Int. J. Data Mining and Bioinformatics, Vol. X, No. Y, XXXX

Effective induction of gene regulatory networks using a novel recommendation method

Makbule Gulcin Ozsoy and Faruk Polat

Department of Computer Engineering,
Middle East Technical University,
06800 Ankara, Turkey
Email: makbulegulcin@gmail.com
Email: polat@ceng.metu.edu.tr

Reda Alhajj

Department of Computer Science,
University of Calgary,
2500 University Dr. NW Calgary,
T2N 1N4 Alberta, Canada
and
Department of Computer Engineering,
Istanbul Medipol University,
Istanbul, Turkey
Email: alhajj@ucalgary.ca
Email: ralhajj@medipol.edu.tr
*Corresponding author[AQ1]

AQ1: Please specify the corresponding author's name.

Abstract: In this paper, we introduce a method based on recommendation systems to predict the structure of Gene Regulatory Networks (GRNs) making use of data from multiple sources. Our method is based on collaborative filtering approach enhanced with multiple criteria to predict the relationships of genes, i.e., which genes regulate others. We conduct experiments on two data sets to demonstrate the applicability and sustainability of our proposal. The first data set is composed of microarray data and Transcription Factor (TF) binding data, and it is evaluated by precision, recall and the F1-measure. The second data set is the Dream4 in Silico Network Challenge data set, and it is evaluated by the measures that are used during the challenge, namely the Area Under Precision and Recall curve (AUC-PR), the Area Under the Receiver Operating Characteristic curve (AUC-ROC) and their averages. The experimental results show that applying algorithms from the recommendation systems domain on the problem of inference of GRN structures is effective. Also, we observed that combining information from multiple data sets gives better results.

Keywords: GRNs; gene regulatory networks, recommendation systems, collaborative filtering, multiple data sources, Pareto dominance.

Reference to this paper should be made as follows: Ozsoy, M.G., Polat, F. and Alhajj, R. (XXXX) 'Effective induction of gene regulatory networks using a novel recommendation method', *Int. J. Data Mining and Bioinformatics*, Vol. X, No. Y, pp.xxx–xxx.

Copyright © 200X Inderscience Enterprises Ltd.

AQ2: Please supply up to 100 words of brief career history for each author.

Biographical notes: Makbule Gulcin Ozsoy [AQ2]

Faruk Polat [AQ2]

Reda Alhajj [AQ2]

AQ3: If a previous version of your paper has originally been presented at a conference, please complete the statement to this effect or delete if not applicable.

This paper is a revised and expanded version of a paper entitled [title] presented at [name, location and date of conference]. [AQ3]

1 Introduction

Network modelling has numerous applications in various branches of science, engineering, and humanities. For example, in sociology, researchers study friendship networks, in information technology, researchers work on networks emerging from the world-wide-web, in engineering, researchers study traffic networks and in biology, scientists study interactions among molecules, drugs, diseases, proteins, and genes (Wang et al., 2014). The structure of a network may reveal valuable information mostly unknown or hard to experiment within a laboratory. For example, in biology, it is very costly to directly observe gene relationships by wetlab experiments (Wang et al., 2014). However, it is easier to measure gene expression levels, which can be used to computationally infer on connections among genes.

In biology, there exists a huge collection of data on DNA, RNA, proteins, and metabolites, which can be used to infer interactions among biological components (Ristevski, 2013). Gene Regulatory Networks (GRNs) are composed of these components and their interactions. Some properties of GRNs are sparseness, scale-free topology, modularity and structurally of the inferred networks (Hecker et al., 2009). In GRNs, the numbers of connections among genes are limited, such that GRNs are sparse. GRNs follow the power distribution for a connectivity (Nicolau and Schoenauer, 2009; Zhang et al., 2014), such that some genes regulate many other genes, while some others regulate only a few or no other genes. This property is related to the scale-free topology feature of GRNs. They are structurally decomposable into network motifs (Hecker et al., 2009), and they can be clustered into groups such that the genes in a group are highly co-expressed or have similar functions (Ristevski, 2013). GRNs and recommendation systems have similar features. For instance, both are sparse, follow the power distribution in their topology (Ye et al., 2011; Gao et al., 2012; Bao et al., 2012) and both are decomposable into clusters or network motifs. Observing these similarities, we conducted an initial study described in Ozsoy et al. (2015) where we used a method from the recommendation systems literature to construct GRNs computationally.

In the literature, four main approaches are used to infer GRNs, namely, Boolean networks, Bayesian networks, relevance networks and differential and difference equations (Ristevski, 2013). Recently, the integration of prior knowledge to the GRN inference process gained attention in the literature (Ristevski, 2013; Michailidis and

d'Alché-Buc, 2013; Tan et al., 2008; Hecker et al., 2009)). In Boolean networks, the gene expression levels and a threshold parameter are used. Using such information, gene interactions are represented as a Boolean function. In the reverse engineering process, for each gene a related Boolean function is found (Ristevski, 2013). A REVerse Engineering ALgorithm (REVEAL) (Liang et al., 1998; Kwon and Cho, 2007; Shmulevich et al., 2002) are some example approaches that are based on Boolean networks. The Bayesian networks approaches use the conditional dependence of genes, where the conditional probabilities are based on the parent nodes only. The probability of the graph is calculated by a joint probability distribution, which is dependent on the probability of existence of edges between genes. These approaches are commonly used for GRN structure prediction (Ristevski, 2013). Some example approaches that are based on Bayesian networks can be found in Friedman et al. (2000), Zhang et al. (2007), Grzegorczyk and Husmeier (2008), Tan et al. (2008) and Shermin and Orgun (2009). In the relevance networks, the structures of GRNs are decided by correlation and/or mutual information among genes. If the correlation score is above a predefined threshold, genes are predicted to be connected. Usually, the output graph of these methods are undirected. ARACNE (Margolin et al., 2006) and Schafer and Strimmer (2005) are example approaches that are based on relevance networks. The use of differential and difference equations form another set of approaches, which use the input gene expression data, the time, model parameters and external effects. They find the changes in the gene expression data and the graph structure. Some example approaches that are based on differential and difference equations may be found in Wessels et al. (2001), Gebert et al. (2007) and Li and Zhang (2007). Recently, different methods that integrate prior knowledge and multiple types of data to the GRN inference process gained more attention in the literature. Zhang et al. (2007), Li and Zhang (2007), Tan et al. (2008), Ideker et al. (2011), Zhang et al. (2014) and Ozsoy et al. (2015) are example approaches that combine multiple data sources to infer the GRNs more accurately. To the best of our knowledge, our work that builds on our proof of concept effort described in Ozsoy et al. (2015) is the first GRN inference method that adopts a method from recommendation systems.

