International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Jan-24)

http://dx.doi.org/10.12785/ijcds/150110

GRNRI: Gene Regulatory Network Inference using Unsupervised Graph Neural Network

Fatema Abdulhameed[1], Hazem Abbas[1] and Cherif Salama[1,2]

[1]*Computer and Systems Engineering Department, Ain Shams University, Cairo, Egypt*
[2]*Computer Science and Engineering Department, The American University in Cairo, Cairo, Egypt*

Received 21 Jun. 2023, Revised 20 Oct. 2023, Accepted 16 Nov. 2023, Published 1 Jan. 2024

Abstract: The reconstruction of gene regulatory networks (GRNs) from gene expression data is a challenging problem. GRNs provide insight into the complex regulatory relationships between genes and can help improve our understanding of biological processes. However, current methods for inferring GRNs have limitations in accurately modeling these relationships. In this work, we propose GRNRI: a variational auto-encoder model that learns to infer GRNs from single-cell RNA sequencing (scRNA-seq) data in an unsupervised way. Our model is a modified version of Neural Relational Inference (NRI), a powerful framework for learning relational structure from data. We developed a version of NRI that explicitly models the regulatory relationships between genes using a variational auto-encoder. Results show that GRNRI achieves comparable or better performance on most benchmark datasets compared with state-of-the-art methods. Our work introduces a powerful tool for advancing our understanding of gene regulation and its role in biological processes.

Keywords: Gene Regulatory Network, Single Cell RNA sequencing, Graph Neural Network, Unsupervised Learning, Bioinformatics

1. Introduction

Inferring a Gene Regulatory Network (GRN) has become a very active area of research due to its importance in understanding the relationships between genes and the products they help introduce. This knowledge can help us understand the phenotype of many diseases. With the advancements in measurement technologies, a wealth of datasets has become available. As a result, computational methods can help us process these datasets and extract information that can help us infer GRNs and develop solutions accordingly.

Previously, gene expression levels in a sample were measured using Bulk RNA sequencing technology. This technology provides the average of gene expression levels of a bulk of cells together. However, today since its introduction in 2009 [1], Single cell RNA sequencing (scRNA-seq) has become widely used. This technology allows us to examine the gene expression level of each cell individually in the sample. However, experimental noise is a significant challenge in scRNA-seq data that is not found in bulk sequencing data and it might introduce biases to the data [2], [3].

ScRNA-seq data can take the form of a matrix with each value representing the gene expression level of a specific cell.

As genes represent the key to all biological processes, figuring out the interaction between them is an attracting scRNA-seq study. GRNs represent the regulatory relationships between Transcription Factors (TFs) and their target genes [3]. Each node of the GRN represents a TF or a gene, and each edge connecting a TF and a target gene represents a regulation. Given the scRNA-seq dataset, the task we are solving is to find out all possible regulations among genes to reconstruct the GRN. Several methods [4], [5], [6], [7], [8], [9], [10], [11] attempted to solve this task, but most of them suffered from the drawback of low accuracy. Some of the methods used to study Gene Regulatory Networks rely on measurements that often necessitate intricate experimental designs. These measurements may also introduce additional noise into the data since they come from different experiments. Furthermore, many of these methods, particularly those that are based solely on scRNA-seq data, utilize statistical algorithms that focus on co-expression networks. However, these algorithms have their limitations.

In this paper, we propose GRNRI to solve the GRN inference task. GRNRI is an unsupervised model structured as a variational auto-encoder that uses Graph Neural Networks (GNNs) in both its encoder and decoder functions relying on scRNA-seq data only without requiring any additional measurements. Examining the structure of the GRNRI model allows us to understand how various genes

E-mails: fatema.mahmoud97@gmail.com, hazem.abbas@eng.asu.edu.eg, cherif.salama@aucegypt.edu

interact to influence the expression levels of individual genes.

GRNRI is an unsupervised model that is based on the Neural Relational Inference (NRI) model [12]. NRI model is a powerful framework that tries to learn the relational structure from observed data and it is a variational auto-encoder that is based on graph neural network.

The main contributions of this paper are:

- Proposing GRNRI, a new unsupervised model for inferring GRNs based on the NRI model.
- Leveraging the usage of scRNA-seq data as opposed to using RNA-seq data for GRN inference.
- Implementing and evaluating the proposed model on several popular benchmark datasets showing that it is able to achieve comparable or better performance with respect to state-of-the-art methods.

The rest of the paper is organized as follows: In Section 2, we briefly introduce Graph Neural Networks (GNNs), Gene Regulatory Networks (GRNs), and Single Cell RNA sequencing (scRNA-seq). Our proposed method, the GRNRI model, is explained in detail in Section 3, while our implementation and our experimental setup are presented in Section 4. Section 5 reports and discusses the obtained results. Finally. Section 6 concludes the paper reiterating over its main contributions and proposing potential extensions of this work.

2. Background

Data that has a graph structure can be processed by a neural network called a GNN. GRNs are a type of biological network that describes the interactions between genes and their regulators. ScRNA-seq is a technique that can be used to measure gene expression in individual cells. The following brief background will describe how these three fields have been used together to study gene regulation at the single-cell level using GNNs to model GRNs from scRNA-seq data.

A. Graph Neural Network

Since our proposed solution heavily relies on using GNNs [13], [14], [15], it is important to have a reasonable understanding of their operation. In this section we give a brief introduction to GNNs. GNNs operate directly on graph-structured data by passing local messages. They have been very effective to solve multiple tasks, such as classification of graphs [16], [17], [18], [19], [20], [21], classification of nodes in graphs [22], [23], modelling interacting systems [24], [25], and relational reasoning tasks [26].

In Gilmer et al. [15] approach, GNNs framework that follows a neighborhood aggregation scheme were proposed. Each node in the graph is represented by a vector that is computed by using aggregation and transformation functions on the neighboring nodes. As we are targeting to learn a representation of the gene expression data, GNNs were a good choice because it is a powerful framework that is able to work effectively on learning a representation for graph structured data. The approach is based on a "message passing neural network" framework using the Graph Nets architecture schematics. This approach was used by Kipf et al. [12] as well.

