Εθνικό Μετσόβιο Πολυτεχνείο

Εργαστήριο Βιοϊατρικών Συστημάτων

ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΩΝ ΚΑΤΑΣΚΕΥΩΝ ΚΑΙ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

Διπλωματική Εργασία

Deep learning for signaling network embeddings

Φοιτητής: Αλεβίζος Γεώργιος

Επιβλέπων Καθηγητής: Αλεξόπουλος Λεωνίδας

Αναπληρωτής Καθηγητής ΕΜΠ

ΑΘΗΝΑ, ΟΚΤΩΒΡΙΟΣ 2020

Εθνικό Μετσόβιο Πολυτεχνείο

Εργαστήριο Βιοϊατρικών Συστημάτων

ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΩΝ ΚΑΤΑΣΚΕΥΩΝ ΚΑΙ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

Διπλωματική Εργασία

Deep learning for signaling network embeddings

Φοιτητής: Αλεβίζος Γεώργιος

Επιβλέπων Καθηγητής: Αλεξόπουλος Λεωνίδας

Αναπληρωτής Καθηγητής ΕΜΠ

Εγκρίθηκε την 16η Οκτωβρίου 2020 από την τριμελή επιτροπή:

Αλεξόπουλος Λεωνίδας	Προβατίδης Χριστόφορος	Κυριακόπουλος Κωνσταντίνος
Αναπληρωτής Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ

ΑΘΗΝΑ, ΟΚΤΩΒΡΙΟΣ 2020

Acknowledgments

First of all, I would like to thank my supervising professor, Mr. Leonidas Alexopoulos for giving me the chance to work in such challenging problems during my stay at the lab as well as his guidance throughout the development of the present thesis.

Moreover, I will like to express my utmost gratitude both from Christos Fotis, the PhD student with whom we developed the origin of this thesis, as well as Nikos Meimetis, one of our lab's Resarch Assistants, for their major aid and co-development of the basic aspects behind this diploma thesis.

Also, special thanks to every other member of the lab, Danae, Mari, Nagia, Kostas and Thomas for making this lab an excellent workspace to be involved at.

Lastly and most importantly, I would like to thank my family and close friends for supporting me not only during this diploma thesis, but also throughout my years in the NTUA.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Μηχανολογικών Κατασκευών & Αυτομάτου Ελέγχου
Εργαστήριο Βϊοιατρικών Συστηματων

Βαθεία εκμάθηση για την κωδικοποίηση σηματοδοτικών δικτύων

Διπλωματική Εργασία

Αλεβίζος Γεώργιος

Επιβλέπων: Αλεξόπουλος Λεωνίδας,Αναπληρωτής Καθηγητής ΕΜΠ

Αθήνα, 2020

Περίληψη

Στην Εποχή της Πληροφορίας, τα νευρωνικά δίκτυα και οι μέθοδοι βαθείας εκμάθησης έχουν αναδειχθεί ως ένα από τα πιο επιτυχημένα εργαλεία για την αντιμετώπιση σύνθετων και δύσκολων εγχειρημάτων, τόσο στον χώρο της έρευνας όσο και σε διάφορες εφαρμογές, από την δημιουργία δικτύν που αναγνωρίζουν όγκους σε εικόνες από κύτταρα, μέχρι συστήματα που ξεπερνάνε τα ανθρώπινα όρια σε παιχνίδια όπως το σκάκι. Στον χώρο της Βιοπληροφορικής, οι συνήθεις έρευνες με σκοπό την αναγνώριση του μηχανισμού δράσης φαρμάκων, περιστρέφονται γύρω από την διαχείριση δεδομένων γονιδιακής έκφρασης, με περιορισμένη χρήση μεθόδων βαθείας εκμάθησης. Ιδιαίτερα για τα σηματοδοτικά δίκτυα πρωτεϊνών, χρησιμοποιούνται απλές μέθοδοι ανάλυσης δικτύων ή μοντελοποίηση δυναμικών συστημάτων για την μέχρι τώρα εξαγωγή συμπερασμάτων εξ αυτών. Στην παρούσα διπλωματική εργασία, γίνεται μία προσπάθεια χρήσης μοντέλων νευρωνικών δικτύων γράφων εμπνευσμένων απο την έρευνα στην εκμάθηση γλώσσας, σε δεδομένα διαφόρων σηματοδοτικών δικτύων. Στη προκειμένη περίπτωση, από όσο γνωρίζουμε, είναι η πρώτη φορά που πραγματοποιείται αυτή η προσπάθεια.

Στο πλαίσιο αυτό, επιδεικνύουμε πως τα χρησιμοποιούμενα μοντέλα εκμάθησης, επιτυγχάνουν ικανοποιητική συσταδοποίηση αναπαραστάσεων σηματοδοτικών δικτύων στο χώρο, με σημείο αναφοράς τον μηχανισμό δράσης του φαρμάκου από το οποίο προήλθαν. Παράλληλα, επιτυγχάνεται η περαιτέρω κατανόηση των λειτουργιών των δικτύων με μεθόδους αναγνώρισης των πιό σημαντικών πρωτεϊνών που επιδρούν στην συσταδοποίηση με βάση τον μηχανισμό δράσης.

National Technical University of Athens

School of Mechanical Engineering

Department of Mechanical Design & Automatic Control

Biomedical Systems Laboratory

Deep learning for signaling network embeddings

Diploma Thesis

Alevizos Georgios

Supervisor: Alexopoulos Leonidas, Associate Professor NTUA

Athens, 2020

Abstract

In the era of Big Data, deep learning has emerged as a succesful approach in dealing with complex and challenging problems in every field of research and scientific applications, from playing chess better than a human to recognising malignant cells from cell image data. Classical bioinformatics research mostly relies on gene expression data to provide insight into the mechanism of action of selected drugs, with minimal use of deep learning methods. Signaling networks, on the other hand, are interpreted using network analysis and dynamical systems modelling methods. In the present thesis, we incorporate a specific class of neural networks for graphs, by exploiting architectures stemming from NLP research and apply them into a dataset of biological signaling networks. To the best of our knowledge, this is the first time Graph Neural Networks are combined with signaling pathway data. We demonstrate that our methods can, in an unsupervised way, cluster pathways with similar mechanisms of action together , while simultaneously providing an interpretable framework for identifying the significance of individual proteins in a pathway.

Acronyms

NTUA	National Technical University of Athens
NLP	Natural Language Processing
ANN	Artificial Neural Networks
GNN	Graph Neural Networks
GCN	Graph Convolutional Networks
PPI	Protein-Protein Interaction
GEx	Gene Expression
GO	Gene Ontology
DAG	Directed Acyclic Graph

Contents

Contents i

1 Introduction 1

1.1 Overview of Deep Learning in Research 1

1.2 Motivation 2

2 Theoretical Prerequisites 3

2.1 Biology 3

2.1.1 Systems Biology 3

2.1.2 DNA 3

2.1.3 RNA 4

2.1.4 Gene Expression 5

2.1.5 Proteins 7

2.1.6 Signaling Networks 8

2.2 Machine Learning 9

2.2.1 Neural Networks 9

2.2.2 Graph Convolutional Neural Networks 10

2.2.3 Overview of Transformer Networks 10

3 Data 15

3.1 Preprocessing and Quality Control 15

3.1.1 CMAP 15

3.1.2 TAS Quality 15

3.1.3 CARNIVAL 16

3.2 Graph Features 17

3.2.1 Node Features 17

3.2.2 Edge Features 20

3.3 Mechanisms of Action 20

4 Methods **23**

4.1 The Signaling Network Transformer 23

4.1.1 Positional Encoding Alternatives 23

4.1.2 Activity Embedding 27

4.1.3 Modified Scaled Dot-Product Attention 27

4.2 Unsupervised Learning 29

4.2.1 Additional Models 29

4.2.2 Graph Autoencoders 30

4.2.3 Deep Graph Infomax 32

4.2.4 Unsupervised Embedding Evaluation Tasks 36

4.3 Semi-Supervised Learning 38

4.4 Masked Semi-Supervised Learning 39

4.5 Supervised Learning 40

4.5.1 Additional Models 40

4.6 Node Importance Calculation 41

5 Embedding Quality Evaluation **43**

5.1 Unsupervised Task Evaluation 43

5.1.1 SST Evaluation 44

5.1.2 DST Evaluation 49

5.1.3 MAD Task Evaluation 54

5.2 Semi-Supervised Task Evaluation 57

5.2.1 Semi Supervised Siamese GED 57

5.2.2 Semi Supervised Signaling Network Infomax 58

5.3 Supervised Learning Evaluation 60

6 Case Studies **63**

6.1 Experimental Setup 64

6.2 HDAC Inhibitors 65

6.2.1 PC3 Entinostat 65

6.2.2 HEK 293T Vorinostat 66

6.3 ATP Synthesis Inhibitors 67

7 Conclusion **69**

7.1 Discussion and Limitations 69

7.2 Future Work 70

7.2.1 On improving Deep Learning performance 71

7.2.2 On improving interpretability 71

Bibliography **73**

Chapter 1

Introduction

1.1 Overview of Deep Learning in Research

Over the past decade, **deep learning** has emerged as a versatile and efficient approach for accomplishing complex tasks in multiple fields, such as computer vision [5], Natural Language Processing [9],[42] as well as autonomous driving [20] and computer/board games. As the years go by, neural networks have evolved as the industry standard both when dealing with structured (images, graphs etc.) as well as unstructured data. In accordance with the subject of this thesis, the use of Artificial Neural Networks (ANNs) has hitherto demonstrated tremendous performance in the field of deep learning both in biology and medicine. Examples of such applications make use of MRI data [8], methods regarding cellular image analysis [36] as well as chemical compound based inference [15], [31] aiming on drug discovery.

The aforementioned capability of ANNs to exhibit human-level performance on various datasets and tasks, comes with the heavy price of reduced interpretability ,i.e. the ability to provide meaningful and understandable explanations to human researchers. This reduced interpretability is an extremely important factor to account for when dealing with biological and medicinal cases as it reduces confidence to predictive results coming from ANNs due to the high stakes of many of its applications such as clinical diagnostics.

1.2 Motivation

The present thesis incorporates deep learning techniques and ANNs in order to extract meaningful representations from **signaling networks**. It is not the first time biological networks have been used as data for various research purposes [16], [14] but to the best of our knowledge, it is indeed the first time that Graph Neural Networks(GNNs) applied on this type of data have been used for the purpose of representation learning. We believe that such biological network data have a promising future in the field of computational biology, due to the fact that they can encapsulate a tremendous amount of information both in terms of their connectivity as well as the available features each node includes. Essentialy, each node (protein) of the signaling network can be treated the same a word appears in a sentence. Essentially, if we consider the case of Directed Acyclic Graphs, each single branch of the DAG contains information on the nodes that compose it and thus on the whole graph, immitating the behaviour of sentences in documents. Such practices allow us to treat these types of networks with techniques that are similarly used in the field of Natural Language Processing leveraging both the connectivity and content in various ways. Moreover, the aforementioned issues of interpretability can be reduced due to not only the selected neural network architecture assigned for each task but through the analysis of each nodes importance in each selected scenario.

In the following pages, unsupervised, supervised as well as semi-supervised techniques against various tasks will be studied, in order to prove that signaling network data do indeed provide both adequate and interpetable results.

Chapter 2

Theoretical Prerequisites

2.1 Biology

2.1.1 Systems Biology

Systems biology is an integrative discipline connecting the molecular components within a single biological scale and also among different scales (e.g. cells, tissues and organ systems) to physiological functions and organismal phenotypes through quantitative reasoning, computational models and high-throughput experimental technologies. Systems biology uses a wide range of quantitative experimental and computational methodologies to decode information flow from genes, proteins and other subcellular components of signaling, regulatory and functional pathways to control cell, tissue, organ and organismal level functions.[54]

2.1.2 DNA

DNA, or deoxyribonucleic acid is the central information storage system of most organisms including a portion of viruses. It is a molecule that is composed of two conjugate polynucleotide chains that coil around each other held by hydrogen bonds and form a double helix. The name comes from its structure, which is a sugar and phosphate backbone which have bases sticking out from it. DNA encodes information through the order, or sequence, of the nucleotides along each strand. Each base—A, C, T, or G—can be considered as a letter in a four-letter alphabet that spells out biological messages in the chemical structure of the DNA. The complete set of information in an organism's DNA is called its genome, and it carries the information for all the proteins the organism will ever synthesize. [3]

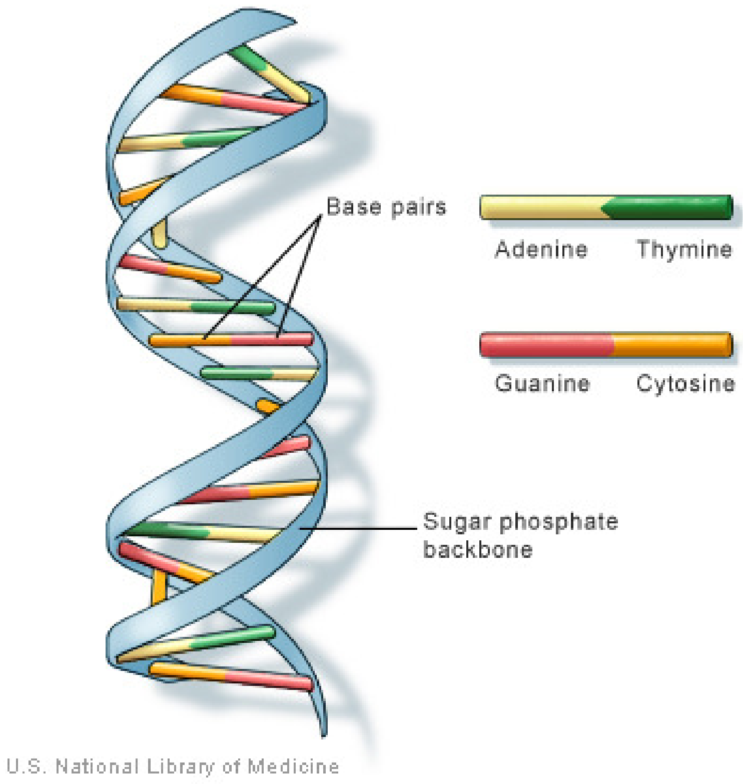

Figure 2.1: *DNA structure, U.S. National Library of Medicine*

2.1.3 RNA

Like DNA, RNA is a linear polymer made of four different types of nucleotide subunits linked together by phosphodiester bonds. It differs from DNA chemically in two respects:

1. The nucleotides in RNA are ribonucleotides—that is, they contain the sugar ribose (hence the name ribonucleic acid) rather than deoxyribose;
2. Although, like DNA, RNA contains the bases adenine (A), guanine (G), and cytosine (C), it contains the base uracil (U) instead of the thymine (T) in DNA since U, like T, can base-pair by hydrogen-bonding with A.

Apart from the small chemical differences, DNA and RNA have a tremendous difference in overall structure.Whereas DNA always occurs in cells as a double-stranded helix, RNA is single-stranded. RNA chains therefore fold up into a variety of shapes, just as a polypeptide chain folds up to form the final shape of a protein. [3]

2.1.4 Gene Expression

In general, with the term **gene expression**(GEx) we refer to the natural process in which the aforementioned information that is stored inside the DNA is converted into functional products like proteins or different types of RNA. The process of gene expression is being deployed by two basic operations, **transcription** and **translation**.

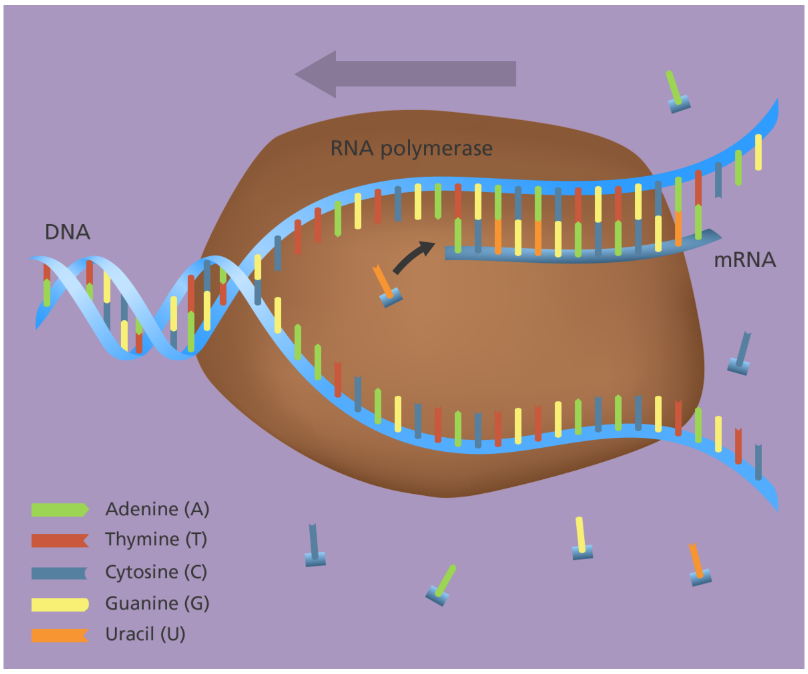

Figure 2.2: *The process of transcription (Image Credit: Genome Research Limited)*

The first step a cell takes in reading out a needed part of its genetic instructions is to copy a particular portion of its DNA nucleotide sequence—a gene—into an RNA nucleotide sequence. The information in RNA, although copied into another chemical form, is still written in essentially the same language as it is in DNA—the language of a nucleotide sequence. Hence the name transcription. RNA in the cell is completely created by DNA transcription, which begins by opening and unwinding of a small portion of the DNA double helix to expose the bases on each DNA strand. One of the two strands of the DNA double helix then acts as a template for the synthesis of an RNA molecule.The enzymes that perform transcription are called RNA polymerases and the transcript is called messenger RNA (mRNA).

The second process, translation, occurs when the aforementioned messenger RNA has carried the transcribed the needed information from the DNA to the cells' ribosomes, in which proteins are being created. The translation of mRNA into protein depends on adaptor molecules that can recognize and bind both to the codon(three letters) and, at another site on their surface, to the amino acid. These adaptors consist of a set of small RNA molecules known as transfer RNAs (tRNAs), each about 80 nucleotides in length.Once the tRNA is bound, it releases its amino acid and the adjacent amino acids all join together into a long chain called a polypeptid, continuing the process above until the protein is formed.[3]

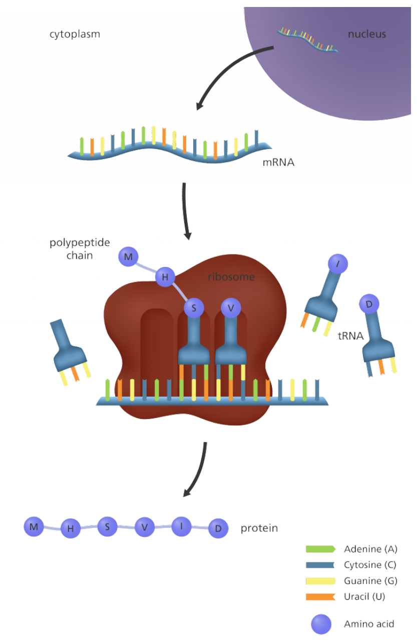

Figure 2.3: *The process of transcription (Image Credit: Genome Research Limited)*

2.1.5 Proteins

Proteins are by far one of the most chemically complex and functionaly sophisticated molecules known so far. A protein molecule is made from a long chain of amino acids (20 different types of amino acids) each linked to its neighbor through a covalent peptide bond, thus the alternate name *polypeptides*. Each type of protein has a unique sequence of amino acids.

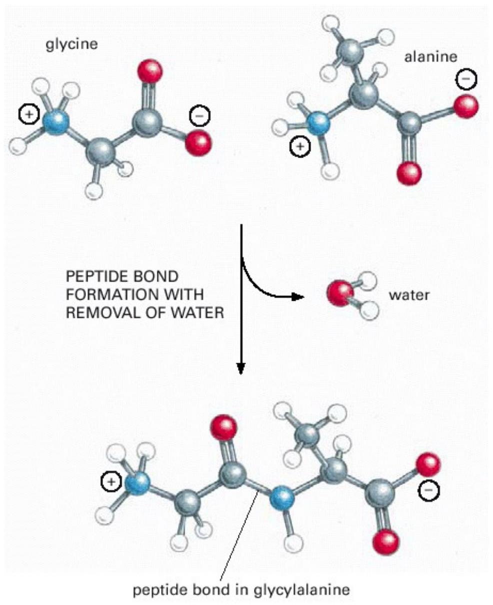
glycine
alanine
PEPTIDE BOND FORMATION WITH REMOVAL OF WATER
water
peptide bond in glycylalanine

Figure 2.4: *Peptide bond [3]*

The biological properties of proteins depend entirely on their physical interaction with other molecules. For example, antibodies, Y shaped proteins that are produced by the immmune system, bind to viruses or bacteria, actin molecules bind to each other to assemble into actin filaments, and so on. The substance that is bound by the protein is called a **ligand**. The region of a protein that associates with a ligand, known as the ligand's binding site, usually consists of a cavity in the protein surface formed by a particular arrangement of amino acids.[3]

2.1.6 Signaling Networks

One of the most important issues in biology is the study of the various interactions between cellular molecules which determine their biological properties. Such interaction networks are usually classified according to the type of the molecules involved, these usuall being genes or proteins. Networks that involve cell signaling, i.e. the response of a cell to internal and external stimuli (chemical or even of mechanical and elecrical nature) and coordinate the regulation of its activity are called **Signaling Networks**.

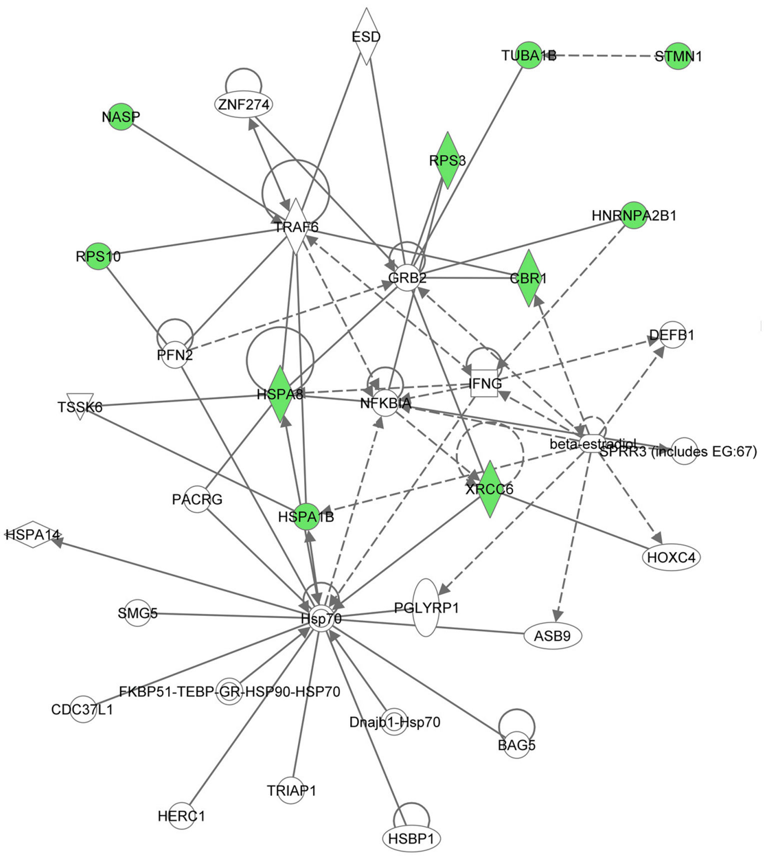

Figure 2.5: *An example signaling network [1]*

Individual pathways transmit signals along linear tracts resulting in regulation of discrete cell functions. This type of information transfer is an important part of the cellular repertoire of regulatory mechanisms.Inside the cell, there exists a particular family of proteins called *receptors* that bind to signaling molecules and initiate an initial response.Such signaling molecules include **Hormones** which are the major signaling molecules of the endocrine system, **Neurotransmitters**, which are signaling molecules of the nervous system and **Cytokines** which are signaling molecules of the immune system. Essentially, carrying out complex biological processes requires the cooperation of several cells along with their specific functions, by the process of cell signaling.[26],[25],[13]

2.2 Machine Learning

Since the mathematical theory behind deep learning and neural networks is beyond the scope of this thesis, the following subsections include a brief overview of neural networks as well as the selected architectures that correspond to the methods chosen to be presented in this thesis.

2.2.1 Neural Networks

Traditional machine learning techniques such as Linear Regression, Support Vector Machines (SVMs) as well as Random Forests have proven extremely useful to the Machine Leaning and Computer Science community for decades. But during the times of the so called Fourth Industrial Revolution, petabytes of data generated every day created the need of scalable models that can match both the tremendous amount as well as the increased structural complexity of the incoming data. Thus, neural networks have been widely employed over the past few years in order to develop models that address the aforementioned issues.

Deep Learning is a subfield of Artificial Intelligence which utilizes ANNs. Inspired by biological neural networks, the basic building block of ANNs are **neurons** which are essentialy a form of mathematical entity that holds a real number. Each neuron accepts the value of previous connected neurons as input, and maps into a non-linear function, also called an activation function: $x_{new} = s(w * x_{prev} + b)$ where w, b are the trainable parameters of the node called weight and bias and s is the non-linear activation function. Neurons, in their turn form **layers** a series of whom describes the basic architecture of a Feedforward Neural Network.

Neural networks are generally trained by optimizing a selected cost function that most adequately describes the task at hand. These cost functions accept the networks output as well as the ground truth labels as input and are trained using optimization algorithms that revolve around Stochastic Gradient Descend (SGD).[19]

2.2.2 Graph Convolutional Neural Networks

Originally introduced by Kipf and Welling [27] Graph Convolutional Networks (GCNs) have become the starting point when working with graphs using neural networks. In their simplest form, GCNs operate on undirected graphs $\mathcal{G} = (\mathcal{V},\mathcal{E})$ where $\mathcal{G}$ is the graph and $\mathcal{V}$, $\mathcal{E}$ describe the set of vertices and edges of the graph respectively. A simple propagation rule would be

$$f(H^{(l)}, A) = \sigma\left(AH^{(l)}W^{(l)}\right) , \tag{2.1}$$

when $W^{(l)}$ is the weight matrix on layer i, A is the adjacency matrix and $H^{(l)}$ is the feature matrix of the graph nodes in layer i. Therefore, this process imitates the way typical Convolutional Neural Networks work, in the sense that a filter propagates around the graph, reading both the information of each node as well as aggregating the features of their neighbourhood. There are countless variations of the original GCNs, few of whom will be addressed later in the thesis, as many of them were used extensively during our first trials with the type of networks this thesis deals with.

2.2.3 Overview of Transformer Networks

Created as an alternative to complex recurrent or convolutional neural networks, transformers (Vaswani et. al) follow a much simpler architecture but exhibit more powerful representations. The main structure makes use of the typical encoder-decoder architecture, binded together with an attention mechanism. It is important to explain the vanilla transformer to the reader, since the basic models used in this thesis belong to this family of neural network architectures. We will be focusing on the encoder part of the transformer and especially the attention mechanisms as the modified variations we employ are inspired by them.

The encoder is composed of a stack of N identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise, fully connected feed-forward network. Each sublayer is also connected using residual connections [21].

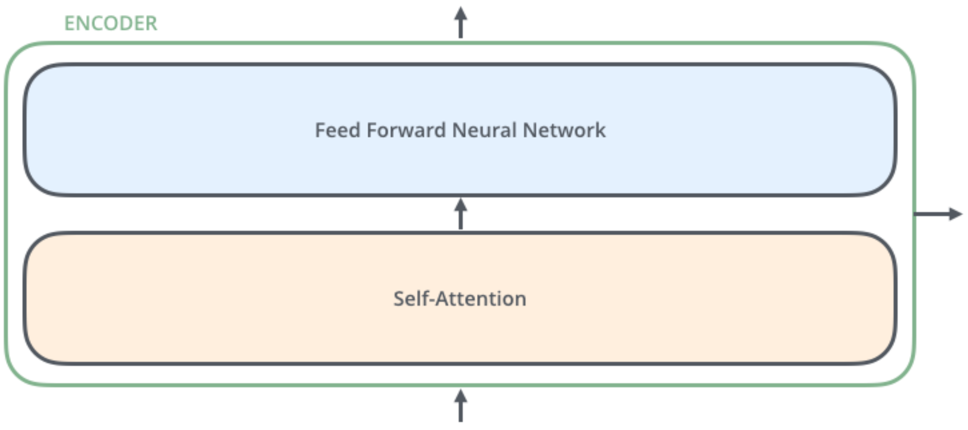

Figure 2.6: *Self-Attention Feed Forward Architecture*

In general, attention functions can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

Scaled Dot-Product Attention

The attention mechanism between the queries, keys and values in the original paper is called scaled dot-product attention. In practice, for each graph we have the three

$\mathcal{Q}$, $\mathcal{K}$, $\mathcal{V}$ matrices that correspond to the total query, key and value vectors of the graph's node features. The attention is calculated as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}}) \tag{2.2}$$

where $\frac{1}{\sqrt{d_K}}$ is a scaling factor. Thus, nodes with features that are similar (due to their dot products) will be binded with a higher attention score between them than dissimilar nodes.

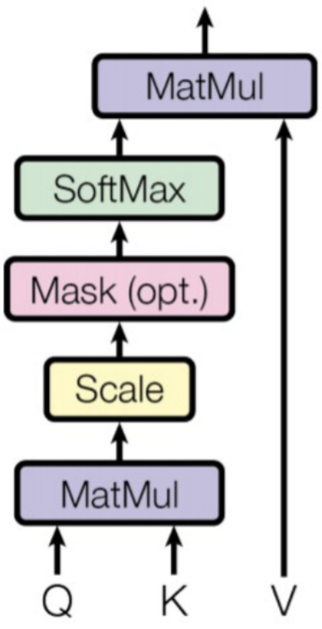

Figure 2.7: *Scaled Dot-Product Attention [58]*

Multi-Head Attention

Instead of single dot-product attention on the whole feature vectors q, k, v it was found that linearly projecting those vectors h times with different, learned linear projection was more beneficial. For example, if the dimensionality of the graph features d_{model} was 1024 and we selected 4 heads, then we would end up with 4 triplets of Q, K, V matrices with 256 dimensions, along with their corresponding weight matrices. In its general form, multi-head attention is described as:

$$MultiHead(Q, K, V) = Concat(head_1, head_2, ..., head_h)W^0 \tag{2.3}$$

with each head corresponding to the previously calculated dot-product attention.

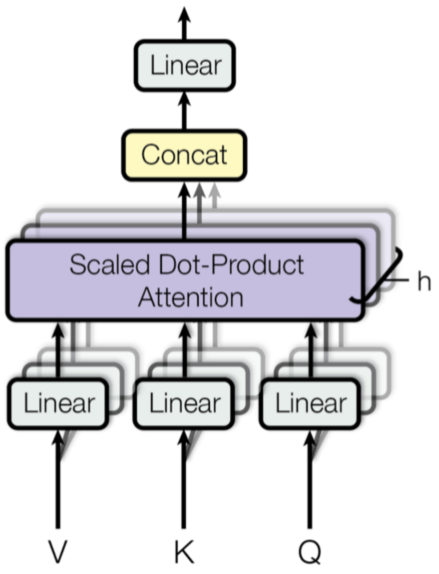

Figure 2.8: *Multi-Head Attention [58]*

Vanilla Encoder Architecture

The previously mentioned attentions mechanisms are the two most important parts concerning the architecture used in this thesis. Since it was originally designed to deal with NLP tasks, the modifed Graph Transformer that we employ differs with the vanilla transfomer in terms of attentional mechanisms but not in the general encoder architecture. Thus, we end up with Figure 2.9 which, in this thesis, will be considered the universal architecture of the transformer encoder.

As far as Positional Encoding is concerned, it is out of the scope of this thesis to thoroughly explain the math behind the original version. In general, it refers to a vector added to the initial node features that poinpoints each words location in the sentence. This positional encoding vector has a sinusoidal form, and the reader is encouraged to briefly study it in the original paper. Our modified positional encoding will be introduces in the **Methods** section of this thesis.[58]

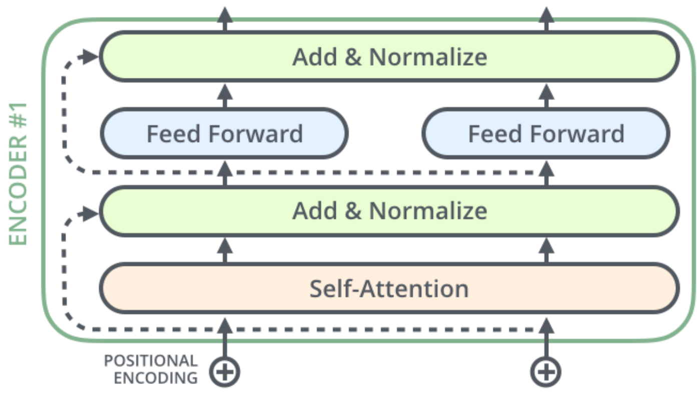

Figure 2.9: *Transfomer Encoder [58]*

Chapter 3

Data

Even with the best neural network architectures and methods available, it is of paramount importance that the data selected to accomplish the given task are of the best available quality in order to attain robust predictive results. The ways of gathering and preprocessing said data will be presented in the next sections of the thesis. Some of this data is available at the NTUA's System Biology Lab Github page [1], but due to the restrictive size of others such as the signaling network graphs, the option of uploading them to our repository was not available.

3.1 Preprocessing and Quality Control

3.1.1 CMAP

CMAP or the Connectivity Map project by the Broad Institute LINCS Center for Transcriptomics, provided us with the transcriptomic signatures needed to develop the appropriate signaling networks. The version of CMAP that was used was the GSE92742, with a level 5 transformed z-score. Note that only the differential expression of the 978 landmark genes in the L1000 assay was considered.[51]

3.1.2 TAS Quality

Quantifying the quality of the given data was based on its Transcriptional Activity

[1] https://github.com/BioSysLab

Score (TAS)[51]. TAS is computed as the geometric mean of the Signature Strength (SS, the number of differentially expressed genes within a signature with absolute z-score greater than 2) and the Replicate Correlation (CC, 75th quantile of the spearman correlations between all pairwise combinations of replicate level 4 profiles on a given experiment) for a signature, scaled by the square root of the number of landmark genes. TAS ranges from 0 to 1 and the quality score ranges from 1 to 8, with the category of quality 1, which corresponds to a transcriptomic activity score greated than 0.4 and more than 2 replicates, containing the best quality signatures. For the selected experiments, only the 7788 available signatures of Quality Score 1 were selected to ensure the validity of our results.

3.1.3 CARNIVAL

CARNIVAL (CAusal Reasoning pipeline for Network identification using Integer VALue programming)[32] is a causal network contextualization tool, that identifies upstream regulatory signaling pathways by using downstream gene expression data. It integrates various sources of prior knowledge, like signed and directed protein-protein interaction networks [55][25], transcription factor targets, as well as pathway signatures.

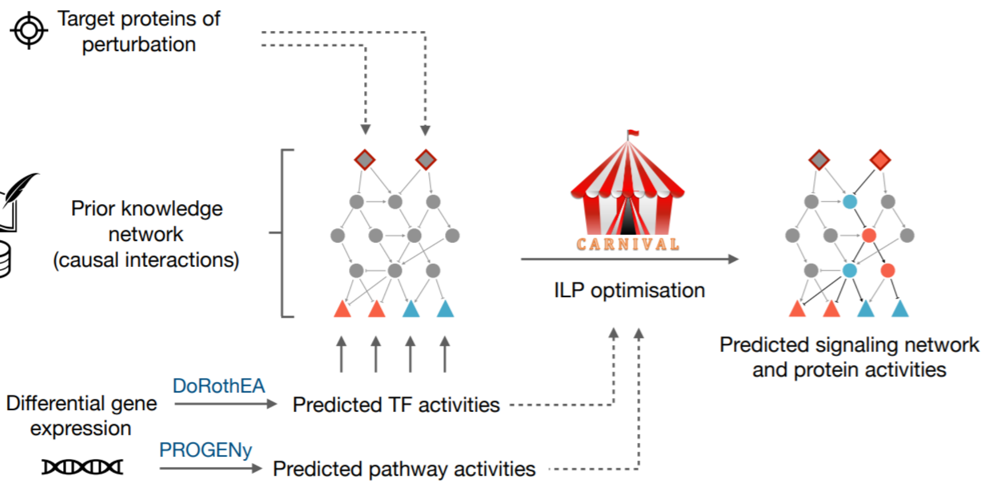
Target proteins of perturbation
Prior knowledge network (causal interactions)
CARNIVAL
ILP optimisation
Predicted signaling network and protein activities
Differential gene expression
DoRothEA
Predicted TF activities
PROGENy
Predicted pathway activities

Figure 3.1: *CARNIVAL Pipeline [32]*

The aforementioned Quality 1 data were processed with CARNIVAL, along with the following recources:

- Transcription factor activities from DoRothEA[18], a gene set resource containing signed transcription factor (TF) - target interactions
- Signed and directed Protein-Protein Interaction (PPI) networks from OmniPath[57]
- Pathway scores out of gene expressions from PROGENy[47]
- An Integer Linear Programming (ILP) solver from which the optimal predicted signaling network topology arises, specifically IBM ILOG CPLEX. The ILP problems are linearly constrainted from both the TF's activities as well as the PPI.

The end result, after using CARNIVAL, were 7788 weighted, signed and directed signaling networks, as well as their corresponding unweighted networks per signature, a number which varies from 5 to 100 per weighted signaling network. The weighted networks are produced by adding the unweighted ones, thus edge weights describe the percentage of times a certain edge appeared in the unweighted graphs.

3.2 Graph Features

After processing the initial GEx data with CARNIVAL, we end up with the previously mentioned 7788 graphs corresponding to the Quality 1 signatures. In order to feed those graphs to the models described in the Methods section, node and edge features need to be properly established so that their mathematical equivalents provide useful representations to be fed in neural networks.

3.2.1 Node Features

Each node of the graphs that this thesis is focused on, is an individual protein of a cell signaling network. Thus, when each graph is being processed, each node's features need to have a multi-dimensional distributed representation that mathematically describes the the proteins' various modes of action and their biological significance i.e. their place in the biological map. In the following subsections, two ways of dealing with this issue will be briefly presented; one concerning protein embeddings that arise from their Gene Ontology (GO) terms and the other of embeddings that are concerned with the amino acid sequence of each individual protein.