We conducted an initial study Ozsoy et al. (2015) to construct GRNs computationally using a method from the recommendation systems literature. The recommendation method has been proposed in Ozsoy et al. (2014) to give recommendations to target users on next check-in location and it is based on Pareto dominance and collaborative filtering. In Ozsoy et al. (2015), we used this recommendation method with multiple different settings to predict interaction between genes. For the evaluation we used a combination of two different data sets, namely microarray data (Spellman et al., 1998) and Transcription Factor (TF) binding data (Lee et al., 2002). In this paper, we extend our previous work by introducing new settings and by experimenting on different data sets. As the first data set, we used the same data set that we used in our previous work Ozsoy et al. (2015); which is a combination of microarray data (Spellman et al., 1998) and TF binding data (Lee et al., 2002). As the second data set, we used Dream4 in Silico Network Challenge (Dream4, 2015) data to see the performance of our method on different sizes of data sets.

The contributions of this work may be listed as follows:

- It adds a new dimension to the study of GRNs by adopting a methodology from recommendation systems to predict the structure of GRNs. The employed method

builds on our proof of concept initiative described in Ozsoy et al. (2015). It integrates Pareto dominance and Collaborative filtering methods to combine multiple features and uses similarity among genes.

- It extends our previous work by integrating new parameters into the proposed method. In our previous work, the number of genes which were predicted to be connected to a target gene was predefined. However, this restriction has been loosened in this paper by introducing a threshold parameter.
- It reports results from experiments which have been conducted on two different data sets. The first data set combines two sources and has already been used in our previous work. The second data set is the Dream4 in Silico Network Challenge (Dream4, 2015). Using various data sets we have been able to observe performance of the proposed method by utilising various parameters combined with different sizes of data data sets.

The rest of this paper is organised as follows. Section 2 describes the employed GRN inference method. Section 3 presents the evaluation process and the results. Section 4 is the conclusion.

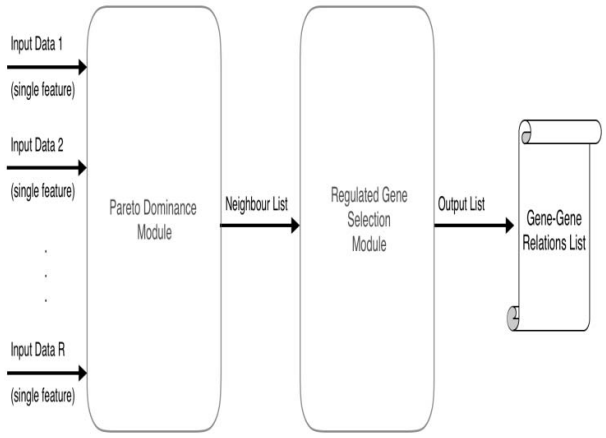
2 Methods

The aim of this work is to predict the structure of GRNs by inferring which genes regulate others using a known method from recommendation systems domain. The recommendation method we use for this purpose is based on Pareto dominance and collaborative filtering approaches. In traditional recommendation methods, the aim is to predict sers' future item preferences and to recommend those items to target users. Mapping the same concept to the molecular interactions domain, target genes are used instead of target users, and genes that the target genes bind to (regulate) are predicted instead of items. Here, it is worth noting that even though GRNs contain many different components, such as biomarkers, genes and proteins, in this work we refer to all of them as genes. Before giving details of how Pareto dominance and collaborative filtering have been combined to predict the structure of GRNs, we want to give a short explanation of these methods. In recommendation systems, the collaborative filtering method collects previous preferences of users and based on that information it predicts future preferences of target users. The main assumption in collaborative filtering is that if two users share similar taste on an item, they are more likely to have more similar opinion on an unknown item than a random user. This assumption can be easily mapped to gene networks, such that if two genes already have similar features, they are more likely to regulate same genes. Pareto dominance aims to optimise the solution when there are multiple criteria (features). For instance, consider a recommendation systems problem: There is one target user and two candidate neighbours. All users are represented using three different features. The aim is to find out the most similar (representative) neighbour to the target user. To solve this problem Pareto dominance technique can be used which decides on the non-dominated (the most representative) candidate and that candidate can be assigned as the neighbour of the target user. Similar approach can be used when genes are represented with multiple features and the aim is to figure out genes most similar to a target gene. The Pareto dominance and collaborative filtering methods can be combined

together by using the output of Pareto dominance technique as input to the collaborative filtering technique. Using the Pareto dominance method lets us to use multiple features to decide the most representative neighbour, which is not possible in traditional collaborative filtering method. The features that are used can be collected from multiple sources as well as a single source.

We execute the proposed recommendation-based GRN inference method in three steps (see Figure 1): similarity calculation, neighbour selection, and regulated genes selection. For the neighbour selection and regulated genes selection steps, it is possible to use several different settings. We have described these settings in our previous work (Ozsoy et al., 2015). In this paper, we extend the proposed method by introducing a new setting for the regulated genes (items) selection step. In our previous work, the number of genes which were predicted to be connected to the target gene was predefined. This restriction has been loosened in this paper by introducing a threshold parameter. The explanation of the steps of the proposed recommendation-based GRN inference method are as follows:

Figure 1 Design of the system



Similarity calculation: Similarity among genes is calculated using the available features which can be extracted from multiple data sets or a single data set. For similarity calculations, there are various similarity or correlation calculation methods described in the literature. In Jaskowiak et al. (2014), several different distance measures are compared to each other in a setting that tackles a bioinformatics problem. The authors concluded that for different methods specific measures perform better than others. They found that Rank-Magnitude, Jackknife, Pearson, and Cosine similarity measures are usually the best choices for the bioinformatics domain. Based on these observations, we preferred to use Cosine similarity (see equation 1).

$$sim(A,B) = \frac{\sum_{i=1}^{n} A_j \times B_i}{\sqrt[2]{\sum_{i=1}^{n} A_i^2} \times \sqrt[2]{\sum_{i=1}^{n} B_i^2}} \tag{1}$$

In equation (1), A and B refer to genes, which may have multiple features and each feature is indicated by the subscript i. For example, assume that A and B represent genes ACE2 and ASH1, respectively. In Spellman et al. (1998) data set, these genes are represented by 77 different measures. These measures are further divided into three phases in Bernard et al. (2005); each of which contains different number and kind of measures. So, ACE2 and ASH1 are represented by three phases, which may be features (i in the equation).

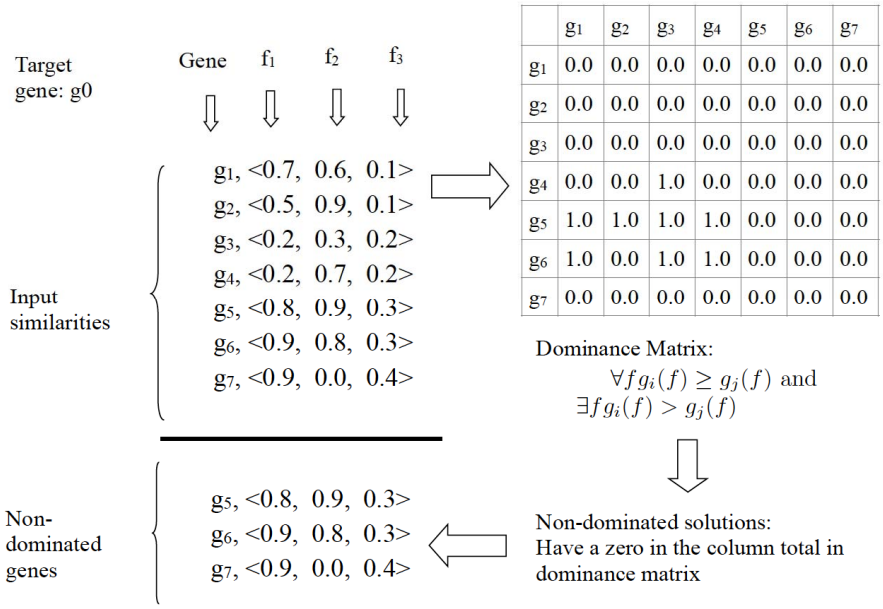
Neighbour selection: The similarities calculated in the previous step are used to decide neighbour genes which are used in the next step to predict target genes' connections. In this step, the neighbour genes are decided by using the Pareto dominance method. This method is able to find out the most representative neighbours by considering multiple objectives. In our setting the objective is to maximise similarity between target and candidate genes for each feature. In Pareto dominance method, the most representative neighbours are found by calculating the dominance relations among candidate genes, such that if a gene is non-dominated then it is assigned as a neighbour. The Pareto dominance of genes is calculated by equation (2), where g_i and g_j represents genes and f indicates the features. According to the equation, if gene g_i has at least one higher similarity value and no lower similarity values than gene g_j, then gene g_i dominates gene g_j.

$$dom(g_i, g_j) = \begin{cases} 1.0 & \forall f\ g_i(f) \geq g_j(f) \text{ and} \\ & \exists f\ g_i(f) > g_j(f) \\ 0.0 & \text{otherwise} \end{cases} \quad (2)$$

For example, a multi-dimensional data is given in Figure 2. In this example, for different features, namely f_1, f_2 and f_3, similarities of seven genes to the target gene are given. The f_i values are the calculated similarities in the previous step. In order to find the non-dominated genes, the first step is to create the dominance matrix. In a dominance matrix, the cell values indicate if the gene in the row dominates the gene in the column. The cell values are filled with the values calculated by equation (2). The output dominance matrix is given in Figure 2, based on the example similarities. Having the dominance matrix, non-dominated genes are decided by looking at the column sums of the dominance matrix. Genes whose column sum equals to 0.0 are the non-dominated genes.

The neighbours found in this step are used in the next step as an input to an item selection step which is similar to what is done in collaborative filtering method. In traditional collaborative filtering method, the number of neighbours is given as an input to the method. Similar approach can be followed in this step, such that the number of neighbours to be found by Pareto dominance can be predefined. However, original Pareto dominance method does not allow this. As suggested in Ozsoy et al. (2014), an iterative process of neighbour collection can be applied to collect as many neighbours as requested. Following the same idea, we first apply the Pareto dominance method explained above. If the number of collected neighbour genes is less than the input neighbour count, we remove the selected genes from the data representation and re-apply the method of finding non-dominated genes. We continue this process until the given number of neighbour count is reached.

Figure 2 Example input and non-dominated solutions



	g1	g2	g3	g4	g5	g6	g7
g1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
g2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
g3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
g4	0.0	0.0	1.0	0.0	0.0	0.0	0.0
g5	1.0	1.0	1.0	1.0	0.0	0.0	0.0
g6	1.0	0.0	1.0	1.0	0.0	0.0	0.0
g7	0.0	0.0	0.0	0.0	0.0	0.0	0.0

In addition to this setting of collecting as many neighbours as the input neighbour count, it is still possible to use other settings. We named these neighbour selection preference settings as "Multi-Objective Optimisation Type" (MOT) setting. The explanations and abbreviations of the neighbour selection preference settings are as follows:

- *Only_Dominates (OD)*: Find non-dominated neighbours in a single iteration. The number of non-dominated genes is not set and it depends directly on the similarity values.
- *N_Dominates (ND)*: Find exactly *N* neighbours by running multiple iterations and pruning when necessary. Pruning is performed, if more than input neighbour count neighbours are extracted as a result of iterative application of Pareto dominance method.
- *At_Least_N_Dominates (AND)*: Find at least *N* neighbours by running multiple iterations. Unlike N_Dominates setting, no pruning is applied in this setting.

Regulated genes (Item) selection: Knowing the neighbours and their connections, the regulated genes are predicted by using collaborative filtering. The directions of edges (i.e., if a gene is regulated by a target gene or not) are decided by considering neighbours' connections and the calculated connection score. First, genes regulated by the neighbours are identified as candidate genes; they are assigned to be more promising to be regulated by the target gene. The score for each candidate gene is calculated by equation (3), where *score(c,t)* represents the connection score of the candidate gene, *c* and *t* represent the candidate and target genes, respectively, *n* represents the neighbour gene and *Nghb* represents the set of neighbour genes. The similarity among the target and

neighbour genes $(sim(t,n))$ and the binding probabilities of the neighbour and candidate genes $(b(n,c))$ are used in the calculation. If there is no previous knowledge on binding probabilities, it is possible to use the similarity between the neighbour and the candidate as well. The higher the resulting connection score is, the more promising is the gene to be regulated by the target gene.

$$score(c,t) = \sum_{n \in \text{Nghb}} sim(t,n) \times b(n,c) \tag{3}$$

In this step, we apply two different settings which are related to how the score is calculated and how predicted genes are decided based on their scores:

- *Regulated genes (item) selection method type (IST)*: A general formula is presented in equation (3). However, it is possible to use different values for the similarities and binding probabilities.
  - *Sum (SUM)*: Without considering the similarities between the target and the neighbour genes, the binding score for each candidate gene is summed, such that $sim(t,n) = 1$ for all neighbours.
  - *Average (AVG)*: After summing up the values as done in the SUM method-, the result is divided into the number of neighbours that suggest the candidate gene.
  - *Maximum (MAX)*: For each candidate gene, the maximum binding probability is used, without considering the similarity between the target gene and the neighbour genes.
  - *Weighted Average (WAVG)*: The AVG method is performed by additionally using the similarities among the target and the neighbour genes. So instead of dividing the summation into the number of neighbour genes, it is divided into the summation of the similarities between the target and the neighbour genes. In the application, we used the binding probabilities between the target and the neighbour genes, instead of calculating the similarities, i.e., $sim(t,n) = b(t,n)$.

An example is given in Figure 3 to show how the different regulated genes (items) selection method affects the predictions. In the figure the aim is to predict a single gene that the target gene $(g0)$ regulates. The neighbours are given as $g1$, $g2$ and $g3$. The binding probabilities of the neighbours and the candidate genes ($g4$ and $g5$) are also given. Based on the different regulated genes (items) selection method, different genes are predicted to be regulated by the target gene. For example, when the summation is used, gene $g4$ is predicted since its connection score is 1.6 and the score of gene $g5$ is 0.9. However, when averages are used the average score of $g4$ is 0.8 and the average score of $g5$ is 0.9, so $g5$ is predicted to be regulated. For the weighted average method, the similarities between the target gene, $g0$, and the neighbours, $g1$, $g2$ and $g3$, affect the prediction results and depending on these values either $g4$ or $g5$ can be predicted to be regulated by the target gene.

- *Outlist Type (OT)*: The output of the method may contain a fixed sized list or may depend on threshold on the similarities, binding probabilities or connection scores of the genes.
    - *Fixed_Length (F)*: The number of genes that are predicted to be regulated by the target gene is fixed and it is given as input to the method (*k*). In this setting the genes with top-*k* connection score are presented as the output recommendation.
    - *Threshold_Based (T)*: The number of genes that are predicted to be regulated by the target gene is based on the input threshold (*T*). In this setting the genes with connection score higher than *T* are presented in the output recommendation. Also, *T* is used to prune candidate genes, such that the candidate genes whose binding probabilities to the neighbours are less than *T* are removed from the candidate list.

Figure 3 Example for regulated genes (items) selection method

Target: g0, Neighbors: g1, g2, g3 Candidates: g4, g5 Binding prob: g1→ g4: 0.8 g2→ g4: 0.8 g3→ g5: 0.9	Predicted gene to be regulated is: • Sum: g4 • Avg: g5 • Max: g5 • Weighted avg: g4 or g5 (Depending on the similarities between the target and neighbor genes)

3 Results and discussion

We run two different experiments to evaluate the applicability and effectiveness of the proposed approach. In both of the evaluation settings, we have information about several different features of each gene and we are able to calculate the similarities among genes. Also, gene-gene relations are either given in the data sets or collected using available tools in the literature (see Figure 4). In order to measure the performance of the proposed method, we exclude the target gene's known relations and try to predict them using the proposed method (see Figure 5).

In the first evaluation setting, we combined two different data sets, namely microarray data from Spellman et al. (1998) and Transcription Factor (TF) binding data from Lee et al. (2002). In the second evaluation setting, we used the Dream4 in Silico Network Challenge data set (Dream4, 2015) to observe the performance of our method on different sizes of data sets[1]. We present the details of these experiments in Sub-sections 3.1 and 3.2.

Figure 4 Relations among genes

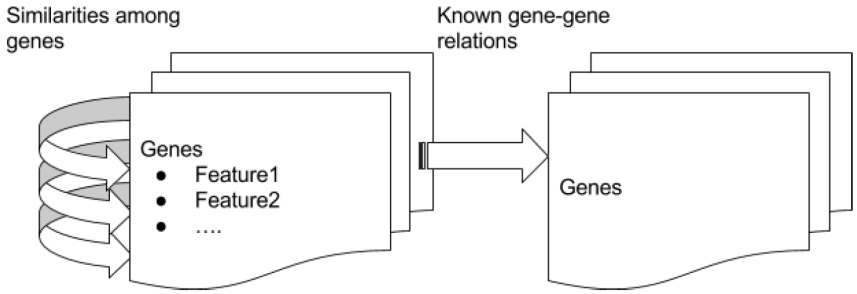
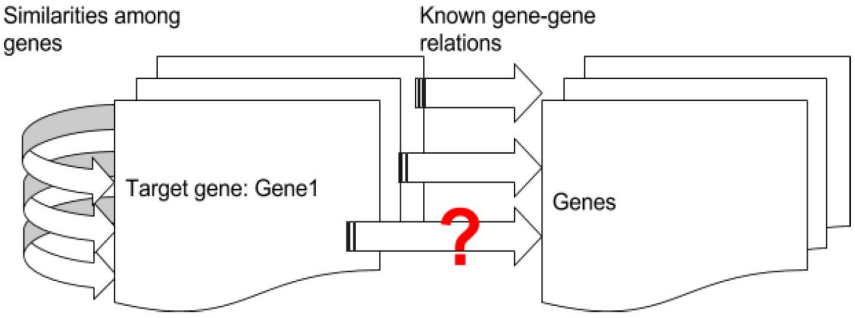


Figure 5 Experimental setting



3.1 Microarray and transcription factor (TF) binding data sets

We have already used in Ozsoy et al. (2015) the combination of the microarray data from Spellman et al. (1998) and the TF binding data from Lee et al. (2002); the same data sets were used in an earlier research project by our group (Tan et al., 2008). Similar to those works, we evaluated the performance of the proposed method and settings using precision@k, recall@k and $f1$-measure. These metrics are computed as given in equations (4), (5) and (6), where k indicates the output list length, tp indicates the true positives, i.e., predicted and actually connected/regulated genes, fp indicates the false positives, i.e., predicted but actually not connected genes, and fn indicates the false negatives, i.e., not predicted but actually connected genes.

$$Precision_k = \frac{tp_k}{tp_k + fp_k} \tag{4}$$

$$Recall_k = \frac{tp_k}{tp_k + fn_k} \tag{5}$$

$$F1 - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

The microarray data represents time series gene expression data and contains 6178 genes and 77 time steps. Instead of using each time step as a feature, we divided them into three phases, as done by Bernards et al. (2005), and used these phases as our features. The reason for using phases instead of time steps is to reduce data set sparsity. The TF binding data contains binding location data of 6270 genes and 106 TFs. In the earlier efforts from our group described in Tan et al. (2008) and Ozsoy et al. (2015), 25 of these genes are chosen, based on the studies reported in Bernard et al. (2005). Following these works, we used the same 25 genes for the evaluation. Also we executed the same pre-processing steps performed in Tan et al. (2008):

- Filling the missing values in the microarray data set (Spellman et al., 1998): The missing values existing in the data set for the selected 25 genes are filled by applying the *k*-nearest neighbours algorithm with $k = 10$, same as it was done in Tan et al. (2008).
- Converting *p*-values in the binding data (Lee et al., 2002) into probability values: The binding data (Lee et al., 2002) gives information on the *p*-values, which indicate the confidence of TF bindings to the genes (Tan et al., 2008). Smaller *p*-values indicate higher confidence. As suggested in Tan et al. (2008), we converted the *p*-values into probabilities of existence of connections. For this purpose, equation (7) is used; it is actually described in Bernard et al. (2005).

$$P\left(E_i \in G \mid P_i = p\right) = \frac{1}{\lambda_H - \lambda_L} \int_{\lambda_L}^{\lambda_H} \frac{\lambda e^{-\lambda p} \vartheta_{ij}}{\lambda e^{-\lambda p} \vartheta_{ij} + \left(1 - e^{-\lambda}\right)\left(1 - \vartheta_{ij}\right)} d\lambda \tag{7}$$

In equation (7), E_i represents one of the edges/connections in graph G, P_{ij} is the *p*-value of the edge E_{ij} which connects genes i and j. λ_H and λ_L are the highest and lowest bounds of λ, which is the parameter of the exponential distribution. The ϑ_{ij} is the short form for $P\left(E_{ij} \in G\right)$, where $\vartheta_{ij} = P\left(E_{ij} \in G\right)$. We assigned the values of λ_H, λ_L and ϑ_{ij} to $10,000$, 0.1 and 0.5, respectively, as suggested in Bernard et al. (2005) and Tan et al. (2008).

- Filling missing probabilities in the binding data (Lee et al., 2002): 10 of the selected 25 genes exist as TFs in the binding data. In the previous step, the probabilities are calculated only for these 10 TFs/genes. For the remaining 15 genes, we set their binding probabilities to 0.50, as done in Tan et al. (2008).

In order to collect the golden data, a commercial tool was used in our previous work described in Tan et al. (2008), and a public tool named (GeneMANIA, 2015) was used in Spellman et al. (1998). For the study described in this paper, we used GeneMANIA, which gives information on various interaction types, namely Genetic interactions, Co-localisation, Co-expression, Physical interactions, Shared protein domains, among others. The golden data for all these interactions of the chosen 25 genes were collected on March 10, 2015. Even though all the available interaction types are evaluated separately, we presented their average results as the overall evaluation results.

In order to combine the microarray data and the binding data, we used two different ways: In the first one, only the three phases extracted from the microarray data are used in the similarity calculations and neighbour selection steps. The binding data is used only on the regulated genes (items) selection step. In the second version, the binding data is also included in the first two steps of the method. In this version four different features; where three of them are from microarray data and one of them is from binding data; are used to decide on the neighbours. We named the ways of combining multiple data as 3F_Experiment and 4F_Experiment, based on the number of features used in the similarity calculations and neighbour selection steps, respectively.

In the experiments, depending on the settings explained in the previous section, we need three variables to be assigned, which are neighbours counts (N), the output list size (k) and threshold (T). We experimentally decided on the best performing parameters, by assigning different values to these parameters: For N and k different values in the range of (Bao et al., 2012; Ristevski, 2013) are assigned. We decided on 25 as the upper limit because it is the total number of the genes to be evaluated. For T, we used the range of [0.51, 1.00] with 0.03 increments. We present the best results for each regulated gene (item) selection method type (IST), using three or four features (3F_Experiment or 4F_Experiment). Depending on the evaluation metric, precision, recall and $f1$-measure, different parameters produced the best results. Even though we performed the experiments for all different N, k and T values with different selection methods and different number of features, in the following parts we only present the best results for different metrics.

In Table 1, we present the best results for each evaluation metric, namely precision, recall and $f1$-measure, while using three features. According to the table, the best performing methods use (at least) N many neighbours (AND or ND). There is no single winner in terms of item selection method. The results for 3F_Experiment show that when we favour $f1$-measure or recall, the settings with the best performance lists 21–23 genes out of 25 genes in the output list as the genes that are predicted to be regulated. This result contradicts with the GRNs sparsity feature. Considering this fact, we think using precision as the main objective is more reliable.

Table 1 The best results for the 3F_Experiment

N	*k*	*T*	*MOT*	*OLT*	*IST*	*Prec.*	*Recall*	*F1*
	21	–	AND	F	MAX	0.199	0.943	0.321
	23	–	ND	F	SUM	0.183	0.953	0.300
	23	–	ND	F	AVG	0. 183	0.953	0.300
	23	–	ND	F	WAVG	0. 183	0.953	0.300
	1	–	AND	F	MAX	0.402	0.092	0.145

We presented the results for precision with while using different item selection methods (see Table 2). According to the table, limiting the number of neighbours, such that using ND or AND as the neighbour selection setting and using MAX as the item selection setting performs the best. For this setting the best values for N and k are found as 3 and 1, respectively. When $k = 1$, the upper-bound for precision is 0.972, for recall it is 0.419 and for $f1$-measure it is 0.586. These results show that predicting few genes reaches nearly half of the upper-bound precision performance. Recall is low as expected because only few predictions are listed in the output list. When we compare the results for

3F_Experiment and 4F_Experiment, we observe a slight performance increase when we add the binding data as a new feature and use it in the similarity calculation and neighbour selection steps.

Table 2 The best results for the precision while using different IST settings with the 3F_Experiment and 4F_Experiment

Type	*N*	*k*	*T*	*MOT*	*OLT*	*IST*	*Prec.*	*Recall*	*F1*
F	12	2	–	ND	F	SUM	0.301	0.157	0.198
F	12	2	–	ND	F	AVG	0.301	0.157	0.198
F	3	1	–	AND	F	MAX	0.402	0.092	0.145
F	12	2	–	ND	F	WAVG	0.301	0.157	0.198
F	6	2	–	AND	F	SUM	0.301	0.157	0.198
F	6	2	–	AND	F	AVG	0.301	0.157	0.198
F	3	1	–	ND	F	MAX	0.404	0.097	0.151
F	5	2	–	AND	F	WAVG	0.310	0.161	0.203

In the experiments presented above, threshold based item selection approaches never provided the best results. When we looked at the results of this approach on both 3F_Experiment and 4F_Experiment setting, the following observations were realised: On 3F_Experiment, for the Item Selection Type (IST), assigning the threshold at least 0.66 and selecting exactly *N* neighbours (ND), where *N* is assigned to 2 gives the best results. In this setting, the best results are found to be around 0.250 for precision, around 0.174 for recall and around 0.198 for $f1$-measure. For 4F_Experiment, assigning the threshold at least 0.72 and selecting at least *N* neighbours (AND) provides the best results when we used threshold based item selection setting. The best *N* value is found to be 13. The best results are obtained as around 0.210 for precision, around 0.388 for recall and around 0.265 for $f1$-measure. The results for 4F_Experiment show that using the threshold is good at predicting the right genes as regulated, however-looking at the precision result-it may list too many of them on the output list.

We compare our results to those reported in Tan et al. (2008). However, we faced two problems. First, in the work described in Tan et al. (2008)a different golden data set is used, i.e., a commercial tool rather than GeneMANIA was used to create the golden set. This made results of the study described in this paper and the results of the work described in Tan et al. (2008) incompatible. However, the work described in Tan et al. (2008) presented the image of the output graph. We used that image to extract the structure of the output graph and we represented it in a way that can be used for the computations. This way, we were able to use the golden data extracted from GeneMANIA for the results in this paper and the results of the work described in Tan et al. (2008). The second problem is related to the directions of the edges on the graph. On the image of the resulting graph in Tan et al. (2008), the directions of the edges were not presented, i.e., it only show that there is a relation between the connected genes. However, our results are directed, i.e., we explicitly show which gene regulates other genes. To make both graphs compatible, we converted our resulting graphs into undirected graphs, by adding the reverse directions of the edges to the graph.

In Table 3, we present the results for 3F_Experiment, 4F_Experiment and Tan et al. (2008), in the order of the sections shown in the graph. As previously stated, we favour the precision results, and presented the results of the settings that perform best for the

precision. According to the table, among all of the methods the best precision result is obtained when exactly N neighbours (ND) are selected and MAX is used as the item selection method. In the table, the best recall performance is obtained by the ND neighbour selection approach together with the SUM or AVG item selection methods. For $f1$-measure, the best method uses ND with the WAVG approach. For all of the measures, the best results belong to the 4F_Experiment setting. From these results, we can conclude that adding the binding data to the similarity calculation and the neighbour selection steps increases the performance.

Table 3 The results for the undirected graph

Type	*N*	*k*	*T*	*MOT*	*OLT*	*IST*	*Prec.*	*Recall*	*F1*
F	1	1	–	ND	F	SUM	0.275	0.124	0.163
F	1	1	–	ND	F	AVG	0.275	0.124	0.163
F	5	1	–	AND	F	MAX	0.333	0.098	0.146
F	1	1	–	ND	F	WAVG	0.275	0.124	0.163
F	6	4	–	AND	F	SUM	0.248	0.470	0.299
F	6	4	–	AND	F	AVG	0.248	0.470	0.299
F	10	1	–	ND	F	MAX	0.342	0.085	0.132
F	6	4	–	ND	F	WAVG	0.250	0.464	0.300
(Tan et al., 2008)	–	–	–	–	–	–	0.213	0.193	0.203

Looking at all the performance results, i.e., precision, recall and $f1$-measure, the best method for item selection is the weighted average (WAVG) method for the directed graphs. For undirected graphs, there is no single winner for the item selection method. For the neighbour selection step, choosing exactly N many neighbours (ND) mostly performed better than the other approaches. For all of the experiments, using a fixed length output list (F) performed better than using a threshold (T). According to these observations, we use the ND and WAVG approaches to decide on the best N, the neighbour count, and k output list length values.

In Figures 6 and 7, we present the plot of precision values for different N and k values for the experiments using three and four features, namely 3F_Experiment and 4F_Experiment, respectively. The figures reveal that increasing the output length, k, decreases the precision@k. This is the expected behaviour, known from the information retrieval literature. The best setting that provides the best precision results is as follows: For the 3F_Experiment, $N = 12$ and $k = 2$ provides the best performance. For the 4F_Experiment, $N = 6$ and $k = 2$ provides the best performance. The results reveal that adding the binding data into the similarity calculation and neighbour selection steps helps the system to perform equally well while considering less number of neighbours. Reduced number of neighbours reduces the calculations for the prediction. Note that for both experiments, 3F_Experiment, and 4F_Experiment, k is found to be 2, which is a small value. Smaller k indicates that genes in general regulate small number of genes in the graph, and this matches the sparsity feature of GRNs.

Figure 6 Precision results for ND and WAVG with different *N* and *k* (3F_Experiment)

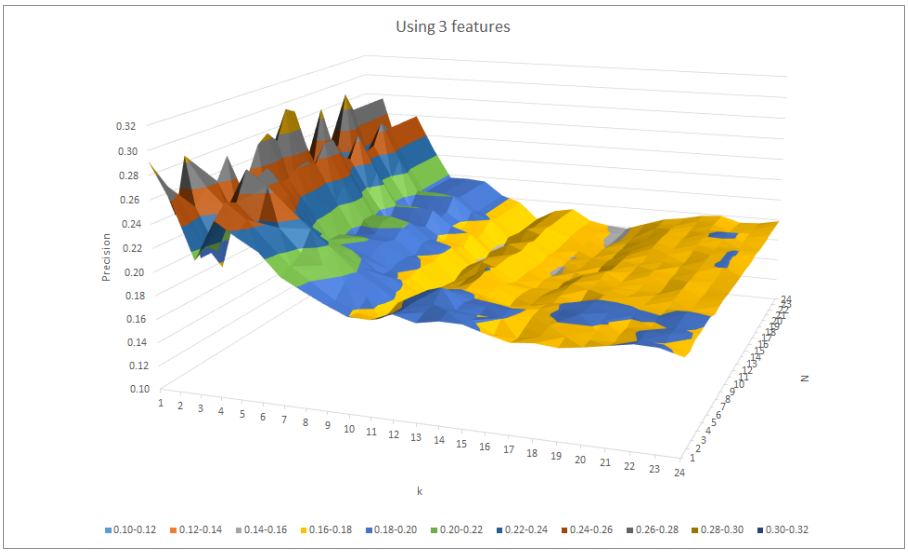
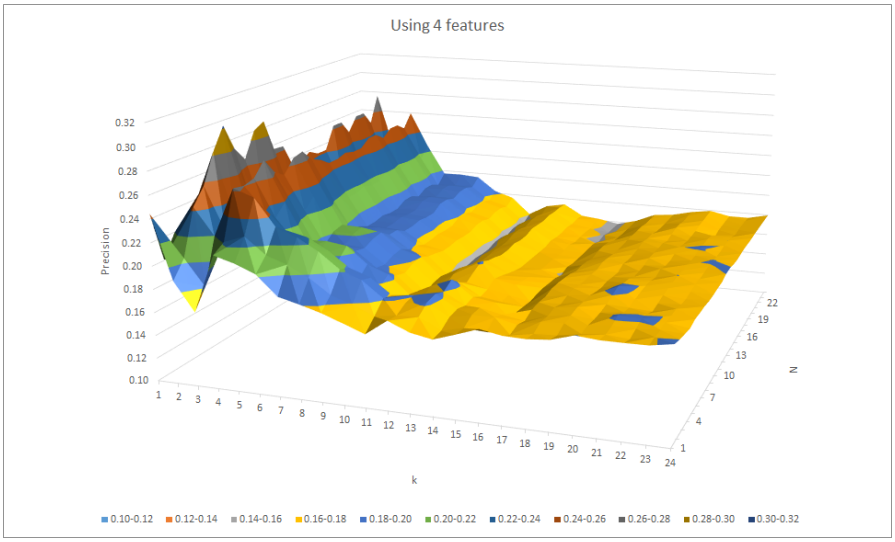


Figure 7 Precision results for ND and WAVG with different *N* and *k* (4F_Experiment)



To be fair to all of the multi-objective optimisation types, we also get the results using the WAVG method for choosing items and fixed length output list, where *N* is set to 6 and 12 and *k* is set to 2, based on the results observed from the previous figure. The precision results are given in Tables 4 and 5. The results for the 3F_Experiment are shown in the upper parts of the tables, while the results for the 4F_Experiment are shown in the lower parts. According to the tables, when *N* is set to 6, the best performance is

obtained by ND with four features (4F_Experiment) and when N is set to 12 the performance is obtained by ND with three features (3F_Experiment). These results are consistent with the previous results and show that fixed the number of neighbours and using multiple features from multiple sources are useful approaches to predict the genes to be regulated.

Table 4 The results for N=6 and k=2 with WAVG item selection method

Type	*N*	*k*	*T*	*MOT*	*OLT*	*IST*	*Prec.*	*Recall*	*F1*
F	–	2	–	OD	F	WAVG	0.263	0.126	0.164
F	6	2	–	AND	F	WAVG	0.284	0.148	0.186
F	6	2	–	ND	F	WAVG	0.279	0.146	0.183
F	–	2	–	OD	F	WAVG	0.255	0.128	0.164
F	6	2	–	AND	F	WAVG	0.288	0.149	0.188
F	6	2	–	ND	F	WAVG	0.302	0.155	0.196

Table 5 The results for N=12 and k=2 with WAVG item selection method

Type	*N*	*k*	*T*	*MOT*	*OLT*	*IST*	*Prec.*	*Recall*	*F1*
F	–	2	–	OD	F	WAVG	0.263	0.126	0.164
F	12	2	–	AND	F	WAVG	0.267	0.142	0.177
F	12	2	–	ND	F	WAVG	0.301	0.157	0.198
F	–	2	–	OD	F	WAVG	0.255	0.128	0.164
F	12	2	–	AND	F	WAVG	0.242	0.133	0.165
F	12	2	–	ND	F	WAVG	0.263	0.141	0.176

Lastly, we present the results for each interaction type provided by GeneMANIA, which are genetic interactions, co-localisation, co-expression, physical interactions, shared protein domains, among others. Based on the results of the previous experiments, we choose to present the results of 4F_Experiment using ND, WAVG, N=6 and k=2. Based on the results presented in Table 6, our method performs well for Genetic interactions, Physical interactions, Shared protein domains and Others, but relatively less well for Co-localisation and Co-expression.

Table 6 The results for ND, WAVG, N=6 and k=2

Int. Type	*Prec.*	*Recall*	*F1*
Genetic interactions	0.469	0.192	0.273
Co-localization	0.125	0.060	0.081
Co-expression	0.318	0.084	0.133
Physical interactions	0.227	0.179	0.200
Shared protein domains	0.300	0.273	0.286
Other	0.375	0.140	0.204

3.2 Dream4 in Silico network challenge data set

Dream4 in Silico Network Challenge (Dream4, 2015) was prepared by researchers from the Laboratory of Intelligent Systems of the Swiss Federal Institute of Technology in Lausanne and IBM T.J. Watson Research Centre in New York (Marbach et al., 2009; Marbach et al., 2010; Prill et al., 2010). Its goal was defined as reverse engineering the GRNs from simulated steady-state and time-series data, i.e., inferring the directed gene network. As part of the main challenge three sub-challenges are defined: namely InSilico _ Size10, InSilico_Size100 and InSilico_Size100_Multifactorial. In this report, we attacked the InSilico_Size10 and the InSilico_Size100 sub-challenges, with the aim of observing the performance on data with different sizes. For all of the sub-challenges, information for five different networks are provided and rankings of the teams are decided based on the predictions made for all the five networks. Their rationale to provide multiple networks instead of one is that they wanted to measure the consistency of the methods on independent networks with different topologies. They provide the golden data and the evaluation scripts for the researchers after the end of the challenge.

Five networks containing 10 nodes are provided in the InSilico_Size10 sub-challenge. The data provided with this challenge are wild-type, knockouts, knockdowns, multifactorial perturbations, and time series data. Similarly, the InSilico_Size100 sub-challenge contains five networks with 100 nodes. The data provided are the same, except for this data set the multifactorial perturbations data is not included. All the data provided corresponds to noisy measurements of mRNA levels, in which the maximum normalised gene expression value is 1. We have chosen to use knockdowns, knockouts, multifactorial perturbations, if available, and time series data to perform the similarity calculations. In the original data set, the time series data contains five different time series for the network sized 10 and 10 time series for the network sized 100 and each time series contains 21 time points. We combined the time series information by getting average of the similarities, which are calculated independently for each time series.

Unlike the data set used in the previous section, the Dream4 in Silico Network Challenge data set does not provide information on known regulated gene relations. This leads to a change in the item selection step of our proposed method. In the item selection step, we normally use information related to genes regulated by neighbour genes. Having no such information, we modified that step into selecting regulated genes according to their similarities to neighbour genes, such that the most similar genes to the neighbour gene are chosen to be listed on the output. However, we are aware that being more similar to the neighbours does not necessarily indicate that these genes are more probable to be regulated.

In order to evaluate our method, we used the evaluation scripts and the golden data provided in the challenge website (Dream4, 2015). For scoring, the area under the precision versus recall curve, precision at 1%, 10%, 50%, and 80% recall, and the area under the ROC curve are calculated. The ranks of the teams in the challenge are decided based on the overall performance. In this report, we used the overall score to present our results as well. We used the SCORE keyword while presenting the overall score in the tables or in the figures.

As in the previous experiment, for the Dream4 in Silico Network Challenge (Dream4, 2015) data set, we need to set neighbours count (N), the output list size (k) and the threshold (T) variables. We performed tests by assigning different values to these parameters: For N and k, we assigned the range to [1, M], where M is the total number of

the nodes provided in the data, e.g., 10 or 100. However, for the InSilico _ Size100 data set, we limited the *M* to 15, as it would take long time to run the experiments up to 100 and we have observed in the previous experiments that less number of neighbours can capture the necessary information. For *T*, we used the range of [0.51, 1.00], with 0.03 increments.

After experimenting with all the combinations of *N*, *k* and *T* values, we have chosen the top performing settings. In Table 7, based on the overall score, the best five settings for the InSilico _ Size10 challenge are presented. For this result, we used the average time series similarities as the parameter $\left(b(n,c)\right)$ to be used in the regulated genes (item) selection step. Based on the table, we observed that in the best performing setting *N* is set to 9, *k* is set to 6, at least *N* many neighbours are selected (AND) and as MAX is used as the regulated genes (items) selection type. Using these settings, we performed the same calculations while using different data as the regulated genes (items) selection parameter. From Table 8, we observe that except for the knockdowns information, all the data performs equally well.

Table 7 The top 5 results for the InSilico_Size10 challenge (avg. time series)

N	*k*	*T*	*MOT*	*OLT*	*IST*	*SCORE*
	6	–	AND	F	MAX	1.119
	6	–	ND	F	MAX	1.119
	6	–	AND	F	MAX	1.001
	7	–	AND	F	MAX	0.902
	7	–	ND	F	MAX	0.902

Table 8 The results for the InSilico_Size10 challenge

Type	*SCORE*
Avg. Time	1.119
Knockdowns	1.078
Knockouts	1.119
Multifactorial pert.	1.119

We performed the similar calculations for the InSilico_Size100 challenge; i.e., we experimented with all the combination of *N*, *k* and *T* values and presented the top-performing settings. According to Table 9, the best performance is obtained when *N* is set to 11, the threshold based calculations are done with the threshold value of 0.99, the best performing method is ND, which chose exact *N* many neighbours, and the best item selection type is WAVG. In Table 10, we present the results while using 11 neighbours with ND and WAVG settings. Instead of directly using the threshold obtained while using the average time series similarity as the binding probability, we searched for the best threshold value. According to this table, best feature to be used for the binding probability is the similarity on knockouts when *T* is set to 0.99.

Table 9 The top 5 results for InSilico_Size100 challenge (Avg. time series)

N	*k*	*T*	*MOT*	*OLT*	*IST*	*SCORE*
	–	0.99	ND	T	WAVG	3.108
	–	0.99	AND	T	WAVG	3.043
	–	0.99	AND	T	WAVG	3.033
	–	0.99	ND	T	WAVG	3.023
	–	0.99	AND	T	WAVG	2.967

Table 10 The results for InSilico_Size100 challenge

Type	*N*	*k*	*T*	*MOT*	*OLT*	*IST*	*SCORE*
Avg. Time	11	–	0.99	ND	T	WAVG	3.108
Knockdowns	11	–	0.54	ND	T	WAVG	2.039
Knockouts	11	–	0.99	ND	T	WAVG	3.211

The scores and the rankings of the teams who attended the challenge are given in the challenge web-page (Dream4, 2015). The number of teams who attended the InSilico_Size10 challenge is 29 and to the InSilico_Size100 challenge is 19. Based on our best performing settings, if we were to attend the challenge, our rank would be 25 for the InSilico_Size10 challenge and 17 for the InSilico_Size100 challenge. Even though the results show that our method performs better than some other methods described in the literature, we observed that not having the information on which genes are regulated by the neighbour genes diminishes the performance of our method. Since the data sets of this challenge is produced only for the challenge and do not correspond to real world/experimentally known genes, it is not possible to map the known genes and collect information on those genes via some other resources, such as GeneMANIA. We expect to have better performance whenever the information on which genes are regulated by the neighbour genes is available.

4 Conclusions

Biological components, such as genes, proteins and metabolites, and their interactions form GRNs. Since directly observing gene interactions by experiments is very costly, recently computational methods have been used to infer the connections among these components, e.g., (Wang et al., 2014). The computational methods should take into account the general features of the GRNs, which are sparseness, scale-free topology, modularity and structurally of the inferred networks (Hecker et al., 2009).

In this work, observing the common features of recommendation systems and GRNs, we used a Pareto dominance and collaborative filtering-based recommendation method (Ozsoy et al., 2014) to predict the gene relationships, i.e., which genes regulate the others. In the original work, target users are recommended with items that are predicted to be used in the future. In this work, instead of target users we used target genes and instead of giving item recommendations to be used in the future we predicted the genes that are regulated by the target gene. For this purpose, several different settings are described. All the combination of settings are used on two different data sets. For the

evaluation, we used combination of microarray data (Spellman et al., 1998) and TF binding data (Lee et al., 2002) and Dream4 in Silico Network Challenge data set (Dream4, 2015). We observed that the use of multiple features from multiple sources improve the performance. The results show that applying a method from the recommendation systems domain to re-construct GRNs is promising.

As a future work, we want to experiment on different data sets and to include other features, if available, to observe the performance of the proposed method. Also, we anticipate that other recommendation approaches can be used in the GRN inference problem in the future. In addition, we want to explore the effect of different similarity measures on the GRN inference problem.

Acknowledgement

This research is supported by TUBITAK-BIDEB 2214/A program.

References

Bao, J., Zheng, Y. and Mokbel, M.F. (2012) 'Location-based and preference-aware recommendation using sparse geo-social networking data', *Proceedings of the 20th International Conference on Advances in Geographic Information Systems, (SIGSPATIAL'12)*, ACM, New York, NY, USA, pp.199–208.

Bernard, A. and Hartemink, A.J. et al. (2005) 'