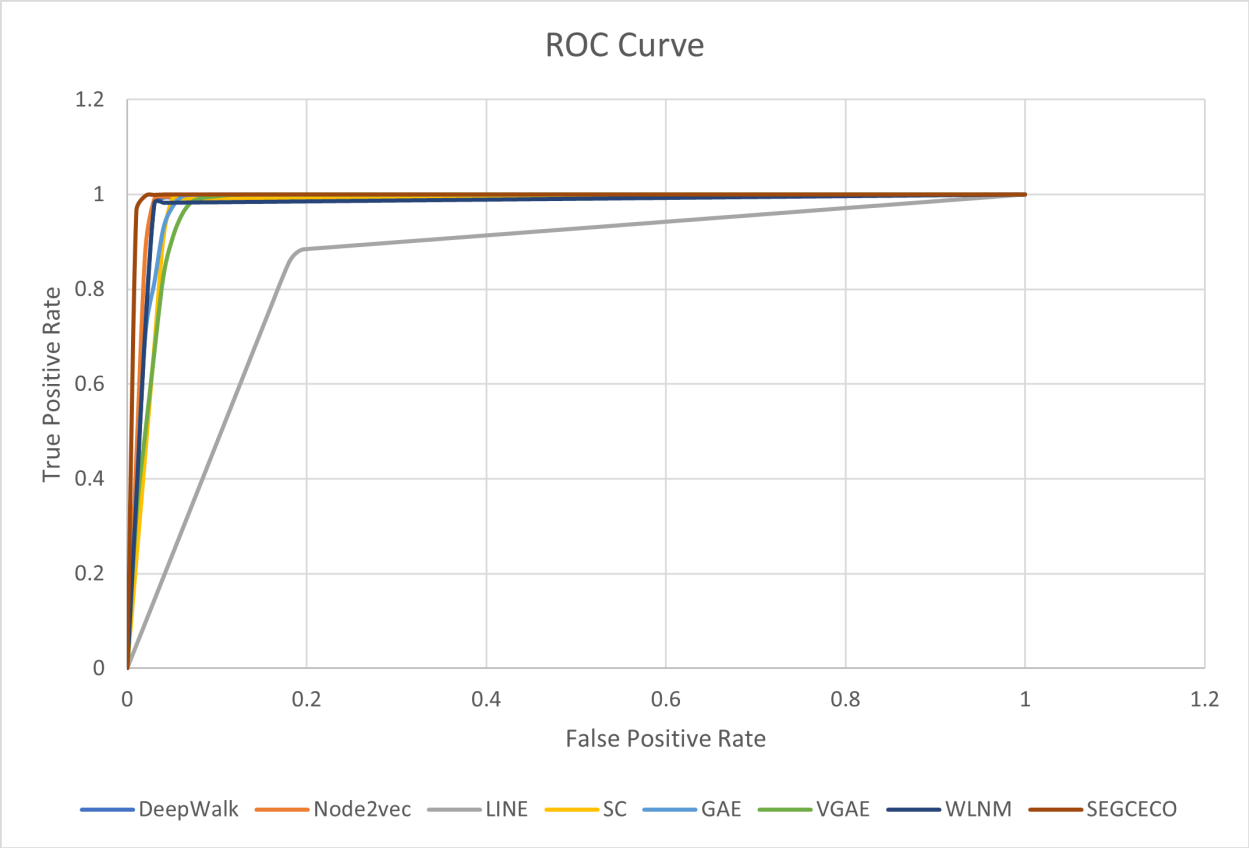

Fig. A.0.5: ROC Curve for BHuman4 dataset

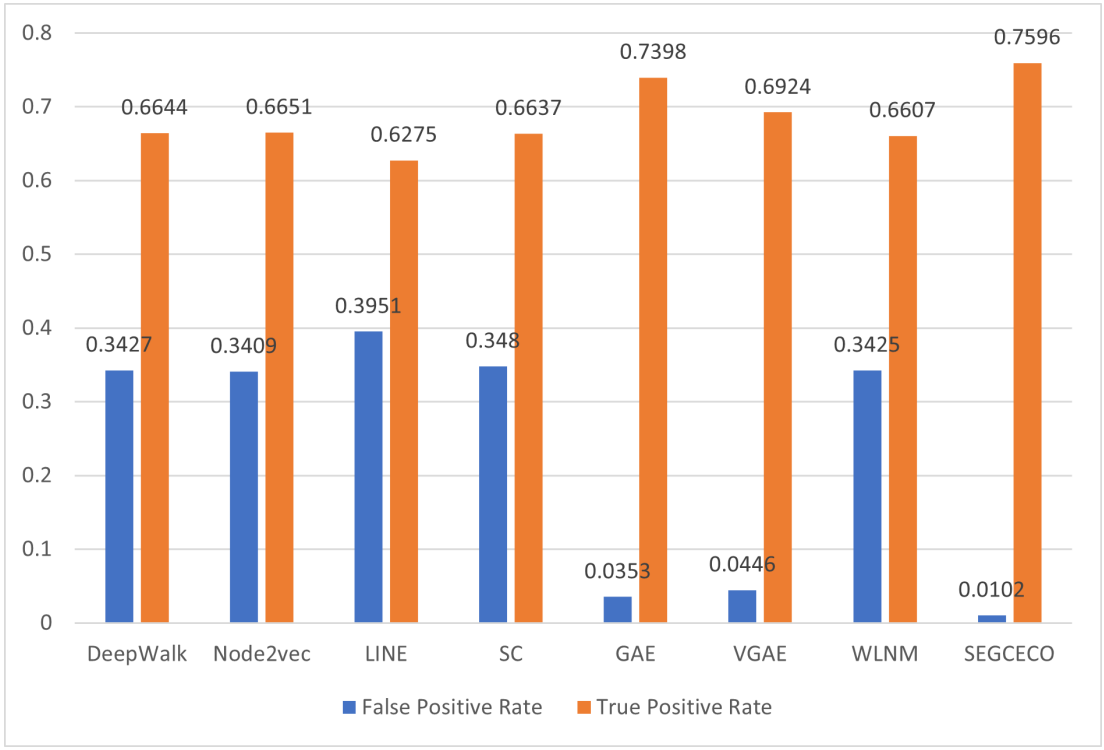

Fig. A.0.6: False Positive Rate and True Positive Rate distribution of BHuman4 dataset

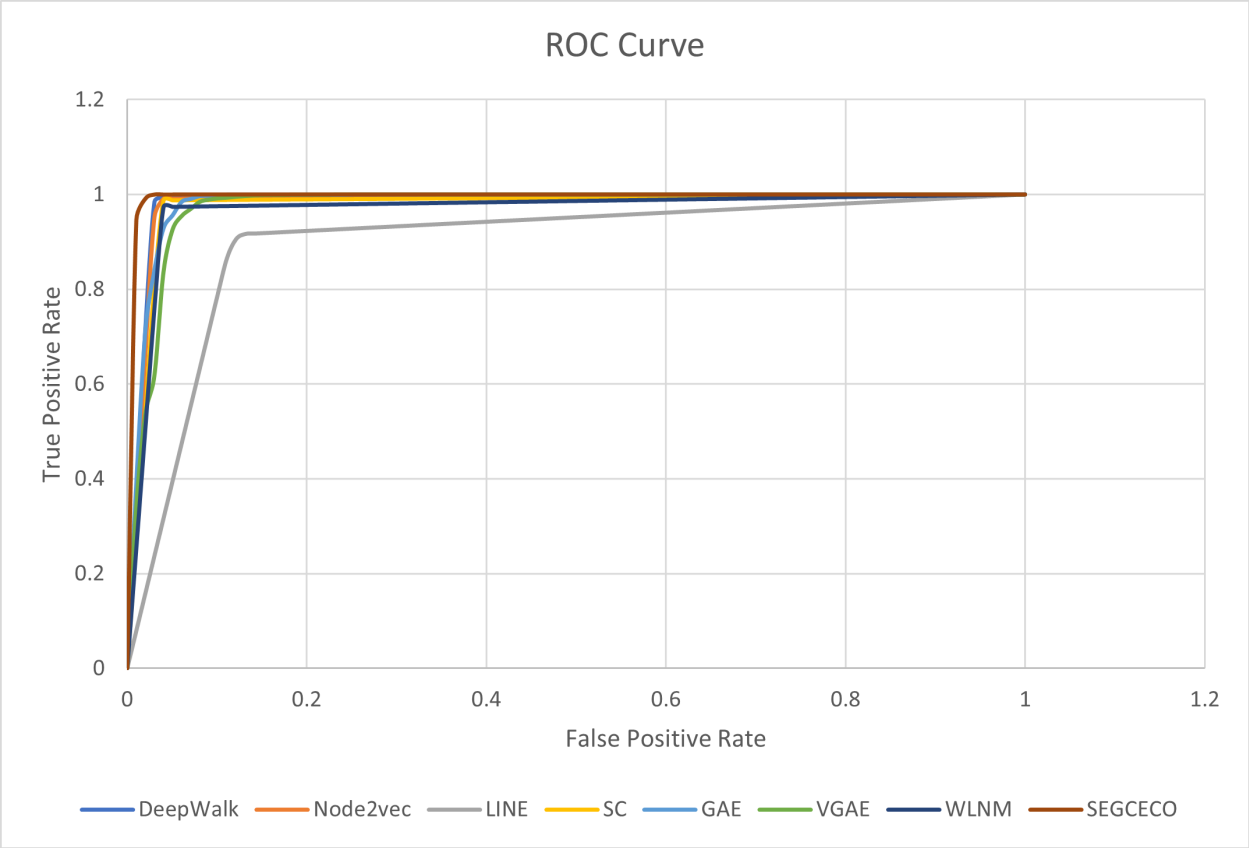

Fig. A.0.7: ROC Curve for BMouse1 dataset

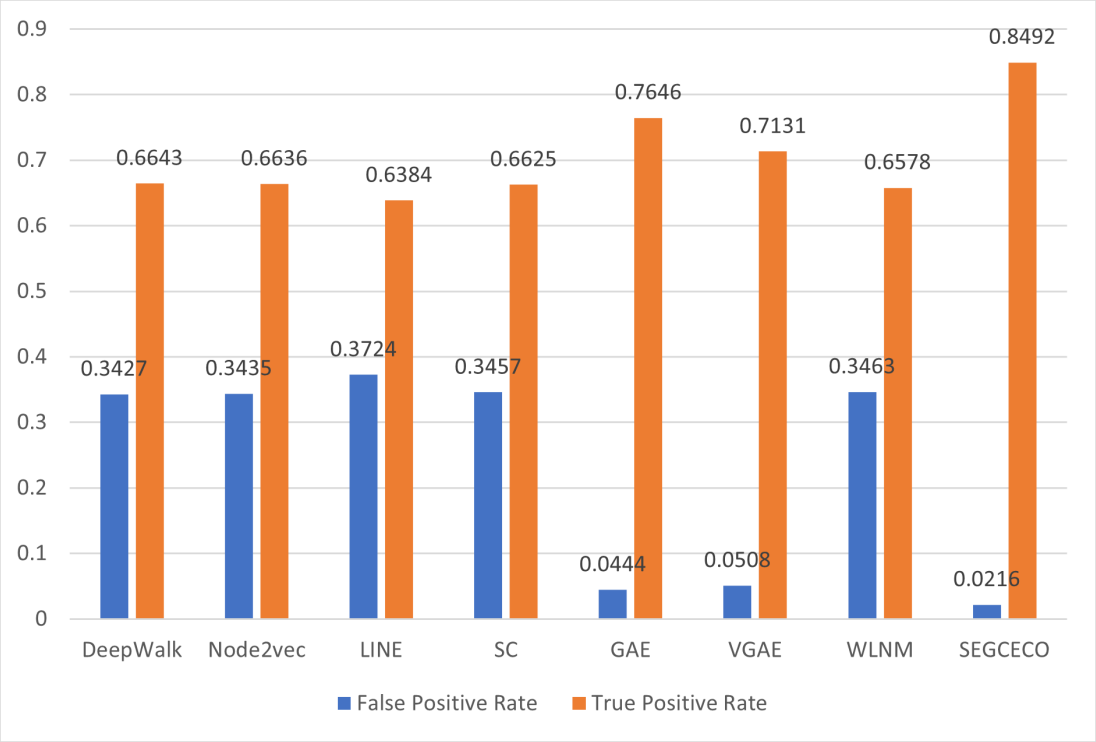

Fig. A.0.8: False Positive Rate and True Positive Rate distribution of BMouse1 dataset

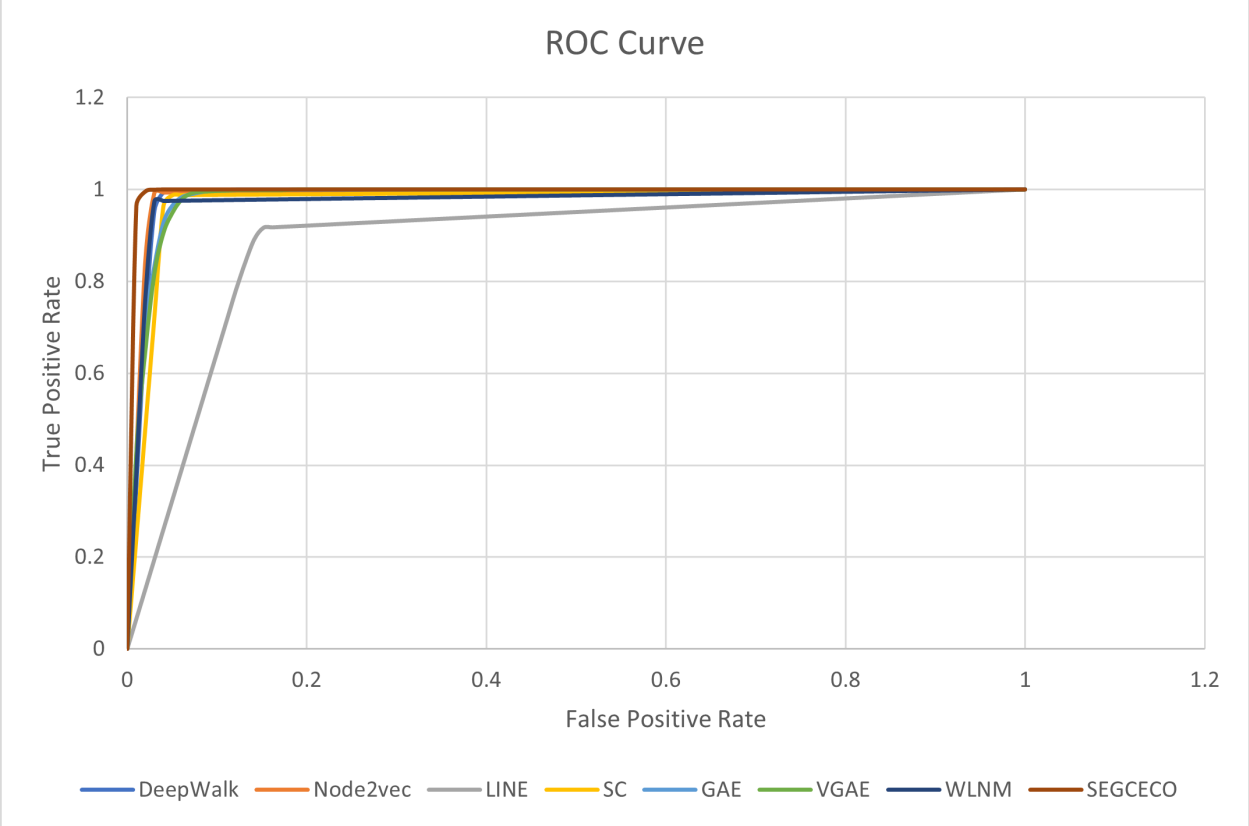

Fig. A.0.9: ROC Curve for BMouse2 dataset

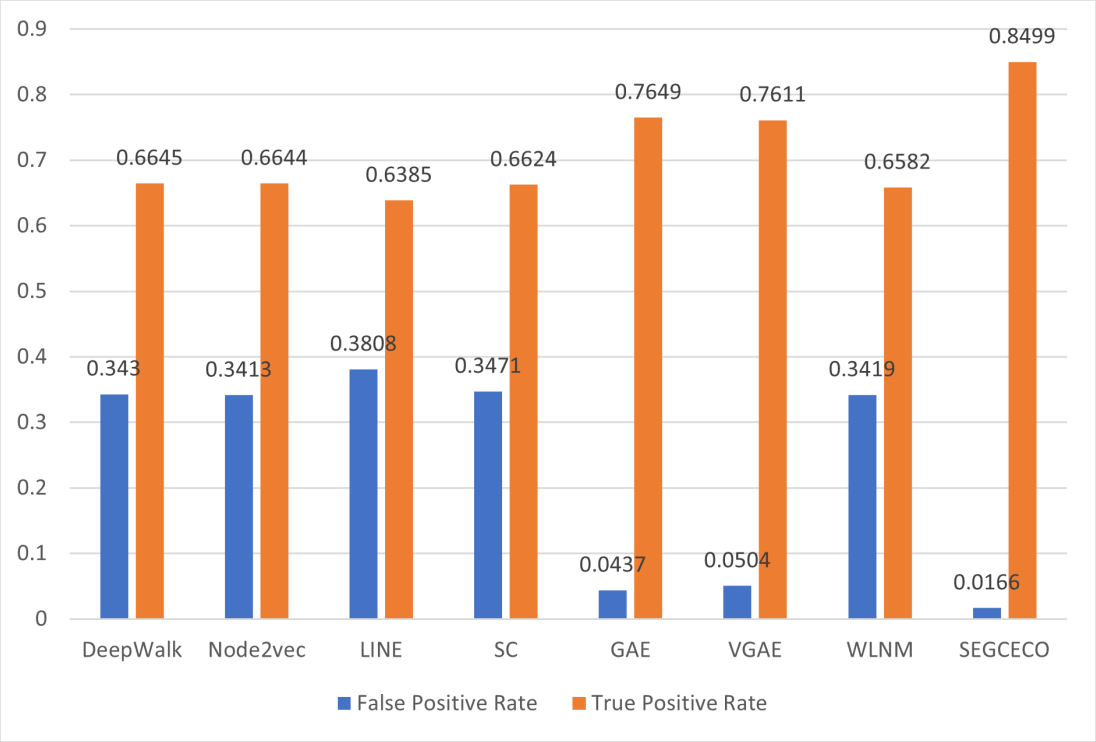

Fig. A.0.10: False Positive Rate and True Positive Rate distribution of BMouse2 dataset

VITA AUCTORIS

NAME:	Sheena Hora
PLACE OF BIRTH:	Chandpur, Uttar Pradesh, India
EDUCATION:	Bachelors in Information Technology, IMS Engineering College, Uttar Pradesh Technical University, Ghaziabad, Uttar Pradesh, India, 2014
	M.Sc. Computer Science, University of Windsor, Windsor, Ontario, Canada, 2020