CorEx

The node features described in the following subsections, due to the complexity of their components, exhibit a dimensionality that exceeds 2000 features. From a certain point of view, this huge dimensionality might seem that is more than adequate to explain the biological features of each protein in the graph. But while this is partially true, the computational complexity that follows this high dimensionality will increase the training of each neural network almost exponentially. This problem which falls under the Curse of Dimensionality can be solved by employing methods of dimensionality reduction.

Thus, we make use of CorEx[49][50], a method used to discover structure in high dimensional data using Correlation Explanation, hence the name. This particular unsupervised algorithm searches for a predefined number of latent factors that can best explain the correlation between the original high dimensional data, using multivariate mutual information. An accurate mathematical explanation of the Correlation Explanation method is thoroughly analyzed in the citations provided in this subsection.

GO Term Features

The GO knowledge base [4] [56] is the world's largest information database on the function of genes. In our case, it provides a set of hierachically controlled vocabulary which is divided in three distinct functional categories:

Biological Processes Also called biological programs, they are the largest processes accomplished by multiple molecular activities. Examples include **DNA**

repair or **signal transduction**, although not referring to a pathway process.

Molecular Function Molecular-level activities performed by gene products. Molecular function terms describe activities that occur at the molecular level, such as "catalysis" or "transport". GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the actions, and do not specify where, when, or in what context the action takes place. Molecular functions generally correspond to activities that can be performed by individual gene products (i.e. a protein or RNA), but some activities are performed by molecular complexes composed of multiple gene products.

Cellular Component The locations relative to cellular structures in which a gene product performs a function, either cellular compartments (e.g., mitochondrion), or stable macromolecular complexes of which they are parts (e.g., the ribosome). Unlike the other aspects of GO, cellular component classes refer not to processes but rather a cellular anatomy.

The tool used to insert the aforementioned GO terms into our dataset was topGO [2], an R package for testing GO terms while accounting for the topology of the Gene Ontology graph.

Amino Acid Sequence Features

Apart from the Gene Ontology terms, another way is to encode the aspects of protein function and structure based on each individual proteins amino acid sequence. A novel way to represent protein sequences as continuous vectors (distributed representations) is presented in [22]. SeqVeq(Sequence-to-Vector)[2] uses a bi-directional model inspired from NLP tasks called ELMo [44] to capture the biophysical properties of sequences from big unlabelled data, specifically the UniProt50 database.

This method has been proved rather effective in terms of predictive results in various tasks, just by using protein sequence data, outperforming even some methods using evolutionary information.

[2] https://github.com/rostlab/SeqVec

3.2.2 Edge Features

In every graph, the edges represent the type of connection two nodes share. In our case, the connection between two neighboring nodes (in a directed fashion) depict three different relational entities:

Protein Interaction In the cell signaling network, which is computationally formulated as a Directed Acyclic Graph(DAG), proteins either upregulate or downregulate the next protein in each branch of the DAG, originally being represented as a 1 or a -1 respectively. Since this formulation cannot be directly understood by neural networks, it was changed with one hot vectors, thus changing 1 to [1 0] and -1 to [0 1], enforcing a categorical attribute to this connection.

Edge Weight In the case of **weighted** graphs, it quantifies the appearance of the edge in each of the unweighted graphs from which the original was produces. Ranges from 0 to 1, where 1 means this edge appeared in every unweighted graph.

PPR Weight In the **Methods** section, Personalised Page Rank is presented as a method to enforce a relative positioning feature of each protein inside the graph. This attribute is represented as a single number ranging from 0 to 1 and signifies the ease of getting from protein A to protein B and is directly linked to the aforementioned edge weight.

3.3 Mechanisms of Action

Each signature ID corresponds to GEx data, after a drug has been administered to a cell line, where the corresponding cell signaling network exhibits the cells' response to each drug. For much less than half of the experiments, specifically 2733 signatures, we were able to acquire the administered drugs' possible mechanism of action(s). Since the original labels were not consistent, we had to group similar labels together, e.g. grouping all DNA or kinase inhibitors, or arbitrarily select one of the available mechanisms of action for each drug and if possible, grouping that as well. By following this procedure, we end up with 255 unique mechanism of action labels.

It is very important to take into account the fact that each one of those labels is not definitive and unique for each drug. More that one labels may correspond to one specific drug, and this heavily undermines the training evaluation metrics. The goal of those labels is to provide us with a baseline evaluation procedure, in order to focus our attention in interpeting each signaling network pathways in terms of an attributed mechanism of action.

Chapter 4

Methods

The ultimate goal of this thesis is to provide a series of supervised and unsupervised methods so that the potential significance of cell signaling networks is exhibited. In the following sections, we will describe the architecture of our implementation of the Transformer, as well as the methods that will be used along the way. The code for each model and implementation can be found at our repository at https://github.com/BioSysLab/deepSNEM.

4.1 The Signaling Network Transformer

In the **Machine Leaning** section of Chapter 2, a brief overview of the original Transformer[58] was presented. In order to process the cell sigaling network graphs using a neural network, instead of the standard way of employing a Convolutional Graph Neural Network, we chose to implement a novel, modified version of the Transformer for graphs. There are three different functional differences between our model and the original version, which will be presented in the following subsections.

4.1.1 Positional Encoding Alternatives

As mentioned in the theoretical section of this thesis, the original Transformer used a method called Positional Encoding to enforce a sense of placement of each word inside the sentence. In similar fashion, in order for each protein to have an edge feature that depicts its relative position with child and parent proteins in the DAG, we make use of the Personalised PageRank algorithm[41], a graph diffusion method based on the PageRank algorithm by former Google CEO Larry Page and others. Another alternative is the use of the Floyd-Warshal algorithm, a much more computationally feasible alternative with similar results.

PPR Algorithm

For the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with node set $\mathcal{V}$ and edge set $\mathcal{E}$ we denote $\mathcal{N} = |\mathcal{V}|$ the number of nodes and $\mathcal{A}$ its adjacency matrix. The diffusion matrix is then

$$\mathcal{S} = \sum_{k=0}^{\infty} \theta_k^{PPR} T_{rw} \tag{4.1}$$

with the weighting coefficients $\theta_k^{PPR} = \alpha(1-\alpha)^k$ and the random walk transition matrix $T_{rw} = AD^{-1}$ with $\mathcal{D}$ being the degree matrix. Those values represent the PageRank algorithm and are selected so that Eq. 4.1 converges.

Essentially, graph diffusion exchanges the normal adjacency matrix A with a sparsified version of the generalized graph diffusion matrix S. This matrix defines a weighted and directed graph, and the model we aim to augment is applied to this graph instead. The sparsified matrix $\widetilde{\mathcal{S}}$ is used in the modified scaled dot-product attention described in the following sections.

Floyd-Warshall Algorithm

The Floyd-Warshall algorithm is an algorithm used to calculate shortest path distances in a weighted graph with positive or negative edges, by comparing all possible paths through the graph between each pair of vertices. Let $dist(k,i,j)$ be be the length of the shortest path from i to j that uses only the vertices $u_1, u_2, ..., u_k$ as intermediate vertices. Then:

- k=0 is our base case as $dist(0, i, j)$ is the length of each vertex i to vertex j if it exists, and its ∞ otherwise.
- $dist(k, i, j) = min(dist(k-1, i, k) + dist(k-1, j, k), dist(k-1, i, j))$

The pseudocode of the Floyd-Warshall algorithm is described bellow:

Algorithm 1 Floyd Warshall Shortest Path Distance Matrix

```
for $i \leftarrow 1$ to $N$ do
    for $j \leftarrow 1$ to $N$ do
        if there exists and edge from $i$ to $j$ then
            $dist[0][i][j] \leftarrow$ length of edge from i to j
        else
            $dist[0][i][j] \leftarrow \infty$
        end if
    end for
end for
for $k \leftarrow 1$ to $N$ do
    for $i \leftarrow 1$ to $N$ do
        for $j \leftarrow 1$ to $N$ do
            $dist[k][i][j] \leftarrow min(dist[k-1][i][j], dist[k-1][i][k] + dist[k-1][k][j])$
        end for
    end for
end for
```

Then $dist[N][i][j]$ describes the shortest path distance between node i to node j.

Relative Positional Encoding

In the original Transformer paper, we mentioned that they used a novel absolute positional encoding scheme, based on sinusoidal functions. Specifically, they use the following positional encoding formulas where *pos* is the absolute position in the sequence and i is the ith element of the positional encoding vector. Note that d_k corresponds to each node feature vector dimension since this positional encoding should fit with every chosen word embedding size.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_k}) \tag{4.2}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_k}) \tag{4.3}$$

The intuition behind this specific encoding lies on the fact that the model would be able to generalise to longer sequences due to the use of sinusoidal functions, instead of a simple [0,1] distance scheme with predifined maximum sequence distances.

Shaw et. al [48] proposed a relative positional encoding method, based on the hypothesis that it is more useful than absolute positional encoding. We adopted this method and constructed a graph relative encoding scheme. For every unweighted/weighted graph in the dataset we initially calculate all the simple paths starting from the perturbation node to every other node in the graph. Since the process of calculating every node pair in each one of the 70000 graphs is computationally ineffective, we precalculated a protein path distance matrix PD in which

$$PD_{ij} = \sqrt{PE(pos_i) * PE(pos_j)^T} \tag{4.4}$$

So, for every graph we construct the sequence distance matrix A_{seq} defined as

$$A_{seq}(i,j) = PD(pos_i, pos_j) \tag{4.5}$$

Another more computationally efficient alternative would be to use the Floyd-Warshall distance matrix. By using its elements with equations 4.2 and 4.3 we construct the matrix $R \in \mathbb{R}^{h \times N \times N \times d_{head}}$, where R_{hij} represents the relative positional embedding vector of head h. Note that d_head corresponds to the dimension of each attention head.

4.1.2 Activity Embedding

Apart from the activity edge features that were described in a previous section, each node(protein) is embedded with two distinct positive values that range from 0 to 1. Those two values refer to the frequency each node in the weighted graphs had an upregulated or downregulated activity value in the initial unweighted graphs from which it was created.

Instead of just plugging those values in the feature vector of each individual protein, thus assigning only two weight values to each activity, we decided to use a more appropriate method to enhance the presence of such importan features. By projecting this 1x2 x_{act} vector into a 1xd_k vector, where d_k is the dimensionality of each protein feature vector, we end up with a trainable activity vector $u_{act} = x_{act}w_{act}$, with w_{act} being the 2xd_k weight matrix. The activity embedding is added with the trainable feature vector u_{prot} of each protein before they are processed by the Transformer encoder.

4.1.3 Modified Scaled Dot-Product Attention

The modified version of the scaled dot-product attention described in a previous section, has to account for both the new positional encoding method as well as the insertion of the edge features the DAG contains, something the word sentences that were processed with the original Transformer did not include.

Recall the query and key matrices Q, K from Eq 2.2. In the modified attention scheme, $U^h_{act} = X^h_{act}W^h_{act}$ and $U^h_{prot} = X^h_{prot}W^h_{prot}$ represent the matrices of each proteins activity and functional features for head h respectively. U_{edge} represents the (3xN) edge matrix of the graph, with features described in the Data section, with N being the number of proteins. So, the modified attention weights W_{attn} are calculated in the following scheme:

$$W_{attn}^h = softmax\left(\frac{(U_{act}^{Qh} + U_{prot}^{Qh}) * (U_{act}^{Kh} + U_{prot}^{Kh})^T}{\sqrt{d_k}} + \beta * (U_{edge} \circledast W_e)\right) \tag{4.6}$$

where W_e is the (1x3) edge weight and b is the bias. Essentialy, the process $U_{edge} \circledast W_e$ describes a pointwise convolution operation on the edge features. Also, β is a trainable parameter that weights the importance of the edge features.

Since the multi-head attention scheme remains the same, after the concatenation of each attention head, the value matrix V is weighted using the concatenated attention heads as

$$X = W_{attn}(U_{act}^V + U_{prot}^V) \tag{4.7}$$

where X is the (N x d_k) resulting node feature matrix before the pointwise feed-forward network, following the typical Transformer architecture.Adding our relative positional embedding and the trainable parameter c, results in the final form of the self attention mechanism:

$$W_{attn}^h = softmax\left(C^h + \beta(U_{edge} \circledast W_e) + c * PD\right) \tag{4.8}$$

where PD is the relative positional encoding matrix and is shared among each head. D is defined as the pairwise protein embedding multiplication matrix for each head:

$$C^h = (U_{act}^{Qh} + U_{prot}^{Qh})(U_{act}^{Kh} + U_{prot}^{Kh})^T \tag{4.9}$$

The Floyd-Warshall alternative exhibits a different attention scheme, which is as follows:

$$W^h_{attn} = softmax\left(C^h + \beta(U_{edge} \circledast W_e) + \frac{(U^{Qh}_{act} + U^{Qh}_{prot})R^{FW}_c}{\sqrt{d_{head}}}\right) \tag{4.10}$$

where R^{FW}_c is the corrected R FW matrix in terms of dimensions, so that the equation above holds. We can now identify three different components that arise from the attention mechanism. C^h is the content attention component, $\beta(U_{edge} \circledast W_e)$ is the edge component and $(U^{Qh}_{act} + U^{Qh}_{prot})R^{FW}_c$ is the position component of each attention head.

4.2 Unsupervised Learning

The aim of unsupervised learning is to distinguish certain patterns in a specific dataset by using unlabeled data and minimal human interaction. In this thesis, we will test our signaling network data in a series of unsupervised tasks to determine whether they prove useful. Additional models that will be used during the process will be briefly presented. Note that our DAG Transformer and some other models were developed with PyTorch[43] and PyTorch Geometric[12] and others were built using Tensorflow-Keras[11].

4.2.1 Additional Models

Graph2Vec

Graph2Vec[38] is an unsupervised neural graph embedding framework in order to learn task agnostic, data-driven distributed representations of arbitrary sized graphs. It is based on recent document embedding models, specifically Doc2Vec[29] in which the way that words/sentences compose documents is being exploited in order to learn specific document oriented embeddings. Note that this method doesnt need the node features that were mentioned in the data section, but each protein is labeled with a unique name. Apart from their original name, we further add a plus sign (+) to upregulated proteins and a minus sign (-) in the case of downregulated proteins to signify this functional attribute.

Siamese GED

As described in the DeepSIBA paper[15], this type of neural network consists of a siamese encoder architecture, where the encoder contains a multitude of graph convolutional, batch normalization, pooling and dropout layers. The hyperparameters of this architecture are given in the appropriate Hyperparameter section of the Supplamentary Material. The Siamese GED architecture is shown in Fig 4.1:

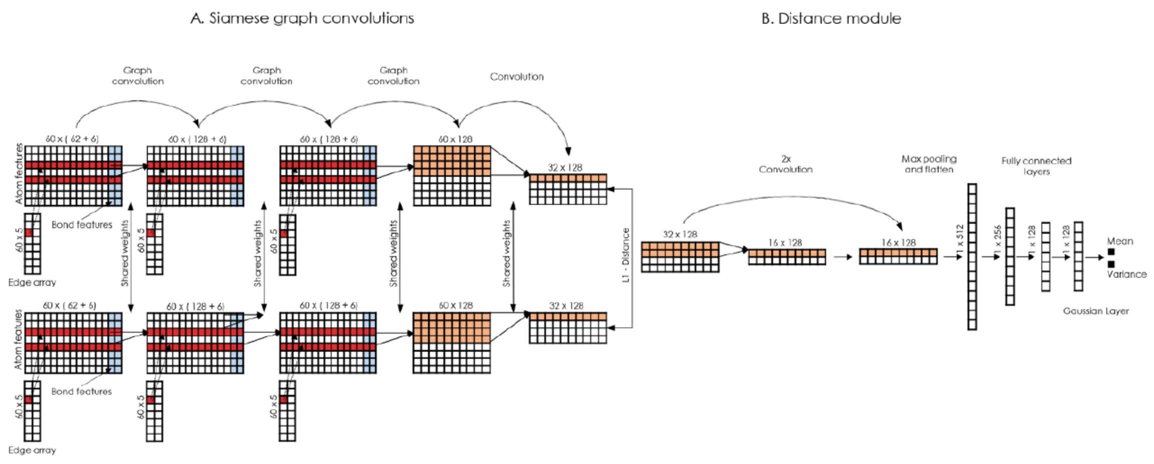

Figure 4.1: *The Siamese GED architecture[15]*

4.2.2 Graph Autoencoders

Initially called Autoassociative Neural Networks[28] autoencoders are a type of ANNs that produce compressed encodings of data in a completely unsupervised manner. It is a form of Non-Linear Principal Component Analysis and remains a basic method that involves neural networks in the process of dimensionality reduction. The input data is fed to the encoder which creates a latent compressed representation of the original data. The decoder then tries to reconstruct the original input data by minimizing a reconstruction loss like the Mean Squared Error (MSE) or Binary Cross Entropy.

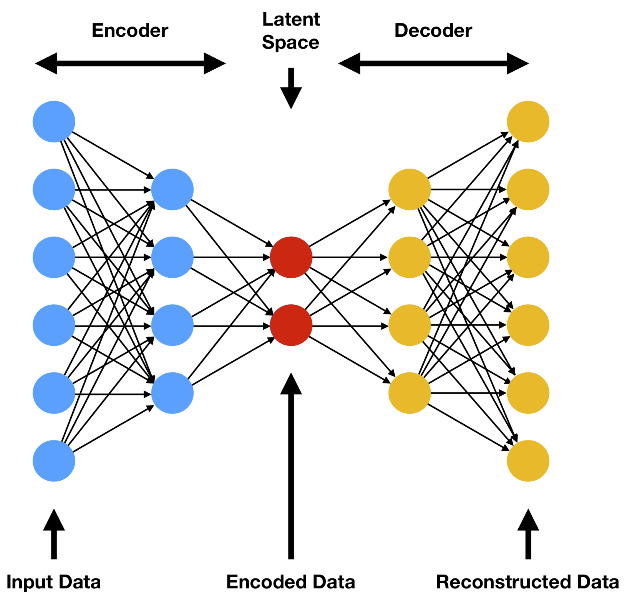

Figure 4.2: *Basic Autoencoder architecture*

Graph autoencoder(GAE)[46][33] share the same logic in terms of architecture, but with a few key differences. In GAEs, the reconstruction error that the network is trained to optimize has to account for both the node features as well as the overall graph structure. Thus, in the typical GAEs the decoder either tries to reconstruct the original adjacency matrix $\mathcal{A}$ or the node features $\mathcal{X}$ and some times even both simultaneously.

The selected architecture of our graph encoders revolves around edge prediction i.e. the network has to reconstruct the original adjacency matrix. We deliberately avoid predicting each nodes features due to the fact that the node embeddings are also trainable, a dangerous practice which in many cases, results in trivial solutions like zeros on each node embedding or identical node embeddings, none of which have any actual meaning. In our case, by using the proposed signaling network Transformer, after the Multi-Head Attention is performed, we end up with the global attentive node feature matrix X from Eq 4.7. As a global pooling method, we use Set2Set[30] and we end up with the summary vector $\vec{s} = Set2Set(X)$. Since we are interested in creating meaningful graph embeddings, we "filter" X with the trainable summary vector, ending up with the final global embedding matrix $X^s = X \odot \vec{s}$ where $X^s : \mathbb{R}^{N \times d_k}$ and $\odot$ denotes the elementwise product of the summary vector s and each one of the X node features.

After calculating X^s, it is used to calculate the pairwise euclidean distances between nodes, thus constructing the pairwise distance matrix D^e, which is defined as:

$$D^e_{ij} = \left\| x^s_i - x^s_j \right\| \tag{4.11}$$

In order to calculate the probability P_e that a distance between two nodes is an edge in the corresponding graph, we use a modified form of the Fermi-Dirac distribution

$$P_e(i,j) = \left[\exp \frac{D^e_{ij} - r}{t} + 1 \right]^{-1} \tag{4.12}$$

in which r is the radius and t is the temperature coefficient. For each (positive) edge that exists in each graph being processed, we sample 10 negative edges ($\widetilde{A}$) to train against. The BCE based loss function that we train to optimize is defined as follows:

$$\mathcal{L} = \mathbb{E}_A \left[\log P_e(i,j) \right] + \mathbb{E}_{\widetilde{A}} \left[\log \left(1 - P_e(i,j) \right) \right] \tag{4.13}$$

4.2.3 Deep Graph Infomax

Information Theoretic Definitions[6][37]

Entropy Let X be a random variable on a (discrete) space $\mathcal{X}$ and x an element sampled from $\mathcal{X}$. For every positive integer d, we denote by **X** a d-dimensional random vector $(X_1, ..., X_d) \in \mathcal{X}^d$, and by the letter **x** an element from $\mathcal{X}^d$. The Shannon entropy of a random variable X on a discrete space $\mathcal{X}$ measures its uncertainty during an experiment and is defined as

$$H[X] = -\sum_{x \in \mathcal{X}} P(X = x) log[P(X = x)] \tag{4.14}$$

The joint entropy of a pair of random variables (X, Y) expresses the uncertainty one has about the combination of these variables:

$$H[X, Y] = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P(X = x, Y = y) log[P(X = x, Y = y)] \tag{4.15}$$

Finally, the conditional entropy of a random variable X given another variable Y expresess the uncertainty on X which remains while Y is known:

$$H[X|Y] = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P(X = x, Y = y) log[P(X = x|Y = y)] \tag{4.16}$$

Mutual Information It is a general measure of the dependence between two random variables X, Y. It expresses the quantity of information one has obtained on X by observing Y. The discrete mutual information between two random variables X and Y is defined as:

$$I(X; Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P(X = x, Y = y) \log \frac{P(X = x, Y = y)}{P(X = x)P(Y = y)} \tag{4.17}$$

If we recall the definition of the Kullback-Leibler divergence between the distributions P, Q

$$KL(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \tag{4.18}$$

then Eq 4.7 describes the KL divergence between the joint distribution P(X, Y) and the product distribution P(X)P(Y). In terms of Shannon entropy, MI can be defined as

$$\begin{aligned} I(X; Y) &= H[X] - H[X|Y] \\ &= H[X] + H[Y] - H[X, Y] \\ &= H[X, Y] - H[X|Y] - H[Y|X] \end{aligned} \tag{4.19}$$

The Deep Graph InfoMax[59][24] approach to learning a suitable encoder relies on maximiziting local mutual information i.e. to obtain node representations that capture the global information content of the entire signaling network, which in this case is represented by a *summary vector* $\vec{s}$. We also refer to $\vec{h}$ as each node's *patch representation* after the DAG Transformer Encoder forward pass.

For the graph-level summary vectors $\vec{s}$, we use a readout function $\mathcal{R} : \mathbb{R}^{N \times d_k}$, which in our case is either an average pooling of the node features, or the Set2Set[30] global pooling method.

In order to maximize the local MI, a *discriminator* $\mathcal{D} : \mathbb{R}^{d_k} \times \mathbb{R}^{d_k} \rightarrow \mathbb{R}$ is used so that for each node i of the graph $\mathcal{D}(\vec{h}_i, \vec{s})$ it represents the probability scores assinged to the summary-patch pair.

The functionality of the discriminator depends on the existence of both positive and negative samples. For the graph $\mathcal{G}$ with a summary $\vec{s}$ negative samples are produced by pairing the summary with patch representations from another graph $\widetilde{\mathcal{G}}$ namely $\vec{\widetilde{h}}_j$. We use a combination of corrupted samples from each input graph as well as patch representations from graphs with a different signature id i.e. a different experiment, that should be different in nature. Corrupted representations are produced from an explicit, stochastic *corruption function* $\mathcal{C} : \mathbb{R}^{N \times d_k} \times \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{M \times d_k} \times \mathbb{R}^{M \times M}$ such that $\widetilde{\mathcal{G}} = \mathcal{C}(\mathcal{G})$. In our case, $\mathcal{C}$ is a random permutation of both the node features as well as the edge positions of the whole graph. Note that we slightly abuse the notation of $\mathcal{G}$. With this symbol, in our case, we represent the node and adjacency matrix ($\mathbf{X}$, $\mathbf{A}$).

Noise Contrastive Estimation (NCE)[40] is used as a lower bound on MI along with a standard Binary Cross Entropy (BCE) loss between the samples fromt the joint (positive examples) and the samples from the product of marginals (negative examples).Thus, the objective to be optimized is:

$$\mathcal{L} = \frac{1}{N+M}\left(\sum_{i=1}^{N} \mathbb{E}_{\mathcal{G}}\left[\log \mathcal{D}(\vec{h}_i, \vec{s})\right] + \sum_{j=1}^{M} \mathbb{E}_{\widetilde{\mathcal{G}}}\left[\log\left(1 - \mathcal{D}(\vec{\widetilde{h}}_j, \vec{s})\right)\right]\right) \tag{4.20}$$

As the discriminator probability estimation function we can choose the Fermi-Dirac distribution used in the Graph Autoencoder, as an alternative to the sigmoid function used in the original paper, due to the extra two degrees of freedom it includes. The loss function of Eq. 4.20 corresponds to the MI estimator used in the original Deep Graph Infomax paper.

We can use a variation of this loss function based on the Jensen-Shannon Mutual Information estimator as in InfoGraph[52] with the following notation, based on Nowozin et.al[39]. Let $I_{\phi,\psi}$ be the mutual information estimator modeled by a discriminator $\mathcal{D}_\psi$ parametrised by a neural network with parameters ψ.As ϕ, we denote the parameters of our modified transformer neural network. The Jensen-Shannon MI estimator is:

$$I_{\phi,\psi}\left(h^i_\phi, s_\phi\right) := \mathbb{E}_{\mathbb{P}}\left[-sp\left(-D_{\psi,\phi}\left(h^i_\phi, s_\phi\right)\right)\right] - \mathbb{E}_{\widetilde{\mathbb{P}}}\left[sp\left(D_{\psi,\phi}\left(\widetilde{h^i_\phi}, s_\phi\right)\right)\right] \tag{4.21}$$

where $\mathbb{P}$ is the empirical distribution of the input data set, $\widetilde{\mathbb{P}}$ is the negative distribution from which we sample from (either via the corruption function $\mathcal{C}$ or even better $\widetilde{\mathbb{P}} = \mathbb{P}$, i.e. masking networks from different signatures as described later in this section) and sp is the softplus function $sp(z) = (1 + e^z)$. Thus, the new model's unsupervised loss is

$$\mathcal{L}_{unsupervised} = -\frac{1}{N}\sum_{i}^{N} I_{\phi,\psi}\left(h^i_\phi, s_\phi\right) + \gamma D\phi, \psi\left(\mathbb{V}||\mathbb{U}_{\phi,\psi}\right) \tag{4.22}$$

The second term is a regularization loss, which denotes matching the pushforward distribution of our summary vectors to a prior distribution, with the most successive being the uniform distribution. The effects of this loss term were exhibited in the Signature as well as the Duplicate Similarity Task as depicted in the Results section. Prior Matching was inspired by research on adversarial methods in deep learning.

Instead of the Jensen-Shannon Divergence used in the original InfoGraph paper,we can make use of MINE (Mutual Information Neural Estimation)[7] which again uses a neural network as a discriminator, but is based on the Donsker Varadhan lower bound on KL divergence. Using the same notation, the unsupervised MINE objective can be formulated as follows:

$$\mathcal{L}^{MINE}_{unsupervised} = -\frac{1}{N}\sum_{i}^{N} D_{\psi,\phi}\left(h^{i}_{\phi}, s_{\phi}\right) + \ln\frac{1}{M}\sum_{j}^{M} e^{D_{\psi,\phi}\left(\widetilde{h^{j}_{\phi}}, s_{\phi}\right)} \tag{4.23}$$

where M refers to the number of nodes of the sampled negative graph. In order for the model to distinguish between duplicate experiments more easily, we make use of a shared mask between samples that come from a same signature, so that when multiplied, the negative expectation term between duplicate signatures will zero out and the positive one has a masking value of 1, thus forcing it to enhance mutual information between both duplicate experiments as well as graphs that come from the same signature, during the training phase.

4.2.4 Unsupervised Embedding Evaluation Tasks

The embeddings that are produced using the aforementioned unsupervised methods should be able to represent the functional and biological characteristics. Signaling networks that exhibit similar characteristics in the biological space should be closer in the embedding space than with those that differ in terms of functionality. We initially employ three different evaluation methods to determine whether the unsupervised embeddings are functionally and biologically distinguishable. These methods usually involve the unweighted networks, since we want to compare graphs from different signatures.

Signature Similarity Task (SST) As mentioned in the data section, each signature identification, after being processed with CARNIVAL, produces a multitude of unweighted signaling networks with a number that ranges from 5 to 100 graphs per signature id. Unweighted graphs that arise from the same signature should be closer in the embedding space than pairs that come from different signatures. We use our unsupervised learning models to learn embeddings that exhibit this behaviour and provide the results with various graphs.

Duplicate Similarity Task (DST) In out dataset, there exist different signature ids but they correspond to the same experiment performed at a different place in time, keeping the exact same characteristics in terms of dosage, drug, cell line etc. Ideally, those experiments should be completely identical both in terms of their GEx as well as the corresponding signaling networks produced. But as it turns out, we notice networks with slightly different structure for those duplicate experiments. Each model should be able to identify networks coming from the same experiment in terms of their distribution in the embedding space, this being the main goal of the task.

Mechanism of Action Distinction (MAD) Task For each of the labeled unsupervised embeddings after training, we attempt to visualize them in 2-D including the mechanism of action labels attributed to each, by using certain visualization techniques based on manifold learning clustering methods(t-SNE or UMAP). The visualization will help us determine whether the unsupervised embeddings encapsulate adequate information that can help distinguish different mechanisms of action without having to perform multi-class classification.

Apart from producing unsupervised embeddings, the models mentioned in this sections have an equally important role. It is common practice to pretrain unsupervised models in order to assist with future supervised downstream tasks, such as classification or regression.By splitting the trained encoder from the whole unsupervised model - when this is technically possible - and plugging it to a classifier, we will be able to leverage the unsupervised pretrained weights.

4.3 Semi-Supervised Learning

Apart from the toy datasets the machine learning commynity is accustomed to, most of the real world data sets come either completely or partly unlabelled. Thus, methods that combine both supervised and unsupervised learning arised to address this issue. As mentioned previously, our dataset consists of 7788 unique signatures and 2733 of them are actually labelled. A fully supervised method would have to not take account of the unlabelled data thus throwing away a potentially valuable amount of information. We modify two of our models to fit with this training method, the Siamese GED network and the Deep Graph Infomax.

In the case of the Siamese GED, we can train based on the mechanism of action labels by introducing a loss based on each pairs labels, apart from their structural distance.

In order to train the Infomax model we introduce a new supervised loss between positive and negative examples. The fact that makes this loss supervised is that the distinction between positive and negative samples depends on their mechanism of action. So, by creating pairs of signatures with different mechanisms of action, we enforce the model to learn their biological and structural distinction via mutual information maximization. Since we use two different encoders for this task, let χ be the parameters of a new encoder and $\psi^{'}$ the parameters of the corresponding discriminator,both with the exact same architecture as the unsupervised case. The supervised MI is then:

$$I_{\chi,\psi'}\left(h_{\chi}^{i}, s_{\chi}\right) := \mathbb{E}_{\mathbb{P}}\left[-sp\left(-D_{\psi',\chi}\left(h_{\chi}^{i}, s_{\chi}\right)\right)\right] - \mathbb{E}_{\widetilde{\mathbb{P}'}}\left[sp\left(D_{\psi',\chi}\left(\widetilde{h_{\chi}^{j}, s_{\chi}}\right)\right)\right] \tag{4.24}$$

and the loss:

$$\mathcal{L}_{supervised} = -\frac{1}{N+M}\left(\sum_{i}^{N}\left(-sp\left(-D_{\psi',\chi}\left(h_{\chi}^{i}, s_{\chi}\right)\right)\right) - \sum_{j}^{M}\left(sp\left(D_{\psi',\chi}\left(\widetilde{h_{\chi}^{j}}, s_{\chi}\right)\right)\right)\right) \tag{4.25}$$

Lastly, we also need a way to connect the two seperate encoder parameters during training. Based on [52] we also tie the global embeddings using mutual information, but in a different way. We want the global embeddings of the positive examples to share more information with the unsupervised ones rather than the negative ones. This global loss is formulated as follows, based on the notation of the previous equations:

$$\mathcal{L}_{global} = sp\left(D_{\psi'',\phi,\chi}\left(s_{\chi}, s_{\phi}\right)\right) - sp\left(D_{\psi'',\chi}\left(s_{\chi}\widetilde{s_{\chi}}\right)\right) \tag{4.26}$$

Thus, the semi supervised objective is

$$\mathcal{L}_{semi} = \mathcal{L}_{supervised} + \mathcal{L}_{unsupervised} + \lambda * \mathcal{L}_{global} \tag{4.27}$$

where λ is a chosen hyperparameter. Since this method does not explicitly train the networks in identifying we dont expect it to be rather confident in case of individual mechanism of action prediction. If we were to further push the network towards the classification of the labelled instances of the dataset, we could change $\mathcal{L}_{supervised}$ to a typical classification loss and update the global parameters accordingly.

4.4 Masked Semi-Supervised Learning

After trying various combinations of the methods described in the previous two sections, we came to the conclusion that masking does indeed boost the model's performance since it introduces much needed bias regarding the input data in a rather safe way. To be more specific, in the semi-supervised environment, we developed a fast and reliable way to introduce the mechanism of action labels into the dataset. Apart from the already used duplicate masking method, each positive and negative sample is masked with a similar one hot encoded binary mask that indicates each labelled graph's attributed mechanism of action.

In each seperate mask, we also encode the uncertainty we have about the mechanism of action labels. We demonstrate this technique with a simple example. Consider the case of three total labels in our dataset 1, 2 and 3. Then the label number 2 would have the one hot encoded vector [0,1,0]. Say that we are 90% sure about the validity of this label. Then, the mechanism of action vector turns into [0.33,0.9,0.33]. This simple masking scheme, after many different experiments with various methods, provided both fast -in terms of computation- as well as equally and usually better clustering results.

4.5 Supervised Learning

During supervised learning procedures, each of the chosen models are being trained to identify one of the drugs' mechanisms of action. The training circumstances are definitely not on our side, since we can only account for 2733 labels in a dataset consisting of 7788 different experiments. Thus, the evaluation metrics for this task will not be ideal, but the ultimate goal is not to provide a tool for predicting drug mechanisms of action(at least for now), but to create an interpretable signaling network embedding framework.

4.5.1 Additional Models

XGBoost

XGBoost[10] is an optimized distributed gradient boosting library. All its algorithms are implemented under the Gradient Boosting framework, i.e. the production of a classification or regression model in the form of an ensemble of weak prediction models. These aforementioned weak learners are in most cases decision trees.

4.6 Node Importance Calculation

Evaluating each model's performance on the task at hand, that being supervised, unsupervised or semi-supervised is one major part of this diploma thesis. But only the task of training, especially for this quite challenging dataset, must be accompanied by interpretable results with a biological significance. A simple way in order to generate said results, one can use a trained model and calculate the gradients of the output with respect to the input. In our case, with the input being the total of the graphs nodes accompanied by their node features - SeqVeq or GO term features- this gradient , per node, is in the following form. Let $F \in R^n \rightarrow [0,1]$ be the function that represents our neural network in a supervised task and x_i^j node j's i^{th} feature. Then, the gradients with respect to each node:

$$Grads_j = \sum_i^{d_k} \frac{\partial F}{\partial x_i^j} \tag{4.28}$$

If y_k represents the k^{th} intermediate layer, then by the chain rule:

$$Grads(x_j) = \sum_i^{d_k} \frac{\partial F}{\partial y_{k_n}} \frac{\partial y_{k_n}}{\partial y_{k_{n-1}}} \cdots \frac{\partial y_{k_1}}{\partial x_i^j} \tag{4.29}$$

Instead of calculating said gradients using Eq 4.28, we can also employ a slightly more complex gradient attribution method, called Integrated Gradients[53]. Let $x^{j'}$ be a baseline input node ,e.g. a black or white image for image classification task or the zero embedding vector for natural language processing tasks and in our case, the original pretrained node embeddings.The integradient gradients of the output with respect to the input are calculated as follows:

$$IntegratedGrads(x_j) := \sum_{i}^{d_k} \left((x_i^j - x_i^{j'}) \times \int_{\alpha}^{1} \frac{\partial F(x_i^{j'} + \alpha \times (x_i^j - x_i^{j'}))}{\partial x_i^j} d\alpha \right) \tag{4.30}$$

With this expression, we can calculate the gradient attribution of each node in the graph thus determining their importance. Each of the gradients above can be calculated easily using PyTorch/Autograd and the integral is calculated using the Gauss-Legendre integral approximation method, by using 50-100 forward-backpropagation passes to estimate the IG.

The problem when using IG, as we verified in our experiments, is that it is very time consuming to evaluate large quantities of experimental data in terms of gradient attribution, due to the increased number of backpropagation steps needed to calculate each gradient attribution with a small convergence delta. So, in order to quantify the importance of each node we use simple saliancy methods described by Equations 4.28 and 4.29.

Chapter 5

Embedding Quality Evaluation

After presenting each one of the methods used in this thesis, both in terms of unsupervised as well as supervised nature, the following sections include the results of each model in the given tasks. Note that in each model, no hyperparameter training was performed due to both the multitude of the methods as well as the huge computational cost of optimizing the hyperparameters of each model. Hyperparameters were chosen empirically with trial and error, and the exact architecture of each model is presented in the corresponding Hyperparameter section of the Supplamentary Material.

5.1 Unsupervised Task Evaluation

Since we are not aiming to create a single model for a predictive task, but exhibit the properties of the signaling network dataset, we consider that there is no point in presenting most of the evaluation metrics during training. Instead, we focus on presenting the results that involve the unsupervised evaluation tasks in Section 4.2.4, the Signature Similarity Task and the Duplicate Similarity Task. The tasks were performed with either the GO term node features or the SeqVeq node features, as we refrained from using a mixture of both in order to determine the performance of each initial representation on each given task.

5.1.1 SST Evaluation

Recall the SST from Section 4.2.4. After training each model we calculate either the cosine distances CD between pairs of graph embeddings u, v

$$CD(u, v) = 1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2} \tag{5.1}$$

or the Euclidean distance between u and v. The cosine distance provided more consistent results, which is why we selected it as the appropriate distance metric for each task. For each of the following distance plots, we select a number of unweighted graphs from the same signature and afterwards calculate their pairwise cosine distances. Then, we calculate the pairwise distance of unweighted graphs against different random signature graphs, ending up with the distance distribution plots presented in the following subsections.

Graph2Vec SST

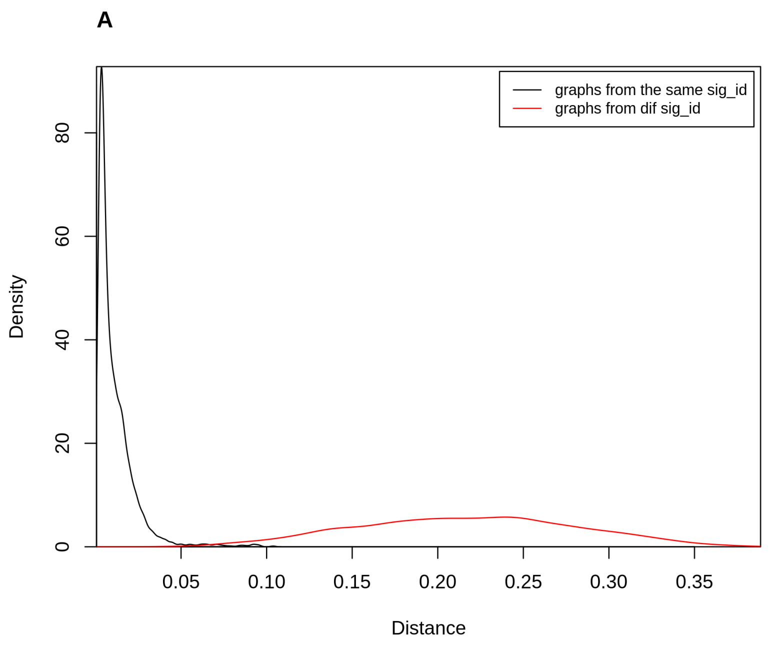

Figure 5.1: *Graph2Vec SST Performance*

As we can see, Graph2Vec is very succesful in distinguishing between graphs from the same signatures against random ones. The form of these distribution is also almost ideal, despite the minor overlapping. The black distribution has a very small variance contrary to the red one, a fact which exhibits that the same signature graphs have a minor cosine distance as opposed to the random graphs, where the large distribution variance accounts for the varying difference between the random graph distances. We also conctruct a UMAP plot of graph embeddings, with each label corresponding to a signature.

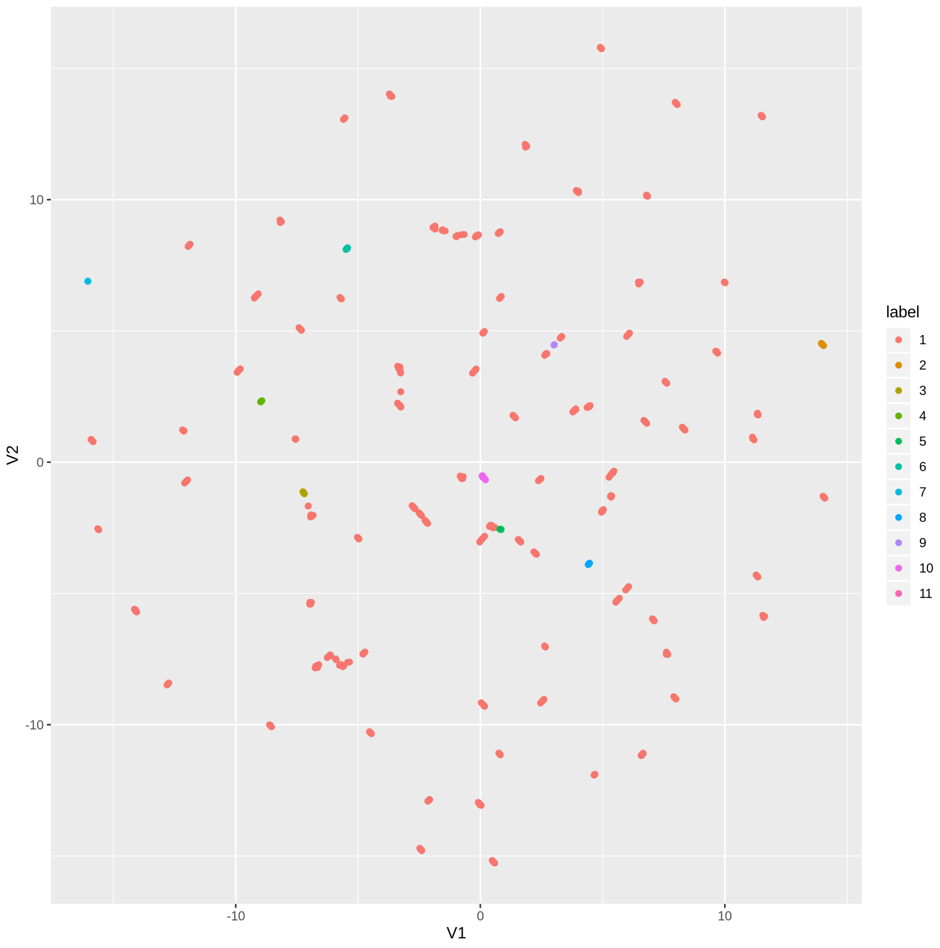

Figure 5.2: *UMAP plot of Graph2Vec graph embeddings. Each label corresponds to a different signature identification.*

Figure 5.2 clearly depicts the success of the Graph2Vec method. Embeddings from the same signatures form very tight clusters that almost seem as one data point, except for the distribution of class 1 labels. The latter, form sparse clusters in 2-D space which may lead us to believe that there exist multiple sub-categories of class 1 signatures, with slightly different biological or structural information.

Siamese GED SST

We trained the Siamese GED model for 4 epochs, with various node embedding sizes, specifically 128, 256 and 512. Since in most of our models the latter prevailed, for constistency reasons we choose to present only the evaluation visualizations concerning this embedding dimension.

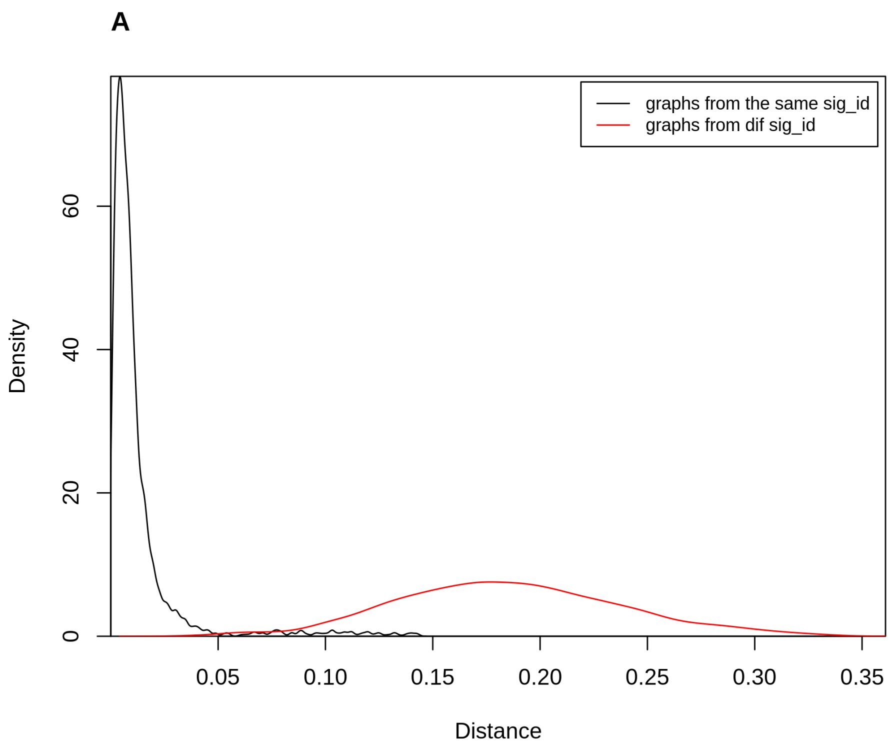

Figure 5.3: *Siamese GED SST Performance*

We can see that Siamese GED is again succesful when dealing with similar versus random signature embedding distances. The black distribution again has very small variance compared to the random distance one with minor overlap. In similar fashion, we present the UMAP embeddings sampled from various signatures:

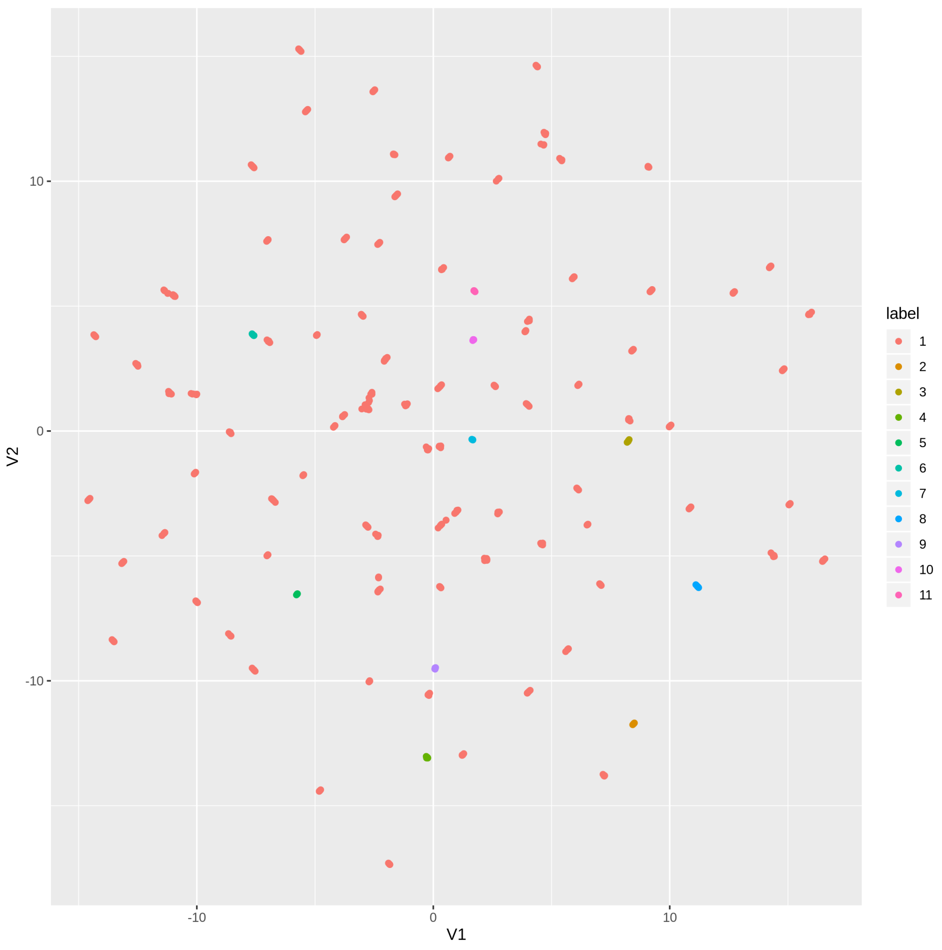

Figure 5.4: *UMAP plot of Siamese GED graph embeddings. Each label corresponds to a different signature identification.*

Based on the figure above, its almost total similarity to Figure 5.2 leads us to the same explanation of its form and the corresponding signaling network embeddings.

Graph Autoencoder SST

Similarly, we construct both the plots used in the previous evaluation methods. Note that for the Graph Autoencoder method presented below, both GO and SeqVeq node embeddings produced almost identical results, with no clear winner regarding the evaluation task. We arbitrarily choose to present the SeqVeq pre-embedded model evaluation graphs.

The density plot of Figure 5.5 is slight dissimilar as opossed to the previous two,

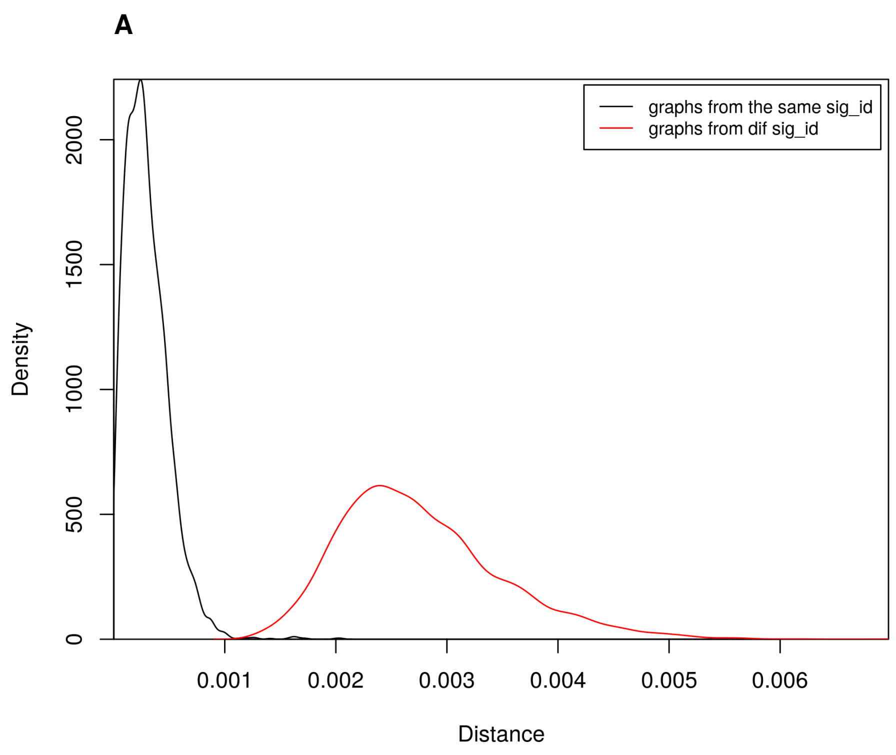

Figure 5.5: *Graph Autoencoder SST Performance*

regarding the variance of the random graph distance distribution. This difference is to be expected since the Graph Autoencoder is a non-contrastive method with a very simple adjacency matrix reconstruction task, but still, it captures the essential information to be able to distinguish between same signature graphs and random ones. The UMAP embedding plot of figure 5.6 is practically identical to the previous two, but we present it for the sake of completeness.

Signaling Network Infomax SST

In the unsupervised setting, SNI was trained with the corruption function method presented in section 4.2.3. We deliberately refrained from using a contrastive task for DGI at the moment, since this adds possibly unwanted bias to our model i.e. by inserting graphs from different signatures as negative samples, since are not sure of their dissimilarity. Instead, the corruption function offers a bias free method with good results regarding the SST.

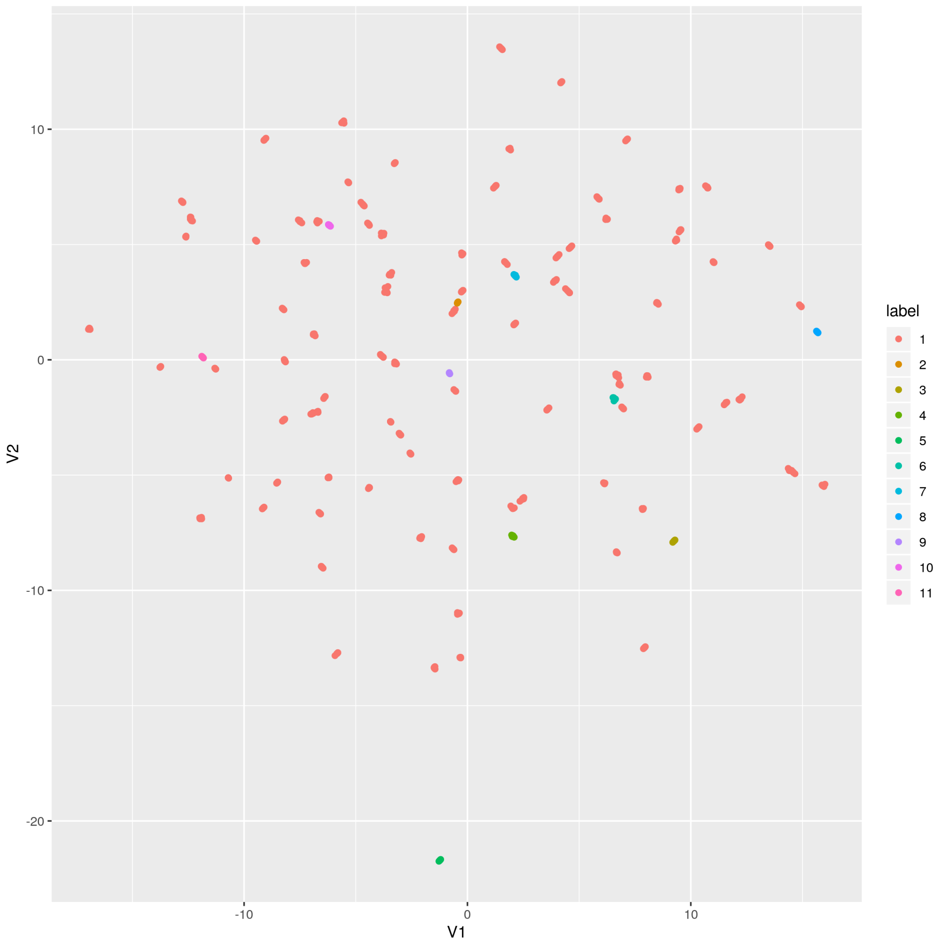

Figure 5.6: *UMAP plot of Graph Autoencoder graph embeddings. Each label corresponds to a different signature identification.*

In the SNI setting of Figure 5.7, we still notice the succesfulness on the SST task in terms of signature distance difference. The expected UMAP embedding plot of again shows that similar signature identifications form tight clusters, with the consistent exception of class 1 signature identifications who are sparsely clustered in small regions which is omitted as it is similar to the rest.

5.1.2 DST Evaluation

As opposed to the SST, the Duplicate Similarity Task (DST) is a slightly harder task to deal with. This mainly due to the inconsistency of the original GEx data. In the ideal case, GEx data obtained from the exact same experiment performed

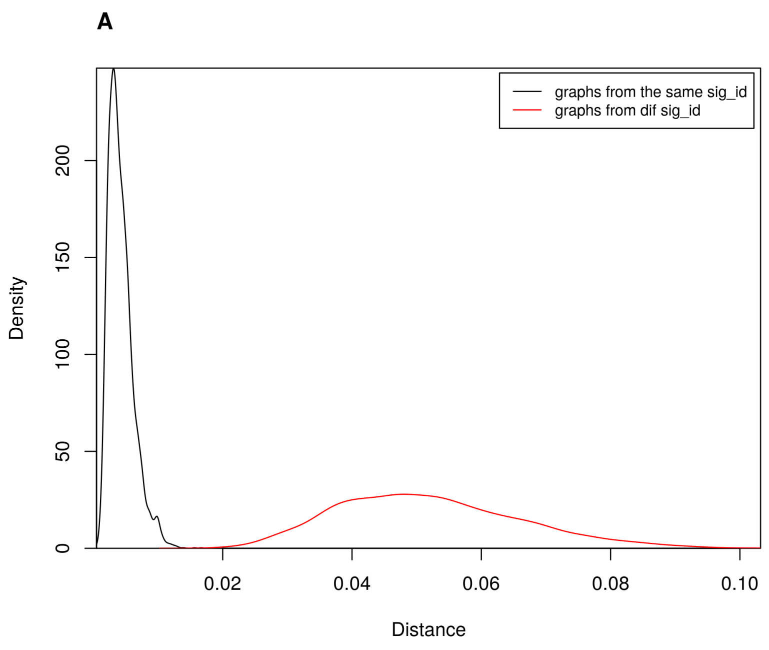

Figure 5.7: *Deep Graph Infomax SST Performance*

in a different place in time - which we refer to as **duplicate** experiment- should be completely identical, but this differs from reality by a margin. The noise that is packed along with the original GEx data is of course administered into the signaling networks produced by CARNIVAL. Thus, in the DST, we can see the effect of noisy input data depicted in the density plots that will be presented in the following sections. The aforemention distance density plots were created together with the ones from the SST, meaning that they both stem from the exact same dataset with the same training hyperparameters in each case to ensure consistency of each models output in terms of post training intepretability.

Graph2Vec DST Evaluation

In the new more challenging task, we present the evaluation graphs in the same order as before, starting from Graph2Vec.

While in the SST Graph2Vec and the other models had no problem distinguishing between random and same signatures, we can clearly see the influence of noisy data

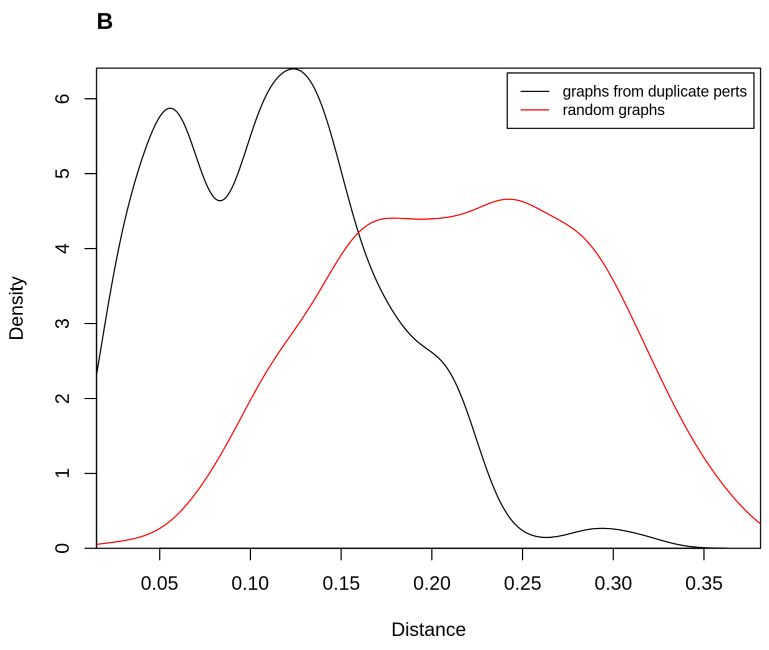

Figure 5.8: *Graph2Vec DST Performance*

in the density plot of Figure 5.8. In this case, not only there is a relatively big overlap among the two distributions, but their distributions have acquired a larger variation than the previous ones, as well as multimodality.

Siamese-GED DST Evaluation

In the Siamese-GED model, we can see that the results are virtually the same. Multimodality in the black distribution with increased variance than the SST and a relatively big overlap between the two distributions. Note that both the Graph2Vec and Siamese GED DST density plots have a distance that revolves around 0.10, whereas in the SST case it was very close to 0.0 showing the significant difference between duplicate signatures. Figure 5.9 contains the aforementioned distance density plot.

Graph Autoencoder DST Evaluation

The Graph Autoencoder produced rather acceptable results both in the SST as well as the DST. In Figure 5.10, we do not spot the same behaviour as in the previous to

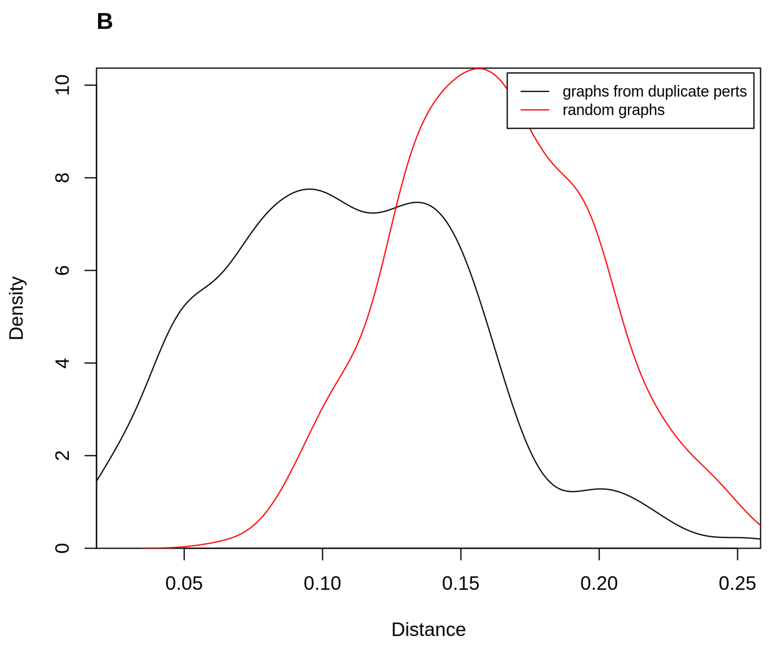

Figure 5.9: *Siamese GED DST Performance*

DST distance density plots. The network is able to produce unsupervised embeddings without the multimodality that Graph2Vec and Siamese GED exhibited in their DST plots. Yet, the unwanted overlap between the two distributions pertains, enhancing the belief that it is mainly a problem regarding the input data. The significance of the training task can be easily seen in the difference between Figures 5.10 and 5.11. Even though both use the same architecture and dataset, their DST density plots differ substantially.

Signaling Network Infomax DST Evaluation

Lastly, we present the results produced by the SNI model on the DST. We notice an even smaller overlap between the red and black distributiion, which indicates that the method in a way succeeded in distinguishing between random and duplicate signatures. This indicates that the method we used to train the model, i.e. masking duplicate signatures under a same mask value, has successfully worked.

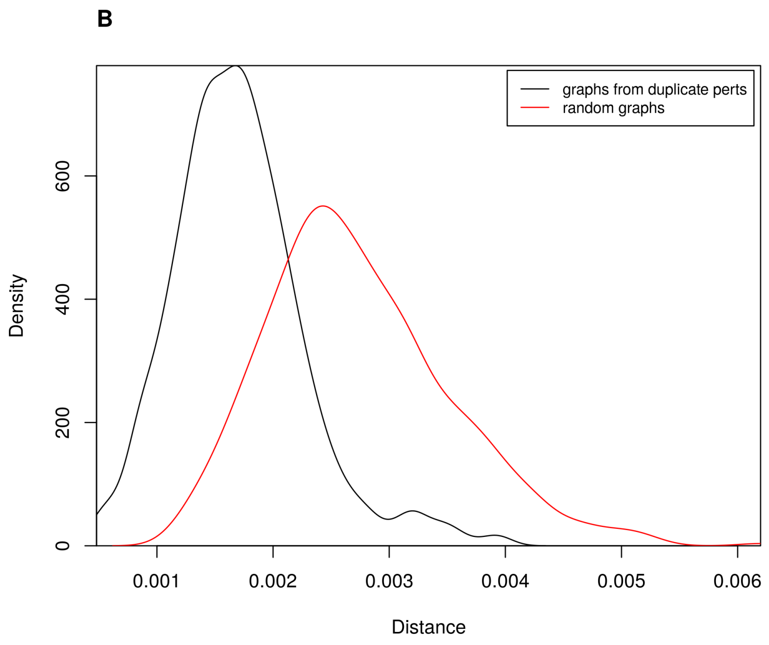

Figure 5.10: *Graph Autoencoder DST Performance*

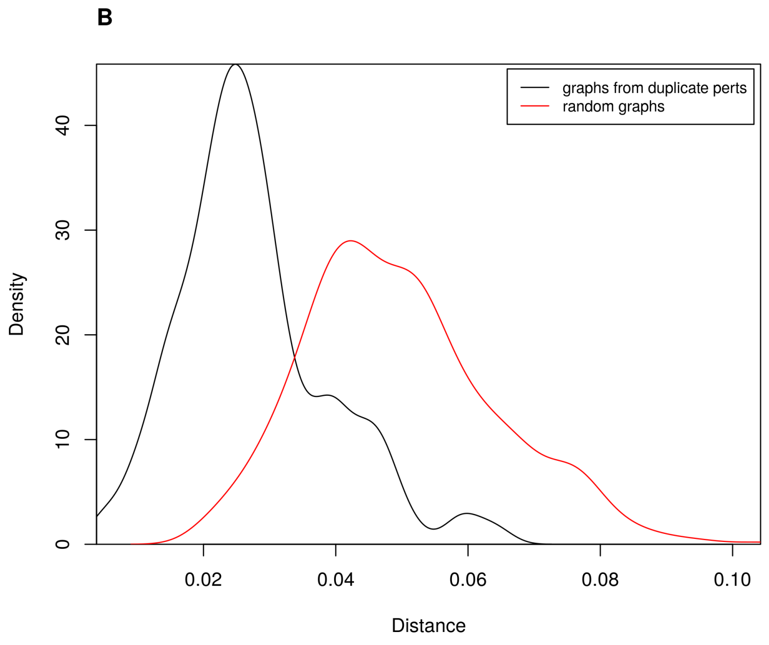

Figure 5.11: *Signaling Network Infomax DST Performance*

5.1.3 MAD Task Evaluation

The third and final evaluation task is essentially the true purpose of this diploma thesis. The ultimate goal that each one of the models must accomplish, is the accurate and interpretable prediction of each drugs mechanism of action. But this goal, from both a data analytics as well as a bioinformatics standpoint, is not as simple as it might sound. Consider the case of a simple object detection neural network which is trained on a labelled image dataset. In this image dataset, an object labelled as "car" is definitely only a car and there is no question that it may be a "plane" or a "dog". So the network can be trained to identify a single label that describes the objects seen in training and testing time. In the case of signaling networks, no one can be definitely sure that the drug administered has a single mechanism of action labelled as "DNA kinase inhibitor" and simultaneously not being an antidepressant. This is exaclty the reason which enhances the difficulty of this task. In the figures presented in the following subsections, each unsupervised model embedding set was reduced in 2-D by t-SNE while attributing each signature with its own label. Ten mechanisms of action were used for each unsupervised embedding plot.

Graph2Vec MAD Evaluation

We evaluate the labelled unsupervised embeddings of Graph2Vec using t-SNE, as shown in Figure 2.11. The output of the model caused the unsupervised embeddings to be somehow unevenly clustered regarding their mechanism of action. We can see some degree of clustering according to their mechanism of action, especially the green, "dopamine antagonist" cluster which seems to be the most stable contrary to the rest.

Siamese GED MAD Evaluation

In Figure 5.13, we repeated the same procedure for the labelled unsupervised embeddings of Siamese GED. The same explanation as in the case of Graph2Vec can

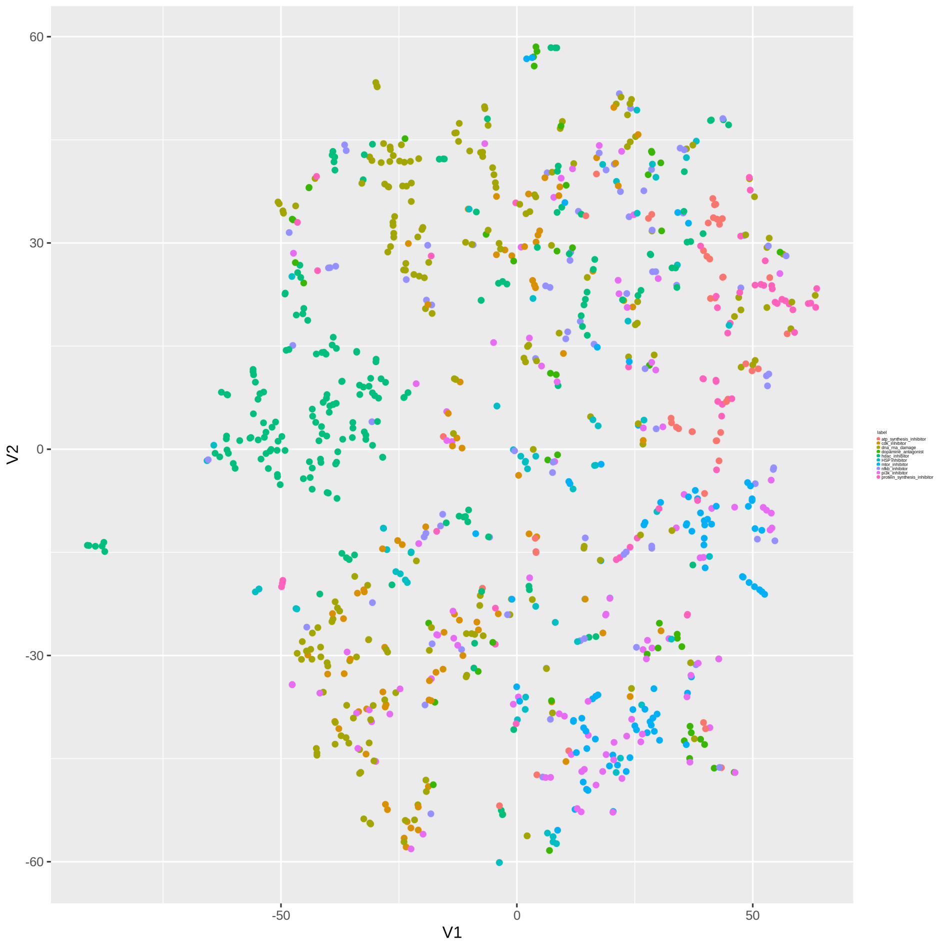

Figure 5.12: *Graph2Vec MAD Performance*

be given for this figure as well.

Graph Autoencoder MAD Evaluation

The same evaluation procedure is followed with the labelled Graph Autoencoder unsupervised embeddings show in Figure 5.14. Again, the "hdac inhibitor"(green) unsupervised embeddings seem to be the most tightly clustered, and in the Graph Autoencoder case, the "atp synthesis inhibitor"(red) and "mtor inhibitor"(blue) ones also seem to form non-sparse clusters. The rest again have small communities of similar mechanism of action but not a centralised area where they all lie.

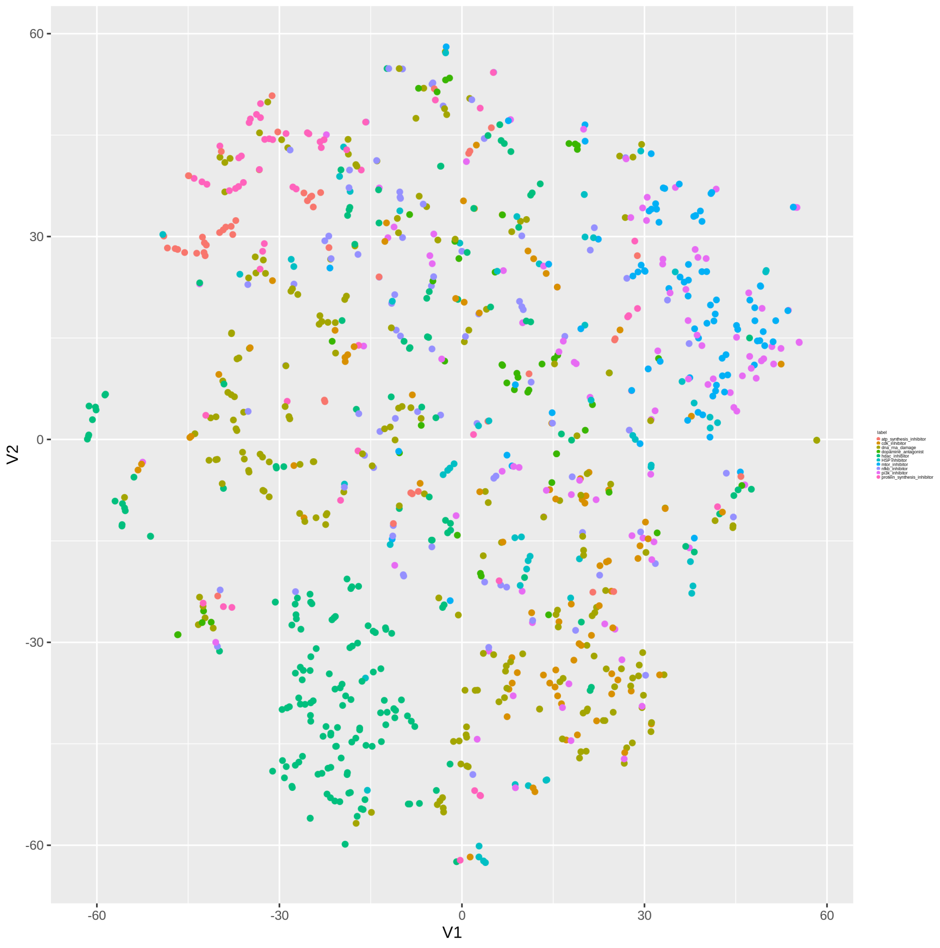

Figure 5.13: *Siamese GED MAD Performance*

Signaling Network Infomax MAD Evaluation

Lastly, the SNI labelled unsupervised embeddings are being presented in Figure 5.15. In this case, we notice that despite the fact that again the "hdac inhibitor" embeddings seem to cluster together, the other ones are more sparsely distributed, in small similar mechanism of action communities. This is not necessarily unwanted, since we do not really know if this 2-D overlapping means that the embeddings are wrongly distributed. It may be the case that they share common structural or even biological similarities.

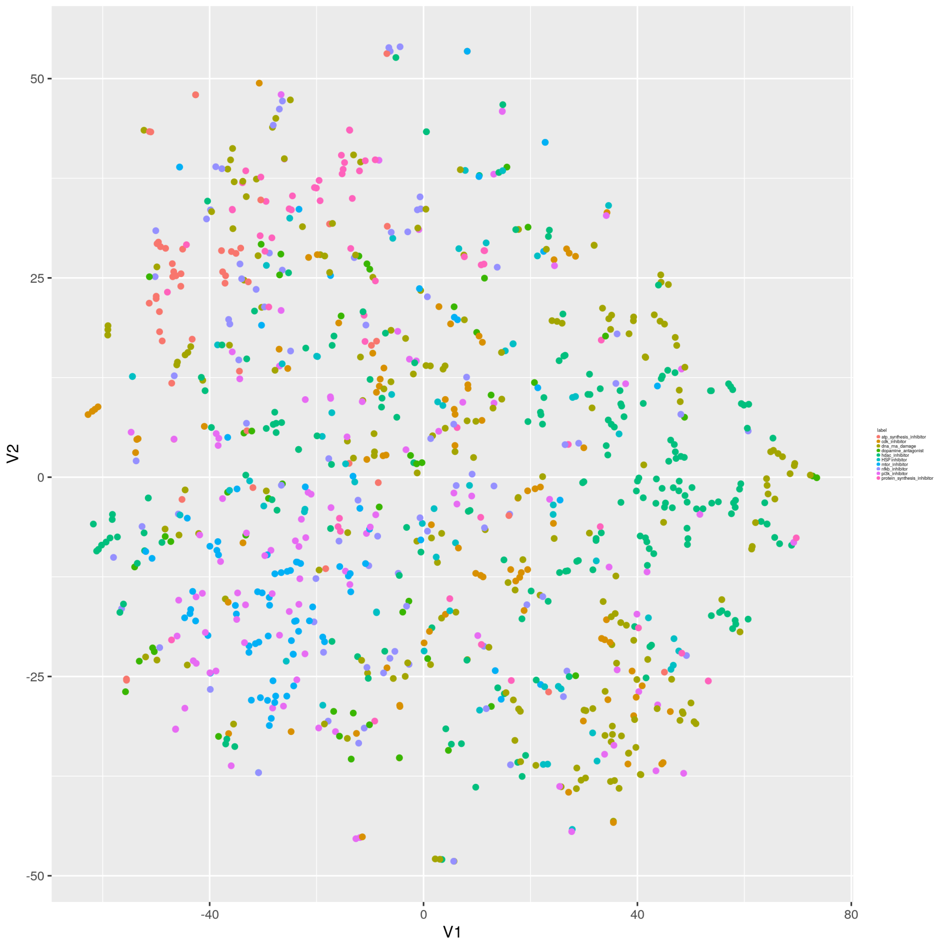

Figure 5.14: *Graph Autoencoder MAD Performance*

5.2 Semi-Supervised Task Evaluation

In this section, the semi-supervised embeddings of the Siamese GED model and the semi supervised Signaling Network Infomax are being presented. Note that the models were trained on different dataset splits, since they tackle this task with a slightly different way in terms of batching.

5.2.1 Semi Supervised Siamese GED

The figure belows depicts almost the ideal clustering of our mechanism of action labels. Yet, this figure shows a case of overfitting, since the same model was trained on a huge dataset for almost 4 hours per epoch, and it greatly failed regarding the

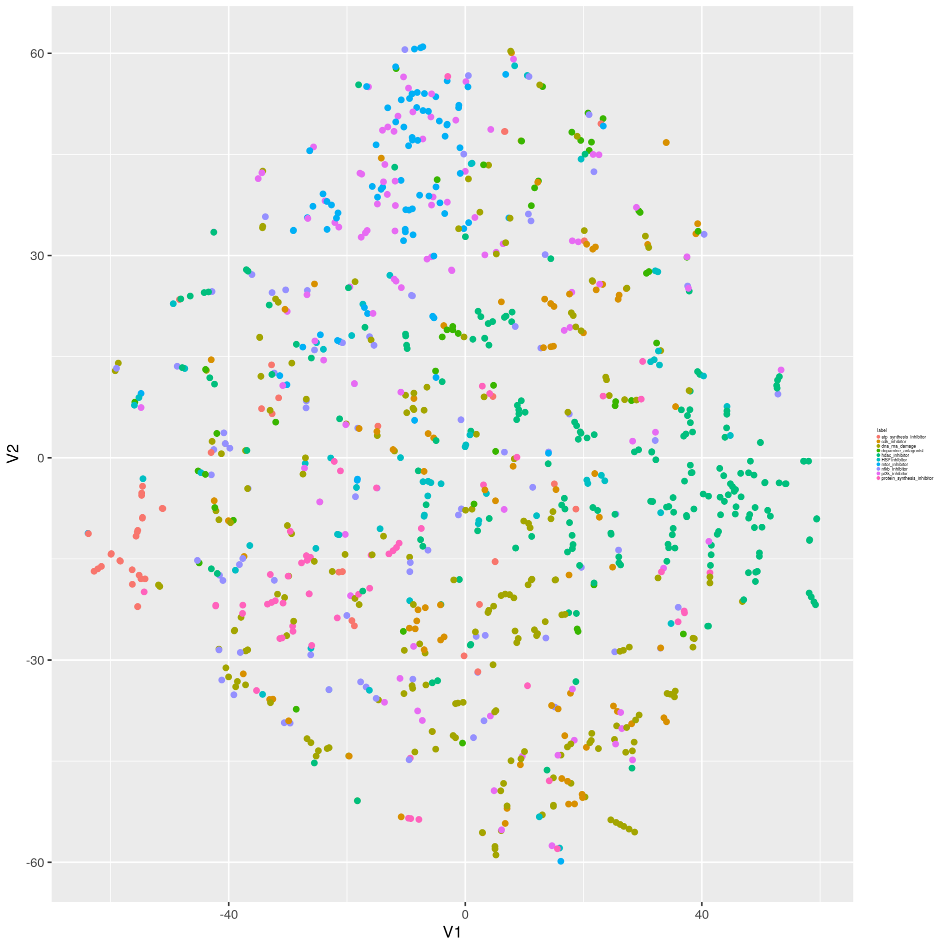

Figure 5.15: *Signaling Network Infomax MAD Performance*

evaluation on the test set. Nevertheless, one interesting aspect of it is the overlap between the "mtor inhibitor" labelled signatures and the "pi3k inhibitor" ones, since it is very usual for a drug to exhibit dual inhibition in the mTOR and PI3K kinases [23].

5.2.2 Semi Supervised Signaling Network Infomax

The semi supervised Signaling Network embeddings are depicted in Figure 5.17. Since this approach is less invasive than the previous one i.e. the embeddings were not trained explicitly to match their labelled mechanisms of action, we do not notice the same almost ideal clustering behaviour as in Figure 5.16. Nevertheless, the semi-supervised SNI, has some improved performance compared to the unsupervised case.

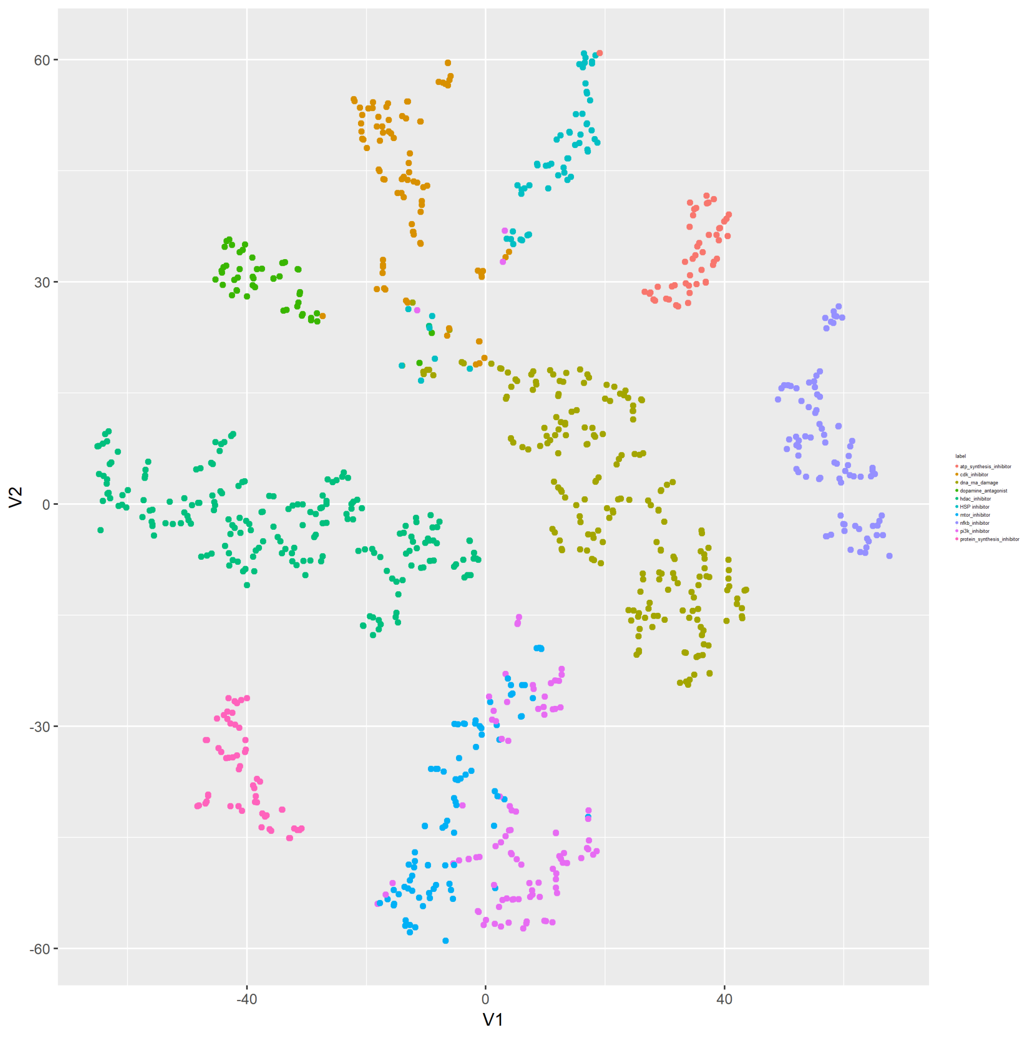

Figure 5.16: *Semi Supervised Siamese GED Clustering Performance*

Again, the HDAC inhibitor and mTOR/PI3K inhibitor mechanism of action seems to have the best clustering. Note that initially, the green data points were either mTOR, PI3K or both, but since they rather oftenly appear close in the embedding space due to reasons of usual dual inhibition, we decided to group them under one mutual category for visualization reasons.

It is very interesting that while the semi supervised model succeeded in clustering the hdac and mtor/pi3k inhibitor labelled networks, it still struggles to clearly distinguish between the other mechanisms of action involved in Figure 5.17. In order to determine this discrepancy quantitatively, one can continue with the node importance calculation scheme provided in Section 4.5, thus determining the importance the model attributed to each node in order to end up with such an embedding. We can also calculate the node importance in the semi supervised or even the unsupervised environment, but this is heavily intensive in terms of computation since we have to calculate the importance of each node on every dimension of the output data, i.e. $N \times 512 \times 1024$ integrated gradients per network.

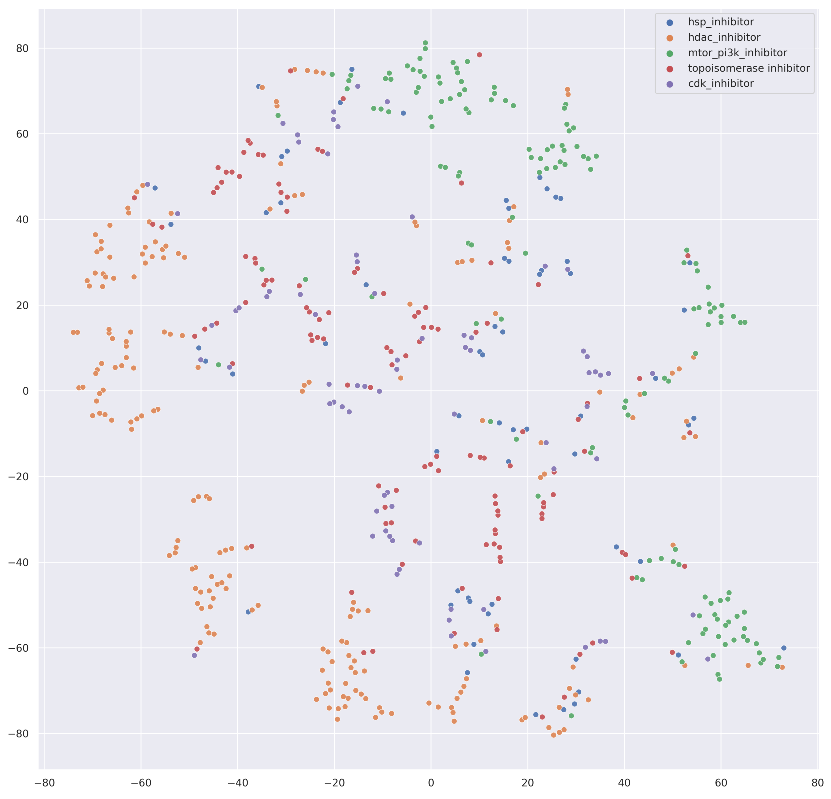

Figure 5.17: *Semi Supervised SNI Clustering Performance*

5.3 Supervised Learning Evaluation

So far, we have seen that both the supervised and semi-supervised techniques have struggled against the diffucult task of distinction between cell signaling networks

Model	Graph Type	Node Embedding Type	Submethod	Accuracy Per Signature ID	Accuracy Per Drug
Siamese GED	Unweighted	GO Term Embeddings	k-NN	0.269	0.375
Siamese GED	Unweighted	GO Term Embeddings	XGBoost	0.291	0.375
Graph2Vec	Unweighted	-	XGBoost	0.3007	0.4375
Graph2Vec	Weighted	-	XGBoost	0.3104	0.3125
SN Infomax	Unweighted		k-NN	0.29	0.375
SN Transformer	Weighted	SeqVeq Embeddings	-	**0.41**	**0.625**

Table 5.1: *Evaluation results using multiple methods and models. Each model was trained on the same validation and test set. The results correspond to the mutual test set. The available model architectures are presented in the corresponding Supplamentary Material section. No hyperparameter optimization has been performed so far. Note that the accuracy for the SN transformer was calculated by removing labelled graphs with less that three labelled instances.*

based on their corresponding mechanism of action. The attempt to train a supervised model explicitly in terms of learning between single mechanisms of action seems both futile and ,in a sense, wrong.

The futility of this practice lies mainly in the scarcity of labelled data throughout the signaling network dataset. Recall that out of the 7788 unique signatures, only 2733 of them are labelled and considering that a validation and test data split should be used, the available training data is reduced even further. The obvious solution is to use the unweighted data , but due to their -even sometimes complete- similarity to the weighted ones, they do not introduce much more variance into the input data, thus not substantially increasing performance. This can also be validated by the results of Table 5.1.

Secondly, the task itself is in a sense flawed. Take for example the already mentioned case between the mTOR and PI3K inhibitor mechanisms of action. In this setting of the supervised case, pushing the model to explicitly chose between two mechanisms of action that may both be actually correct, is inherently wrong from both a biological as well as a data analytics standpoint.

Chapter 6

Case Studies

Evaluating the quality of the embeddings was a major and extremely important part of this thesis. If the models were not able to distinguish between signaling networks with different biological and structural features, then there would be no point in further developing the methods described in the previous sections. Yet, such relatively simple tasks are not the main concern of predictive models. We need to be able not only to distinguish between signaling networks, but also get the appropriate explanation from the model as to why this distinction was made.

Saliency methods as those described in Section 4.6 offer a baseline approach in assessing the true quality of the embeddings. For example, if we were to analyze the effect of a drug that was applied on a certain cell line with a specific dosage, provided that an appropriately trained and evaluated model was used, a saliency method would provide us with the most important nodes(proteins) that affected the selected loss function. If our models are actually useful, they have to be able to identify important pathway features that can be validated by existing literature and research. In the following sections, we study specific cases stemming from the input dataset in order to determine whether the models are able to infer useful biological and structural properties of selected signaling networks.

6.1 Experimental Setup

Throughout the following case studies, we decide to use the Signaling Network Infomax models, trained in the Masked Unsupervised scheme. Since the labels are somehow "dangerous" due to their grouping. Instead, the unsupervised labels include only the bias of masking duplicate experiments in each batch, which is something desirable. We employ two models with the exact same architecture used to evaluate the quality of the embeddings, one trained with GO terms as node features and the other with the SeqVeq amino acid sequence embeddings. Each model used embeddings with a dimension of 512 and the output global embedding vector has a size of 1024 real valued features. Both models were trained for 5 epochs with a batch size of 32 graphs, and were validated in the SST, DST and MAD tasks in order to ensure their validity.

Each evaluation was based on the global summary vectors of selected experiments. For each 1024 embedding dimension vector, we calculate the attribution score in each level of representation for 10 different graphs coming from the same signature identification, since the Saliency method is generally faster in calculating such quantities. Lastly, due to the large amount of different labels (255 different mechanisms of action), even after grouping them in categories with mutual biological features, the following case studies cannot possibly contain each and every one of them. Since every mechanism of action involves a number of different drugs, we select a subset of signature IDs to evaluate our methods on. We concern ourselves with two different categories:

HDAC Inhibitors Histone deacetylase inhibitors are chemical compounds that inhibit histone deacetylases, which are a class of enzymes that operate by removing acetyl groups from histones and other protein regulatory factors, with functional consequences on chromatin remodeling and gene expression profiles[35]. We select this mechanism of action as, based on our embedding evaluation results, was the most easily clustered, and we expect the model to provide reasonable intepretations.

ATP synthesis inhibitors Experiments that mainly concern the double motor enzyme ATP Synthase, participating not only in ATP synthesis but also in ATP hydrolysis-dependent processes and in the regulation of a proton gradient across some membrane-dependent systems. In Section 5.1, these labels are the red ones in each plot which seem to generally cluster well together.

6.2 HDAC Inhibitors

In this setup, our aim is to first determine whether the initial starting point for each network, i.e. the initial node embeddings, exhibits differences in the aspect of node importance per graph. We sample two different experiments from the HDAC inhibitor labelled data:

Signature ID	**Cell Line**	**Primary Site**	**Drug Name**	**# Graphs**
PCLB003_PC3_24H_BRD-K77908580-001-04-7_10	PC3	Prostate	Entinostat	10
BRAF001_HEK293T_24H_BRD-K81418486-001-15-2_10	HEK293T	Kidney	Vorinostat	10

For the 20 unweighted graphs that concern Table 6.1, we obtain 20 global embedding vectors of size 1024. Each node gradient attribution was calculated for each one of the 1024 dimensions of the 20 global embeddings , a procedure that is being deployed in order to account for the volatility of the gradients that are calculated using the saliency method as well as the variability of each unweighted graph stemming from the same experiment. We collect the top 20 nodes that concern the global embeddings. The aforementioned attribution score histograms are shown in the following figures.

6.2.1 PC3 Entinostat

For the first sampled signature ID, we present the three different mean attribution plots along with the most important nodes calculated.

While figures 6.1 and 6.2 which concern the SeqVeq and the GO terms respec-

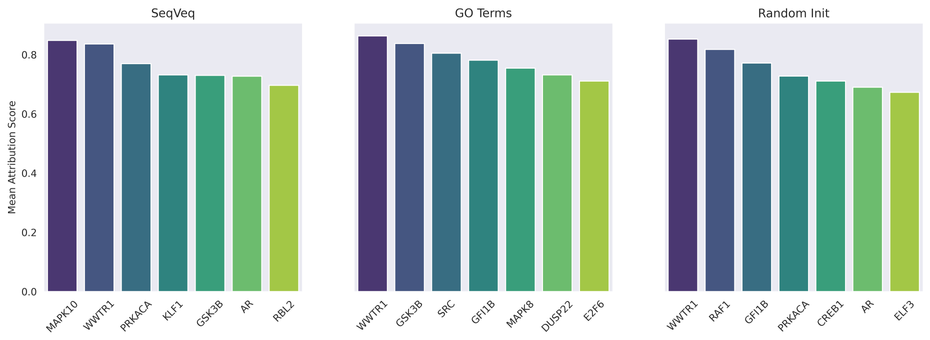

Figure 6.1: *PC3-Entinostat six most important nodes per model initial embedding type, based on their mean attributions to each dimension of the global embeddings.*

tively, seem to generally agree on the major importance of proteins like MAPK10, WWTR1, GSK3B or PRKACA, the randomly initialized model seems to follow a slightly different approach, that of attributing high importance to RAF1 but still including both WWTR1 and PRKACA in the top five most important genes. In instance, MAPKs (Mitogen Activated Protein Kinases) seem to play a very important role in prostate cancer [45]

The AR (Androgen Receptor), who in all three cases except the GO term model has a place in the top seven most important proteins and is also a succesor(is signaled by MAPK8 in each of the directed graphs) of the MAPK8 kinase is also reported to play pivotal roles in prostate cancer[17]. Transcriptional activity of the androgen receptor (AR) is crucial for growth and survival of prostate cancer and therapies aim at suppressing the transcriptional activity of the AR gene[60].

6.2.2 HEK 293T Vorinostat

HEK 293T is a human cell line, derived from the HEK 293 cell line, that expresses a mutant version of the SV40 large T antigen, stemming from human embryonic kidney 293 cells. The signaling networks that are created due to the effect of the drug vorinostat, along with the three different models, attribute the most importance to the proteins in the following figures. Note that we now selected the 5 most important nodes per case.

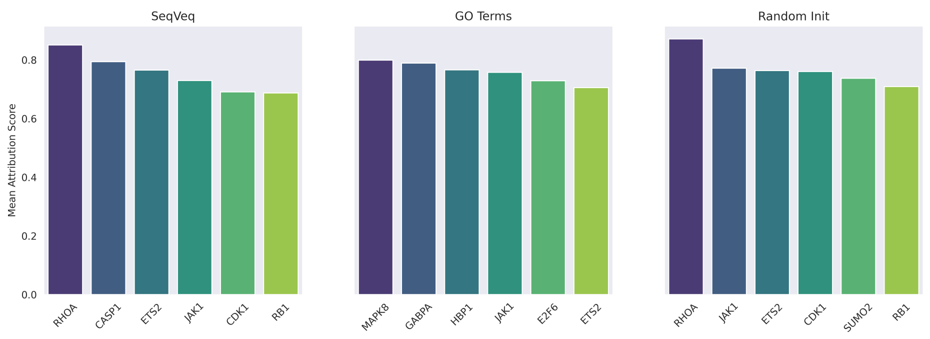

Figure 6.2: *HEK 293T-Entinostat six most important nodes per model initial embedding type, based on their mean attributions to each dimension of the global embeddings.*

While in the PC3 signature ID there was a general agreement between the utmost important nodes of each unweighted graph, this does not hold at such a level regarding the HEK important genes of Figure 6.2. The interesting aspect is the agreement between the SeqVeq and the randomly initialized model, both attributing the most importance to the RHOA, ETS2, JAK1, CDK1, RB1 with the only difference being the CASP1-SUMO2 pair.

Nevertheless, while the GO Terms model embeddings were generally inforced by the GABPA and again, MAPK8 genes, the presence of JAK1 and ETS2 seems to enforce every model's embedding decision.

6.3 ATP Synthesis Inhibitors

For the ATP synthesis inhibitor(ATPSI) labelled graphs, we decided to follow a different approach. Upon noticing the generally tight clustering amongst the ATPSI labelled data, we hypothesized that it would be beneficial if one would try to identify the most important proteins involing all the experiments with the aforementioned label.

The ATPSI labels involve 49 unique signature IDs, from experiments performed using ten different drugs, including **antimycin-a**, **digitoxin**, **bafilomycin** and other known drugs that have a reported mechanism of action "ATP Synthase Inhibitor".

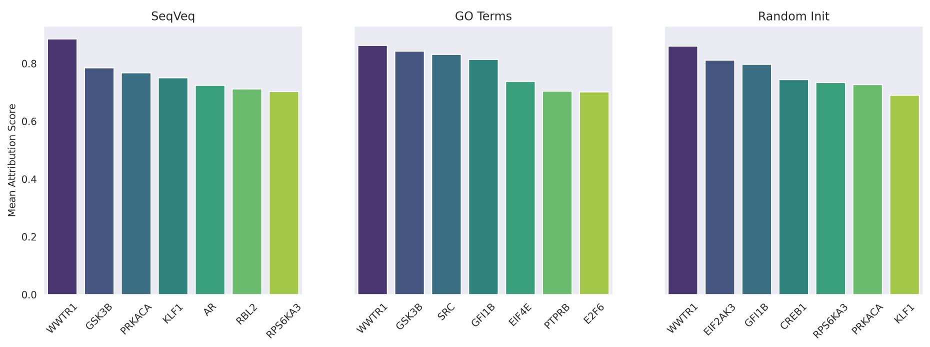

Figure 6.3: *ATPSI seven most important nodes per model initial embedding type, based on their mean attributions to each dimension of the global embeddings.*

The last two models seem to be heavily impacted by the WWTR1 protein. WWTR1, often referred to as TAZ, stands for WW domain-containing transcription regulator protein 1. It is a transcriptional coactivator which acts as a downstream regulatory target in the Hippo signaling pathway that plays a pivotal role in organ size control and tumor suppression by restricting proliferation and promoting apoptosis.

Chapter 7

Conclusion

7.1 Discussion and Limitations

As mentioned, approaches that combine GNNs and signaling networks have hitherto not, to the best of our knowledge, been tested or developed. We consider such biological data to be of paramount importance due to their increased interpretability.

The goal of our method is to "reduce" the size of biological signaling networks to a single representation, with a specific place in a multi-dimensional euclidean space. When an experiment is performed and the corresponding signaling network is produced, our methods can be used to identify the drug's potential mechanisms of action, usually in an unsupervised way, in comparison to the representations available in the existing training library. Our methods were evaluated in three different tasks, each one with a specific purpose. The SSTs, DSTs and MAD tasks validated the use of each proposed neural network architecture in accordance to the proposed general use. All proposed models passed the test, with a variable degree of success, distinguishing among the Graph2Vec and Signaling Network Infomax models.

However, the limited amount and the questionable biological completeness and validity of the data used, did not allow for a high degree of predictive confidence in further evaluation tests. Supervised methods face the barrier of insufficient and unbalanced training and validation data, and important node identification in an unsupervised manner may succumb to pure structural components rather than an accurate and interpretable biological meaning. The task itself as well, implies the simultaneous existance of multiple mechanisms of action per drug projecting a widely known and extremely challenging issue in the field of Systems Pharmacology.

Even the protein importance calculation methods suffer from an obvious flaw. Unlike image classification tasks, were one can validate if the model spotted the correct pixels in an image to make a prediction, the case with such data is nothing similar. For a single experiment, even the extensive biological public research is not always helpful in evaluating the quality of a representation..

However, the epistemic uncertainty induced by each model is insignificant compared to the aleatoric uncertainty that depends on the input data. A major issue with our input signaling networks is that rely on the public Protein to Protein Interaction network, public GEx data and the hyperparameters of CARNIVAL. Even with the best and most complete graph neural network architectures, if the input data are incomplete, the results will be even more incomplete, as each method starting from the initial experiment to the output of a neural network includes its own degree of uncertainty. Also, CARNIVAL, while being a great and potent tool, is not designed for large amounts of signaling networks. Even with a relatively small amount of unique input GEx data - since about 8000 data points are insignificant compared to the usual datasets in the Big Data era- the amount of time needed to extract the signaling networks reached almost a month.

7.2 Future Work

Despite the limitations that arise from such challenging type of data, we consider the methods described in this thesis as adequate for the baseline evaluation of each model's representations of the input data. However there is still a lot of room for improvement, both from a Deep Learning/Data Analysis as well as from a biological interpetability standpoint.

7.2.1 On improving Deep Learning performance

Base models included a modification of the original Transformer model in order to fit into the graph learning scheme. This architecture can be further improved by trying to make the positional encoding architecture more computationally efficient as well as more appropriate for dealing with DAGs. Furthermore, it is essential to perform hyperparameter tuning for the Infomax models and probably dispose of the Graph Autoencoder since mutual information or constrastive methods in general seem to consistently outperform it.

Mutual Information is also a major part that requires essential modifications. Lower bounds on KL divergence such as the Donsker-Varadhan LB or estimators like the Jensen-Shannon divergence suffer from various statistical limitations [34] and are heaviliy dependent on the choice of architecture of the discriminator. So, one has to establish better lower bounds on MI in order to improve the estimations and thus model performance. Such MI estimators include Contrastive Predicting Coding [40] or even other metrics for constrastive learning such as the newly proposed *predictive V-information* [61].

7.2.2 On improving interpretability

So far, the gradient attribution methods used in this thesis may up to a certain point increase the intepretability of the model in terms of node importance, but they are unfortunately based on the global summary vectors instead of a more appropriate metric. Using estimators other than MI, such as BCE loss if the task was supervised, would definitely increase the interpetational capabilites of the whole procedure.

Furthermore, apart from extensive literature research we need to first establish, with appropriate certainty, that the representations of each model are based both on the biologically infused features of each node as well as the structure, without those two being mutually exclusive. Consider the case of Graph2Vec for example, which only relies on the structure of each network and is still succesful in contrast to the SeqVeq or GO term embeddings which are embedded with previous biological knowledge. Trying to interpret each model, as seen in Chapter 6, may end up in different explanations that are highly dependant on the input data.

Bibliography

[1] DOI: 10.1371/journal.pone.0021977.g005.

[2] Jorg Rahnenfuhrer Adrian Alexa. *topGO*. 2017. DOI: 10.18129/B9.BIOC.TOPGO.

[3] B. Alberts et al. *Molecular Biology of the Cell 4th Edition: International Student Edition*. Routledge, 2002. ISBN: 9780815332886. URL: https://books.google.gr/books?id=ozigkQEACAAJ.

[4] Michael Ashburner et al. "Gene Ontology: tool for the unification of biology". In: *Nature Genetics* 25.1 (May 2000), pp. 25–29. DOI: 10.1038/75556.

[5] Evren U. Azeloglu and Ravi Iyengar. "Signaling Networks: Information Flow, Computation, and Decision Making". In: *Cold Spring Harbor Perspectives in Biology* 7.4 (Apr. 2015), a005934. DOI: 10.1101/cshperspect.a005934.

[6] Lejla Batina et al. "Mutual Information Analysis: a Comprehensive Study". In: *Journal of Cryptology* 24.2 (2011), pp. 269–291. ISSN: 1432-1378. DOI: 10.1007/s00145-010-9084-8. URL: https://doi.org/10.1007/s00145-010-9084-8.

[7] Mohamed Ishmael Belghazi et al. *MINE: Mutual Information Neural Estimation*. 2018. arXiv: 1801.04062 [cs.LG].

[8] T. Brosch et al. "Deep learning of brain images and its application to multiple sclerosis". In: *Machine Learning and Medical Imaging*. Elsevier, 2016, pp. 69–96. DOI: 10.1016/b978-0-12-804076-8.00003-7.

[9] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[10] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: http://doi.acm.org/10.1145/2939672.2939785.

[11] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[12] Matthias Fey and Jan E. Lenssen. "Fast Graph Representation Learning with PyTorch Geometric". In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 2019.

[13] Valeria Fionda. “Networks in Biology”. In: *Encyclopedia of Bioinformatics and Computational Biology*. Elsevier, 2019, pp. 915–921. DOI: 10.1016/b978-0-12-809633-8.20420-2.

[14] Nikolaus Fortelny and Christoph Bock. “Knowledge-primed neural networks enable biologically interpretable deep learning on single-cell sequencing data”. In: *Genome Biology* 21.1 (Aug. 2020). DOI: 10.1186/s13059-020-02100-5.

[15] C. Fotis et al. *DeepSIBA: Chemical Structure-based Inference of Biological Alterations*. 2020. arXiv: 2004.01028 [q-bio.QM].

[16] Chris Fotis et al. “Network-based technologies for early drug discovery.” eng. In: *Drug discovery today* 23 (3 Mar. 2018), pp. 626–635.

[17] Kazutoshi Fujita and Norio Nonomura. “Role of Androgen Receptor in Prostate Cancer: A Review”. In: *The World Journal of Men’s Health* 37.3 (2019), p. 288. DOI: 10.5534/wjmh.180040.

[18] Luz Garcia-Alonso et al. “Benchmark and integration of resources for the estimation of human transcription factor activities”. In: *Genome Research* 29.8 (July 2019), pp. 1363–1375. DOI: 10.1101/gr.240663.118.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[20] Sorin Grigorescu et al. “A survey of deep learning techniques for autonomous driving”. In: *Journal of Field Robotics* 37.3 (Apr. 2020), pp. 362–386. DOI: 10.1002/rob.21918.

[21] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[22] Michael Heinzinger et al. “Modeling aspects of the language of life through transfer-learning protein sequences”. In: *BMC Bioinformatics* 20.1 (Dec. 2019). DOI: 10.1186/s12859-019-3220-8.

[23] Petra Hillmann and Doriano Fabbro. “PI3K/mTOR Pathway Inhibition: Opportunities in Oncology and Rare Genetic Diseases”. In: *International Journal of Molecular Sciences* 20.22 (Nov. 2019), p. 5792. DOI: 10.3390/ijms20225792.

[24] R Devon Hjelm et al. *Learning deep representations by mutual information estimation and maximization*. 2018. arXiv: 1808.06670 [stat.ML].

[25] F. Jordan, T.-P. Nguyen, and W.-c. Liu. “Studying protein-protein interaction networks: a systems view on diseases”. In: *Briefings in Functional Genomics* 11.6 (Aug. 2012), pp. 497–504. DOI: 10.1093/bfgp/els035.

[26] J.Dedrick Jordan, Emmanuel M Landau, and Ravi Iyengar. “Signaling Networks”. In: *Cell* 103.2 (Oct. 2000), pp. 193–200. DOI: 10.1016/s0092-8674(00)00112-4.

[27] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2016. arXiv: 1609.02907 [cs.LG].

[28] Mark A. Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AIChE Journal* 37.2 (Feb. 1991), pp. 233–243. DOI: 10.1002/aic.690370209.

[29] Quoc Le and Tomas Mikolov. “Distributed Representations of Sentences and Documents”. In: *Proceedings of the 31st International Conference on Interna-*

tional Conference on Machine Learning - Volume 32. ICML'14. Beijing, China: JMLR.org, 2014, II–1188–II–1196.

[30] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. *Self-Attention Graph Pooling*. 2019. arXiv: 1904.08082 [cs.LG].

[31] Junying Li, Deng Cai, and Xiaofei He. *Learning Graph-Level Representation for Drug Discovery*. 2017. arXiv: 1709.03741 [cs.LG].

[32] Anika Liu et al. "From expression footprints to causal pathways: contextualizing large signaling networks with CARNIVAL". In: *npj Systems Biology and Applications* (2019). DOI: 10.1038/s41540-019-0118-z.

[33] Qi Liu et al. "Constrained Graph Variational Autoencoders for Molecule Design". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 7795–7804. URL: http://papers.nips.cc/paper/8005-constrained-graph-variational-autoencoders-for-molecule-design.pdf.

[34] David McAllester and Karl Stratos. *Formal Limitations on the Measurement of Mutual Information*. 2020. arXiv: 1811.04251 [cs.IT].

[35] Giorgio Milazzo et al. "Histone Deacetylases (HDACs): Evolution, Specificity, Role in Transcriptional Complexes, and Pharmacological Actionability". In: *Genes* 11.5 (May 2020), p. 556. DOI: 10.3390/genes11050556.

[36] Erick Moen et al. "Deep learning for cellular image analysis". In: *Nature Methods* 16.12 (May 2019), pp. 1233–1246. DOI: 10.1038/s41592-019-0403-1.

[37] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. 1st ed. Adaptive Computation and Machine Learning. The MIT Press, 2012.

[38] Annamalai Narayanan et al. *graph2vec: Learning Distributed Representations of Graphs*. 2017. arXiv: 1707.05005 [cs.AI].

[39] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. *f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization*. 2016. arXiv: 1606.00709 [stat.ML].

[40] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. 2018. arXiv: 1807.03748 [cs.LG].

[41] Lawrence Page et al. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Previous number = SIDL-WP-1999-0120. Stanford InfoLab, Nov. 1999. URL: http://ilpubs.stanford.edu:8090/422/.

[42] Monarch Parmar et al. *NLPExplorer: Exploring the Universe of NLP Papers*. 2019. arXiv: 1910.07351 [cs.IR].

[43] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[44] Matthew E. Peters et al. *Deep contextualized word representations*. 2018. arXiv: 1802.05365 [cs.CL].

[45] Gonzalo Rodriguez-Berriguete et al. “MAP Kinases and Prostate Cancer”. In: *Journal of Signal Transduction* 2012 (2012), pp. 1–9. DOI: 10.1155/2012/169170.

[46] Guillaume Salha, Romain Hennequin, and Michalis Vazirgiannis. *Simple and Effective Graph Autoencoders with One-Hop Linear Models*. 2020. arXiv: 2001.07614 [cs.LG].

[47] Michael Schubert et al. “Perturbation-response genes reveal signaling footprints in cancer gene expression”. In: *Nature Communications* 9.1 (Jan. 2018). DOI: 10.1038/s41467-017-02391-6.

[48] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. *Self-Attention with Relative Position Representations*. 2018. arXiv: 1803.02155 [cs.CL].

[49] Greg Ver Steeg and Aram Galstyan. *Discovering Structure in High-Dimensional Data Through Correlation Explanation*. 2014. arXiv: 1406.1222 [cs.LG].

[50] Greg Ver Steeg and Aram Galstyan. *Maximally Informative Hierarchical Representations of High-Dimensional Data*. 2014. arXiv: 1410.7404 [stat.ML].

[51] Aravind Subramanian et al. “A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles”. In: *Cell* 171.6 (Nov. 2017), 1437–1452.e17. DOI: 10.1016/j.cell.2017.10.049.

[52] Fan-Yun Sun et al. *InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization*. 2020. arXiv: 1908.01000 [cs.LG].

[53] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. *Axiomatic Attribution for Deep Networks*. 2017. arXiv: 1703.01365 [cs.LG].

[54] Iman Tavassoly, Joseph Goldfarb, and Ravi Iyengar. “Systems biology primer: the basic methods and approaches”. In: *Essays in Biochemistry* 62.4 (Oct. 2018), pp. 487–500. DOI: 10.1042/ebc20180003.

[55] Ian W. Taylor and Jeffrey L. Wrana. “Protein interaction networks in medicine and disease”. In: *PROTEOMICS* 12.10 (May 2012), pp. 1706–1716. DOI: 10.1002/pmic.201100594.

[56] “The Gene Ontology Resource: 20 years and still GOing strong”. In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D330–D338. DOI: 10.1093/nar/gky1055.

[57] Dénes Türei, Tamás Korcsmáros, and Julio Saez-Rodriguez. “OmniPath: guidelines and gateway for literature-curated signaling pathway resources”. In: *Nature Methods* 13.12 (Nov. 2016), pp. 966–967. DOI: 10.1038/nmeth.4077.

[58] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[59] Petar Veličković et al. *Deep Graph Infomax*. 2018. arXiv: 1809.10341 [stat.ML].

[60] D. S. Welsbie et al. “Histone Deacetylases Are Required for Androgen Receptor Function in Hormone-Sensitive and Castrate-Resistant Prostate Cancer”. In: *Cancer Research* 69.3 (Jan. 2009), pp. 958–966. DOI: 10.1158/0008-5472.can-08-2216.

[61] Yilun Xu et al. *A Theory of Usable Information Under Computational Constraints*. 2020. arXiv: 2002.10689 [cs.LG].