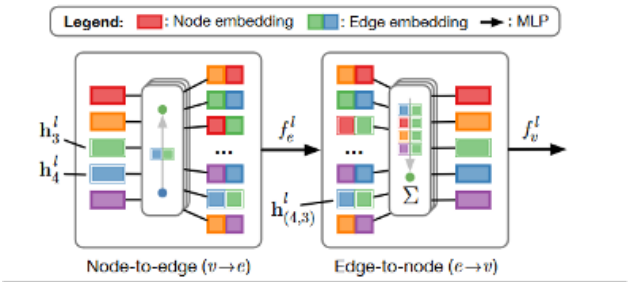

Figure 1. In a Graph Neural Network (GNN), there are two operations for moving between node and edge representations: node-to-edge ($v \to e$) and edge-to-node ($e \to v$). The $v \to e$ operation concatenates the embeddings of nodes connected by an edge, while the $e \to v$ operation aggregates the embeddings of all incoming edges. Each operation is followed by a small neural network. This figure is reproduced from [12].

More formally, in a graph represented by G = (V, E), where V represents the set of nodes and E represents the set of edges connecting the nodes, the message passing operation in a GNN is defined by the following equations and shown in Fig. 1:

$$v \to e : h_{(i,j)}^l = f_e^l([h_i^l, h_j^l, x_{(i,j)}]) \quad (1)$$

$$e \to v : h_j^{l+1} = f_v^l([\Sigma_{i \in N_j} h_{(i,j)}^l, x_j]) \quad (2)$$

In the Equation 1 and Equation 2, xi is the initial features of node v_i, $x_{(i,j)}$ is the initial features of edge $e_{(i,j)}$ connecting node v_i to node v_j. In layer l, h_i^l is the embedding of node v_i and $h_{(i,j)}^l$ is the embedding of edge $e_{(i,j)}$. N_j denotes the set of indices of neighbour nodes connected by an incoming edge. The function f_e is a neural network that is specific for the edge while f_v is specific for the node. $[\cdot, \cdot]$ represents concatenation of vectors. While Equation (1) allows for mapping from edge to node representation, Equation (2) allows for mapping from node to edge representation through multiple rounds of message passing.

Here, we use a general GNN formulation similar to the work done in [25], [15]. However, the GNN formulation used by Scarselli et al. [13] was not general enough because the node embedding $h_{(i,j)}^l$ is affected by the embedding of the sending node h_i^l and the edge type only while the embedding of the receiving node has no effect on it.

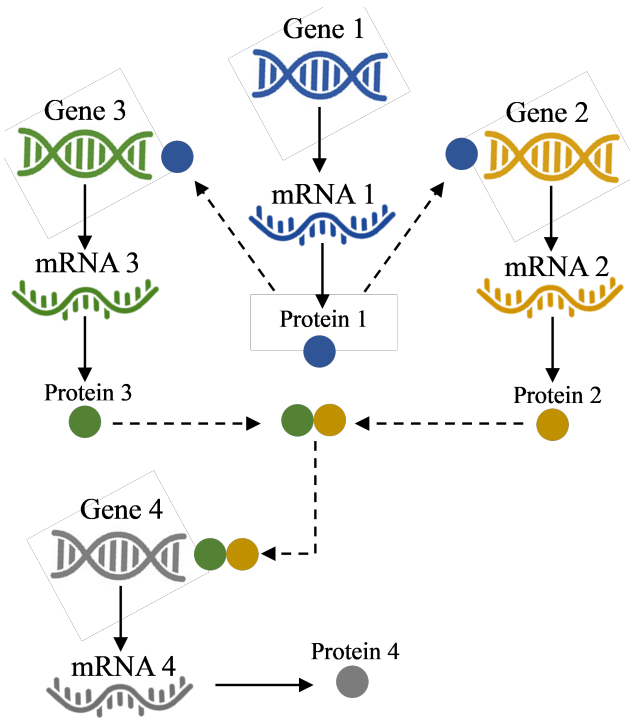

Figure 2. Gene 1 produces protein 1 which affects gene 2 and gene 3 expression. Gene 2 and gene 3 produce protein 2 and protein 3, respectively, which affect gene 4 expression.

B. Gene Regulatory Network

The DNA in our cells is the same, but the cells have different shapes and functions. For example, the genes that are active in brain cells are not the same as the ones that are active in skin cells or hair cells. A gene is a part of the DNA that has the instructions to make a protein. Some proteins help to build and run our body, while others control the activity of other genes. The way that genes and proteins affect each other is like a network of switches that decides how much mRNA and proteins are made in the cell, which affects the cell's function. This network is called a gene regulatory network.

A simple example of a GRN is shown in Fig. 2. Gene 1 produces a protein that turns on Gene 2 and Gene 3. Then, Gene 2 and Gene 3 produce Protein 2 and Protein 3 respectively. Protein 2 and Protein 3 turn on Gene 4. So, Gene 4 is now expressed and produces Protein 4. Fig. 3 shows how this process is represented by a GRN. One of the latest developments in this field is a machine-learning model called Geneformer, which has learned from a huge gene-expression data set of about 30 million single-cell profiles from many human tissues [27]. Geneformer can make guesses about gene-network biology when there is not enough gene-expression data, and can find out which genes are important for the network and which ones could be targets for treatments [27].

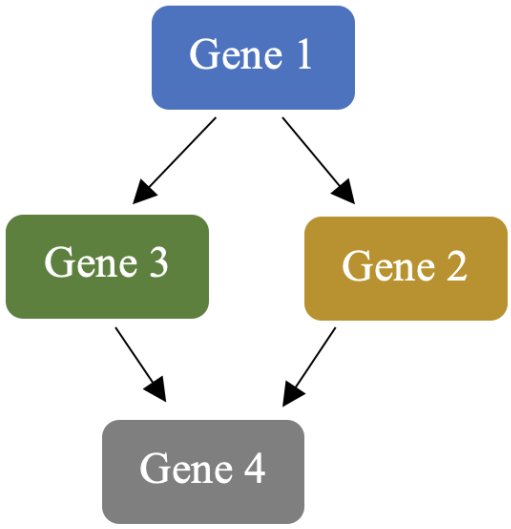

Figure 3. As GRN show the interaction between genes, directed edges between genes represent the dependency between them. Gene 2 and gene 3 depend on gene 1, and gene 4 depend on gene 2 and gene 3.

C. Single Cell RNA Sequencing

The transcriptome of cells is the set of all RNA molecules that are produced by the cells, and it reflects their gene expression and function. Researchers have used two different methods to study the transcriptome of cells: ScRNA-seq and bulk RNA sequencing. ScRNA-seq is a technique that allows researchers to examine the transcriptome of each cell individually, by isolating the cells and measuring the gene expression levels of each cell. As shown in Fig. 4, this process can reveal the diversity and heterogeneity of cells in a sample, and it has been very useful in studying cell behavior in various biological processes, such as development, differentiation, and disease [28], [29]. In contrast, bulk RNA sequencing is a technique that examines the average gene expression levels of all cells together, by pooling the RNA from many cells and sequencing it as a whole. This method can provide a global overview of the transcriptome of a sample, but it cannot capture the variation and complexity of individual cells [30]. Therefore, ScRNA-seq and bulk RNA sequencing have different advantages and limitations, and they can complement each other in providing a comprehensive understanding of the transcriptome of cells.

3. Proposed GRNRI Model

The NRI model is a type of variational autoencoder that uses graph neural networks to learn the relations and dynamics of a system based on observed dynamics [12]. It passes messages over a latent graph to jointly learn the relations and the dynamics. The task of the encoder in the model is to predict the interactions based on the observations while the task of the decoder is to learn the dynamics based

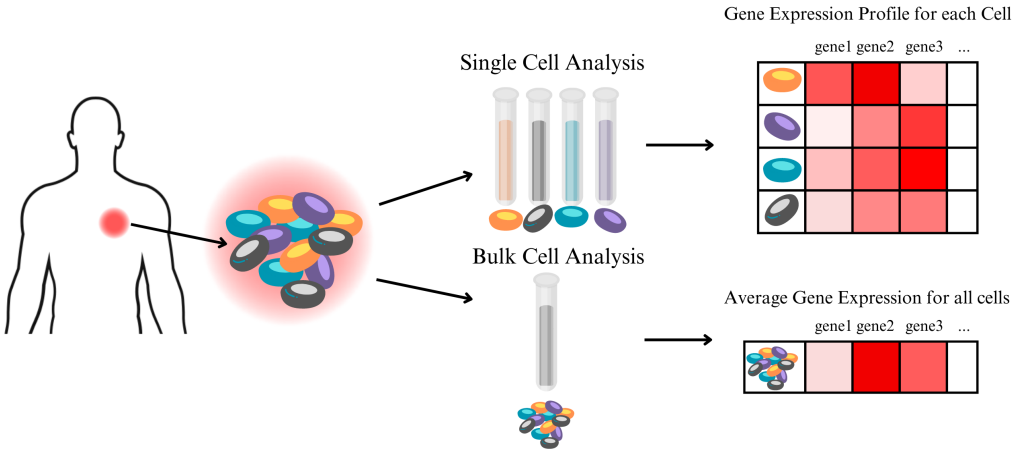

Figure 4. In this example, bulk cell analysis shows very low expression level in gene 1, very high gene expression level in gene 2, and medium gene expression level in gene 3. However, single-cell analysis reveals that each cell has its unique profile which can provide us with a lot of insight.

on the predicted graph. Both encoder and decoder parts are trained together in NRI model. GRNRI is a simplified version of NRI designed for addressing the problem of inferring a GRN that represents gene interactions in single cells by embedding gene expression data in a completely unsupervised way.

The input to our GRNRI model consists of gene expression levels of N genes for a single cell in the sample. In Fig. 4, each row in the matrix describing gene expression profile for each cell could be a sample for the input to our model before preprocessing. We denote the gene expression level of gene v_i by x_i. We denote the set of gene expression levels of all V genes for a single cell by $X = x_1, ..., x_v$. Unlike NRI, our input is not consisting of timesteps.

Our target is to predict the GRN that represents the interactions between all genes. We represent the unknown GRN by graph z, and a single interaction between genes v_i and v_j is represented by edge z_{ij}.

In our model, edges are directed and have two types representing the existence of an interaction between two genes. The first edge type is the type we use when there is no such interaction, while the second edge type is the one we use when an interaction exists. Our model gives the flexibility of adding more edge types.

GRNRI model is formally represented as a variational autoencoder. The goal is to maximize the ELBO:

$$L = \mathbb{E}_{q_\phi(z|x)}[log p_\theta(x|z)] - KL[q_\phi(z|x)||p_\theta(z)] \tag{3}$$

In Equation 3, the encoder models the factorized distribution $q_\phi(z|x)$ of z_{ij} where z_{ij} is the edge between genes v_i and v_j. All K edge types for z_{ij} are represented in a one-hot representation. The decoder models the distribution $p_\theta(x|z)$ with a GNN given the latent gene regulatory network graph structure z. The prior $p_\theta(z) = \Pi_{i\neq j} p_\theta(z_{ij})$ is a factorized uniform distribution that represents edge types.

GRNRI model is schematically shown in Fig. 5. The encoder, sampling, decoder, and training components of our model are described in detail in the following subsections.

A. Encoder

The task of the encoder is to infer the interactions z_{ij} between genes given expression data. If we don't know any information about the structure of the graph regulatory network, we shall use a fully connected GNN at first and its structure will be formed accordingly after learning. We formally model the encoder as $q_\phi(z_{ij}|x) = softmax(f_{enc}, \phi(x)_{ij,1:V})$, where $f_{enc}, \phi(x)$ is a fully connected graph neural network. We exclude self-loops because we don't have relationships between genes and themselves. Given gene expression levels for each cell, multiple message passing operations are computed by the encoder to get information from neighbourhood nodes as follows:

$$h_j^1 = f_{emb}(x_j) \tag{4}$$

$$v \to e : h_{(i,j)}^1 = f_e^1([h_i^1, h_j^1]) \tag{5}$$

$$e \to v : h_j^2 = f_v^1(\Sigma_{i\neq j} h_{(i,j)}^1) \tag{6}$$

$$v \to e : h_{(i,j)}^2 = f_e^2([h_i^2, h_j^2]) \tag{7}$$

The posterior $q_\phi(z_{ij}|x)$ shown in ELBO equation rep-

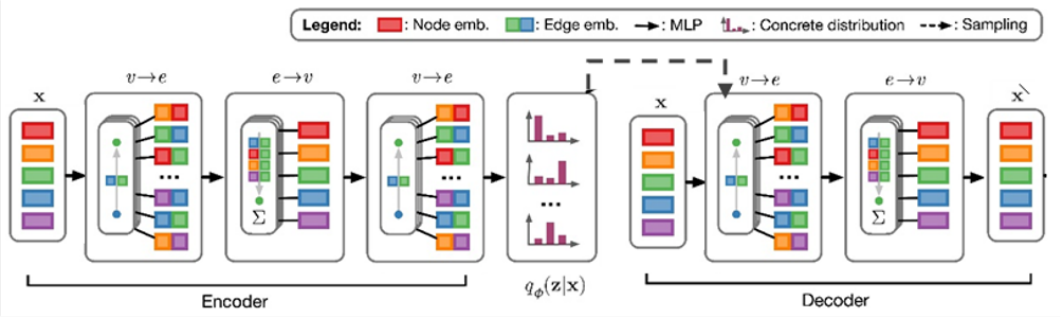

Figure 5. In GRNRI model, the encoder task is to predict a probability distribution $q_\phi(z|x)$ over latent interactions based on gene expression data while the decoder task is to convert the hidden representation back into gene expression values. Both parts of GRNRI use GNNs with multiple rounds of node-to-edge ($v \rightarrow e$) and edge-to-node ($e \rightarrow v$) message passing as their inner structure. This figure is adapted from [12].

resents the posterior of each edge and we model it by a $softmax(h^2_{(i,j)})$ function. The letter ϕ shown in the posterior represents all the parameters of all the networks shown in Eqs.(4- 7). In Eqs. 4 and 5, the embedding of the edge between genes v_i and v_j that is represented by $h^1_{(i,j)}$ depends only on gene expression data x_i and x_j ignoring any information from farther nodes. So, we used multiple message passing operations framework to get more information from neighbours of nodes that is represented in Eqs. 6 and 7. All used functions $f_{(...)}$ are 64 fully-connected Multi-Layer Perceptrons (MLPs).

An edge type posterior $q_\phi(z_{ij}|x)$ represents the probability of edge z_{ij} existence between genes v_i and v_j. If the probability is high, it means that there is a relation between both genes and gene v_i is a transcription factor that affects the transcription of gene v_j. Otherwise, it means that there is no relation between both genes v_i and v_j.. Accordingly, we can form and evaluate the GRN that represents the interactions between all G genes.

B. Sampling

A recent approach to overcome the challenge of using the reparameterization trick to backpropagate through sampling for discrete latent variables is to form a continuous approximation from the discrete values we have in the distribution and take samples accordingly. We use the reparameterization trick to get biased gradients from the approximation [31]. Each edge z_{ij} is sampled as follows:

$$z_{ij} = softmax((h^2_{(i,j)} + v)/\tau) \tag{8}$$

We use a Gumbel (0, 1) distribution to draw independent and identically distributed real values samples forming vector v. The smoothness of the samples can be controlled by the softmax temperature τ. We can get a very smooth samples by setting a high value for τ and a one-hot samples by setting τ approaches to 0.

C. Decoder

The task of the decoder is to transform the hidden representation back to the gene expression values. After using K neural networks for K edge types, we formally model the decoder as following:

$$v \rightarrow e : \tilde{h}_{(i,j)} = \Sigma_k z_{ij,k} \tilde{f}^k_e([x_i, x_j]) \tag{9}$$

$$e \rightarrow v : \mu_j = \tilde{f}_v(\Sigma_{i \neq j} \tilde{h}_{(i,j)}) \tag{10}$$

$$p(\grave{x}_j|z) = N(\mu_j, \sigma^2 I) \tag{11}$$

where z_{ij} is a vector and $z_{ij,k}$ is the k-th element in the vector. When $z_{ij,k}$ is a discrete one-hot sample, the messages $\tilde{h}_{(i,j)}$ are $\tilde{f}^k_e([x_i, x_j])$ for the selected edge type k, and we get a weighted sum for the continuous relaxation. We put a fixed value for the variance σ^2.

D. Training

After outlining all the components of GRNRI, the training process is as follows: For a training example x (gene expression levels of scRNA-seq data), the encoder is run to compute $q_\phi(z_{ij}|x)$. Then, a vector z_{ij} is sampled from the reparameterizable approximation of $q_\phi(z_{ij}|x)$.The decoder is then run to compute μ. The reconstruction error $\mathbb{E}_{q_\phi(z|x)}[log p_\phi(x|z)]$ is the first term in the ELBO equation is calculated as follows:

$$-\Sigma_j ||x_j - \mu_j||^2/(2\sigma^2) + const \tag{12}$$

while the KL divergence $KL[q_\phi(z|x)||p_\theta(z)]$ is the second term and is calculated as the sum of entropies, with the addition of a constant if necessary:

$$\Sigma_{i \neq j} H(q_\phi(z_{ij}|x)) + const \tag{13}$$

And finally, we can optimize our model and compute the gradients by backpropagating.

E. Used Hardware

The specification of the hardware used for the project are the following: an 11th Generation Octacore Intel Core i9 with 16 MB Cache, 64GB DDR4, and an NVIDIA GeForce RTX3090 GPU with 24GB GDDR6X.

4. Implementation and Experimental Setup

The proposed model was implemented using Python and its source code made publicly available in the following repository:

https://github.com/FatemaMahmoud/GRNRI

On the other hand, the following datasets were analyzed during this study and are publicly available: mESC dataset [32], mDC dataset [33], mHSC-E [34], mHSC-GM [34], mHSC-L [34] for mouse datasets and hHEP dataset [35], hESC dataset [36] for human datasets.

To assess the performance of GRNRI, we used the BEELINE framework [37], which collected four types of ground-truth networks and seven scRNA-seq datasets from sources including mESC [32], mDC [33], mHSC-E [34], mHSC-GM [34], mHSC-L [34], hHEP [35] and hESC [36] datasets. We followed the same preprocessing strategy for raw gene expression data as in the BEELINE framework [37]. For each dataset, we considered only highly variable TFs and the top 500 most-varying genes, following the BEELINE framework. For each dataset, number of transcription factor, number of genes, and density of the edges for each ground truth network are represented in table I. We did not compute results for the 1000 most-varying genes due to computational limitations.

The following subsections describe the datasets in details and the used evaluation metric:

A. Datasets

1) mESC single cell RNA-seq dataset is from mouse embryonic stem cells (mESCs) at different stages of development. It contains information about the cell cycle phases, chromatin accessibility, and transcription factors of mESCs. The dataset can be used to study the molecular mechanisms of cell fate determination, differentiation, and reprogramming in mESCs.
2) mDC single cell RNA-seq dataset is from mouse dendritic cells (mDCs) under various conditions. It contains information about the cell types, stimulation conditions, and time points of mDCs. The dataset can be used to study the molecular mechanisms of immune response, inflammation, and infection in mDCs.
3) mHS-E single cell RNA-seq dataset is from mouse hematopoietic stem and progenitor cells (mHSPCs) enriched for erythroid progenitors. It contains information about the cell types, developmental stages, and transcriptional regulators of mHSPCs. The dataset can be used to study the molecular mechanisms of erythropoiesis, the process of red blood cell production, in mice.
4) mHSC-GM single cell RNA-seq dataset is from mouse hematopoietic stem and progenitor cells (mHSPCs) enriched for granulocyte-monocyte progenitors. It contains information about the cell types, developmental stages, and transcriptional regulators of mHSPCs. The dataset can be used to study the molecular mechanisms of granulopoiesis and monopoiesis, the processes of granulocyte and monocyte production, in mice.
5) mHSC-L single cell RNA-seq dataset is from mouse hematopoietic stem and progenitor cells (mHSPCs) enriched for lymphoid progenitors. It contains information about the cell types, developmental stages, and transcriptional regulators of mHSPCs. The dataset can be used to study the molecular mechanisms of lymphopoiesis, the process of lymphocyte production, in mice.
6) hHEP single cell RNA-seq dataset is from human hepatic progenitor cells (hHEPs) and their differentiated derivatives. The dataset contains information about the cell types, developmental stages, and transcriptional regulators of hHEPs. The dataset can be used to study the molecular mechanisms of liver development, regeneration, and disease in humans.
7) hESC single cell RNA-seq dataset is from human embryonic stem cells (hESCs) at different stages of differentiation. It contains information about the cell types, developmental stages, and transcriptional regulators of hESCs and their derivatives. The dataset can be used to study the molecular mechanisms of pluripotency, lineage specification, and cell fate determination in hESCs.

B. Evaluation Metric

We evaluated the performance using the Early precision ratio (EPR) which is a metric used by BEELINE to evaluate the performance of gene regulatory network inference algorithms from single-cell transcriptomic data. EPR measures how many of the top-ranked edges predicted by an algorithm are true positives, compared to a random predictor. EPR is calculated by dividing the early precision score, which is the fraction of true positives in the top-k edges, by the expected precision of a random predictor, which is the edge density of the ground truth network. An EPR value of one means that an algorithm has the same early precision as a random predictor, while an EPR value greater than one means that an algorithm has a higher early precision than a random predictor. EPR is useful for assessing the accuracy of algorithms in recovering the most important interactions in a network.

5. Obtained Results

We evaluated GRNRI against seven baseline algorithms: DeepSEM [4], GENIE3 [5], PIDC [6], SCODE [7], SINCERITIES [8], GRNBoost2 [9], ppcor [11]. According to

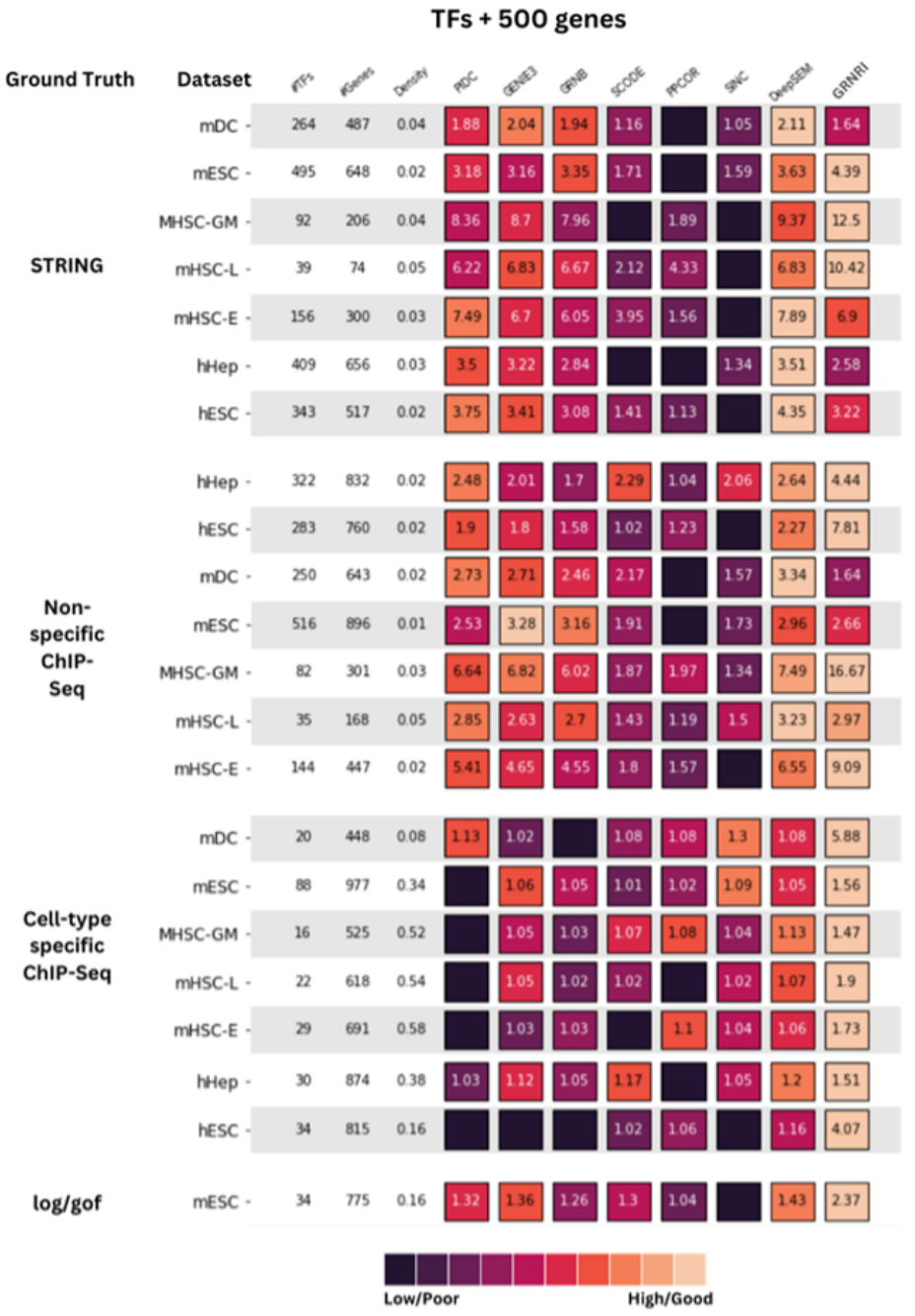

Figure 6. Performance was evaluated using EPR on seven datasets with four different ground-truth networks using the 500 most-varying genes and all varying TFs. For each dataset, the color scale ranges from 0 to 1 based on the minimum and maximum EPR values. Black squares indicate performance worse than random predictors. The first three columns represent number of transcription factors, number of genes, and density of the edges, respectively, in the ground truth network after following the preprocessing done by BEELINE framework. The rest of the columns are the names of the evaluating algortihms. The first seven rows represent the results calculated while using STRING as ground truth network, the following seven rows use Non-specific ChIP-Seq as ground truth network, the following seven rows use Cell-type specific ChIP-Seq and the last row uses log/gof network. The figure shows a full comparison between GRNRI and state-of-the-art algorithms.

TABLE I. The table shows for each dataset the number of transcription factors, number of genes, and density of the edges for each ground truth network.

		Ground Truth Networks TFs + 500 genes											
		STRING			Non-specific ChIP-Seq			Cell-type specific ChIP-Seq			Iog/gof		
		#TFs	#Genes	Density	#TFs	#Genes	Density	#TFs	#Genes	Density	#TFs	#Genes	Density
Datasets	mESC	495	648	0.02	516	896	0.01	88	977	0.34	34	775	0.16
	mDC	264	487	0.04	250	643	0.02	20	448	0.08			
	mHSC-E	156	300	0.03	144	447	0.02	29	691	0.58			
	mHSC-GM	92	206	0.04	82	301	0.03	16	525	0.52			
	mHSC-L	39	74	0.05	35	168	0.05	22	618	0.54			
	hHEP	409	656	0.03	409	656	0.03	322	832	0.02			
	hESC	343	517	0.02	283	760	0.02	283	760	0.02			

TABLE II. According to BEELINE assessment and the setting used in table I, the results of state-of-the-art algorithms are shown in the table.

		Ground Truth Networks																											
		STRING							Non-specific ChIP-Seq							Cell-type specific ChIP-Seq							Iog/gof						
	Methods	PIDC	GENIE3	GRNBOOST2	SCODE	PPCOR	SINCERITIES	DeepSEM	PIDC	GENIE3	GRNBOOST2	SCODE	PPCOR	SINCERITIES	DeepSEM	PIDC	GENIE3	GRNBOOST2	SCODE	PPCOR	SINCERITIES	DeepSEM	PIDC	GENIE3	GRNBOOST2	SCODE	PPCOR	SINCERITIES	DeepSEM
Datasets	mESC	3.18	3.16	3.35	1.71	0.88	1.59	3.63	2.53	3.28	3.16	1.91	0.83	1.73	2.96	1	1.06	1.05	1.01	1.02	1.09	1.05	1.32	1.36	1.26	1.3	1.04	0.84	1.43
	mDC	1.88	2.04	1.94	1.16	0.91	1.05	2.11	2.73	2.71	2.46	2.17	0.99	1.57	3.34	1.13	1.02	0.91	1.08	1.08	1.3	1.08							
	mHSC-E	7.49	6.7	6.05	3.95	1.56	0.74	7.89	5.41	4.65	4.55	1.8	1.57	0.95	6.55	0.98	1.03	1.03	0.99	1.1	1.04	1.06							
	mHSC-GM	8.36	8.7	7.96	0.91	1.89	0.4	9.37	6.64	6.82	6.02	1.87	1.97	1.34	7.49	0.97	1.05	1.03	1.07	1.08	1.04	1.13							
	mHSC-L	6.22	6.83	6.67	2.12	4.33	0.76	6.83	2.85	2.63	2.7	1.43	1.19	1.5	3.23	0.98	1.05	1.02	1.02	0.99	1.02	1.07							
	hHEP	3.5	3.22	2.84	0.93	0.82	1.34	3.51	2.48	2.01	1.7	2.29	1.04	2.06	2.64	1.03	1.12	1.05	1.17	0.98	1.05	1.2							
	hESC	3.75	3.41	3.08	1.41	1.13	0	4.35	1.9	1.8	1.58	1.02	1.23	0	2.27	0.91	0.95	0.93	1.02	1.06	0	1.16							

the BEELINE evaluation [37], these algorithms have been shown to achieve state-of-the-art performance on benchmark datasets.

The performance of the seven algorithms was assessed by BEELINE, and their results are displayed in table 2. Table 3 shows the results of GRNRI only. Fig. 6 presents a complete comparison of GRNRI and the other seven algorithms.

GRNRI achieves the best prediction performance on 68.18% (15/22) of the benchmarks. Compared to the maximum results achieved by state-of-the-art algorithms, the maximum improvement that could be achieved by GRNRI is 4.24 and the average improvement is 1.04. The worst result achieved by GRNRI compared to other algorithms is 1.64 when evaluating mDC dataset using STRING or Non-specific ChIP-Seq as ground truth networks. All results achieved by GRNRI using Cell-type specific ChIP-Seq or Iog/gof as ground truth networks were better than all the results achieved by other algorithms.

Due to computation limitations, the maximum used

TABLE III. According to BEELINE assessment and the setting used in table I, the results achieved by GRNRI are shown in the table.

		Ground Truth Networks			
		STRING	Non-specific ChIP-Seq	Cell-type specific ChIP-Seq	log/gof
Datasets	mESC	4.39	2.66	1.56	2.37
	mDC	1.64	1.64	5.88	
	mHSC-E	6.9	9.09	1.73	
	mHSC-GM	12.5	16.67	1.47	
	mHSC-L	10.42	2.97	1.9	
	hHEP	2.58	4.44	1.51	
	hESC	3.22	7.81	4.07	

batch size was 20, and the number of hidden layers used in MLPs was 64. We expect our proposed model to perform better if these parameters are allowed to increase.

6. Conclusion and Future Work

We proposed GRNRI, a new method for inferring GRNs from scRNA-seq data in an unsupervised way. GRNRI is based on a variational auto-encoder model, which is a type of generative model that can learn the latent structure of complex data [38]. GRNRI uses the variational auto-encoder to encode the scRNA-seq data into a low-dimensional latent space, and then infers the GRN from the latent variables. To evaluate the performance of GRNRI, we followed the BEELINE framework [37], which is a comprehensive evaluation framework for GRN inference algorithms for single-cell gene expression data. The BEELINE framework collected four different types of ground-truth networks and seven scRNA-seq datasets for evaluation [37]. We compared our results to state-of-the-art methods in the BEELINE framework, and we found that GRNRI achieves comparable or superior performance using EPR as a metric for evaluation. Moreover, GRNRI can be extended to solve other potential tasks as well such as visualizing and simulating scRNA-seq data. Following the work in [5], we can use the variational auto-encoder model to generate realistic synthetic scRNA-seq data from the learned latent space. The model being modified in [12] was already able to solve the simulation task.

References

[1] A. Wagner, A. Regev, and N. Yosef, "Revealing the vectors of cellular identity with single-cell genomics," *Nature Biotechnology*, vol. 34, pp. 1145–1160, 11 2016.

[2] P. Kharchenko, L. Silberstein, and D. Scadden, "Bayesian approach to single-cell differential expression analysis," *Nature methods*, vol. 11, 05 2014.

[3] D. Marbach, J. Costello, R. Küffner, N. Vega, R. Prill, D. Camacho, K. Allison, A. Aderhold, R. Bonneau, Y. Chen, J. Collins, F. Cordero, M. Crane, F. Dondelinger, M. Drton, R. Esposito, R. Foygel, A. de la Fuente, J. Gertheiss, and R. Zimmer, "Wisdom of crowds for robust gene network inference," *Nature Methods*, vol. 9, pp. 796–804, 07 2012.

[4] H. Shu, J. Zhou, Q. Lian, H. Li, D. Zhao, J. Zeng, and J. Ma, "Modeling gene regulatory networks using neural network architectures," *Nature Computational Science*, vol. 1, pp. 491–501, 07 2021.

[5] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, "Inferring regulatory networks from expression data using tree-based methods," *PloS one*, vol. 5, 09 2010.

[6] T. E. Chan, M. P. Stumpf, and A. C. Babtie, "Gene regulatory network inference from single-cell data using multivariate information measures," *Cell Systems*, vol. 5, no. 3, pp. 251–267.e3, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405471217303861

[7] H. Matsumoto, H. Kiryu, C. Furusawa, M. Ko, S. Ko, N. Gouda, T. Hayashi, and I. Nikaido, "Scode: An efficient regulatory network inference algorithm from single-cell rna-seq during differentiation," *Bioinformatics (Oxford, England)*, vol. 33, 04 2017.

[8] N. Gao, M. Ud-Dean, O. Gandrillon, and R. Gunawan, "Sincerities: Inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles," *Bioinformatics (Oxford, England)*, vol. 34, 09 2017.

[9] T. Moerman, S. Aibar, C. Bravo González-Blas, J. Simm, Y. Moreau, J. Aerts, and S. Aerts, "Grnboost2 and arboreto: Efficient and scalable inference of gene regulatory networks," *Bioinformatics (Oxford, England)*, vol. 35, 11 2018.

[10] K. Kamimoto, B. Stringa, C. Hoffmann, K. Jindal, L. Solnica-

Krezel, and S. Morris, "Dissecting cell identity via network inference and in silico gene perturbation," *Nature*, vol. 614, pp. 1–10, 02 2023.

[11] S. Kim, "Ppcor: An r package for a fast calculation to semi-partial correlation coefficients," *Communications for Statistical Applications and Methods*, vol. 22, pp. 665–674, 11 2015.

[12] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," *Proceedings of ICLR*, 02 2018.

[13] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[14] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016.

[15] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 1263–1272.

[16] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

[17] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. GÃ³mez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in Neural Information Processing Systems (NIPS)*, vol. 13, 09 2015.

[18] H. Dai, B. Dai, and L. Song, "Discriminative embeddings of latent variable models for structured data," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 2702–2711.

[19] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 2014–2023.

[20] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 3844–3852.

[21] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: Moving beyond fingerprints," *Journal of Computer-Aided Molecular Design*, vol. 30, 08 2016.

[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[23] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 1025–1035.

[24] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 2252–2260.

[25] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 4509–4517.

[26] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4974–4983.

[27] A. Sonawane, D. DeMeo, J. Quackenbush, and K. Glass, "Constructing gene regulatory networks using epigenetic data," *npj Systems Biology and Applications*, vol. 7, 12 2021.

[28] L. Liang, J. Yu, J. Li, N. li, J. Liu, L. Xiu, J. Zeng, T. Wang, and L. Wu, "Integration of scrna-seq and bulk rna-seq to analyse the heterogeneity of ovarian cancer immune cells and establish a molecular risk model," *Frontiers in Oncology*, vol. 11, 09 2021.

[29] G. Carangelo, A. Magi, and R. Semeraro, "From multitude to singularity: An up-to-date overview of scrna-seq data generation and analysis," *Frontiers in Genetics*, vol. 13, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fgene.2022.994069

[30] N. Lytal, D. Ran, and L. An, "Normalization methods on single-cell rna-seq data: An empirical survey," *Frontiers in Genetics*, vol. 11, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fgene.2020.00041

[31] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *CoRR*, vol. abs/1611.00712, 2016. [Online]. Available: http://arxiv.org/abs/1611.00712

[32] T. Hayashi, H. Ozaki, Y. Sasagawa, M. Umeda, H. Danno, and I. Nikaido, "Single-cell full-length total rna sequencing uncovers dynamics of recursive splicing and enhancer rnas," *Nature Communications*, vol. 9, 02 2018.

[33] A. Shalek, R. Satija, J. Shuga, J. Trombetta, D. Gennert, D. Lu, P. Chen, R. Gertner, J. Gaublomme, N. Yosef, S. Schwartz, B. Fowler, S. Weaver, J. Wang, X. Wang, R. Ding, R. Raychowdhury, N. Friedman, N. Hacohen, and A. Regev, "Single cell rna seq reveals dynamic paracrine control of cellular variation," *Nature*, vol. 510, 06 2014.

[34] S. Nestorowa, F. K. Hamey, B. Pijuan Sala, E. Diamanti, M. Shepherd, E. Laurenti, N. K. Wilson, D. G. Kent, and B. GÃ¶ttgens, "A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation," *Blood*, vol. 128, no. 8, pp. e20–e31, 08 2016. [Online]. Available: https://doi.org/10.1182/blood-2016-05-716480

[35] J. Camp, K. Sekine, T. Gerber, H. LÃ¶hffler-Wirth, H. Binder, M. Gac, S. Kanton, J. Kageyama, G. Damm, D. Seehofer, L. Belicova, M. Bickle, R. Barsacchi, R. Okuda, E. Yoshizawa, M. Kimura,

H. Ayabe, H. Taniguchi, T. Takebe, and B. Treutlein, "Multilineage communication regulates human liver bud development from pluripotency," *Nature*, vol. 546, 06 2017.

[36] L.-F. Chu, N. Leng, J. Zhang, Z. Hou, D. Mamott, D. Vereide, J. Choi, C. Kendziorski, R. Stewart, and J. Thomson, "Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm," *Genome Biology*, vol. 17, 08 2016.

[37] A. Pratapa, A. Jalihal, J. Law, A. Bharadwaj, and T. Murali, "Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data," *Nature Methods*, vol. 17, pp. 1–8, 02 2020.

[38] M. Tsutsumi, N. Saito, D. Koyabu, and C. Furusawa, "A deep learning approach for morphological feature extraction based on variational auto-encoder: an application to mandible shape," *NPJ systems biology and applications*, vol. 9, no. 1, p. 30, July 2023. [Online]. Available: https://europepmc.org/articles/PMC10322894

Fatema Abdulhameed Fatema Abdulhameed received her B.Sc. degree with an Excellent grade from the Computer Engineering and Software System program of Ain Shams University in 2019. She is currently working towards obtaining her M.Sc. degree from the department of Computer and Systems Engineering in the same university were she also worked as a teaching assistant for two years. She taught courses in Digital Design, Pattern Recognition, Data Structures and Algorithms, and Android Development.

Hazem Abbas Hazem Abbas received his B.Sc. and M.Sc. degrees in Computer Engineering from Ain Shams University, Cairo, and Ph.D. from Queen's University, Kingston, Canada. He is the currently a Visiting Professor at the Queen's University School of Computing. He has held leading positions at the Dept. Computer and Systems Engineering at Ain Shams University and at the German University in Cairo. He worked for the Royal Military College, IBM Toronto Lab, and Mentor Graphics. Prof. Abbas is a Senior Member of the IEEE and chaired the IEEE Signal Processing Chapter in Cairo. His research interests are in the areas of Machine Learning and Computational Intelligence.

Cherif Salama Cherif Salama received his B.Sc. and M.Sc. degrees from the computer and systems engineering department of Ain Shams University (ASU) in 2001 and 2006 respectively. In 2010, he received his Ph.D. degree in computer science from Rice University, Houston, Texas. He worked as assistant professor in the Computer and Systems Engineering Department of ASU and as adjunct lecturer in the EELU, the MIU, and the GUC. He served as unit head of the Computer Engineering and Software Systems program at Ain Shams University (ASU). He is currently an associate professor and associate chair of the Computer Science and Engineering Department at The American University in Cairo (AUC). He was also the key person in the design and development of the Verilog Preprocessor funded by Intel through the SRC. His research interests and publications span a wide spectrum including computer architecture, CAD, hardware description languages, programming languages, parallel computing, and AI.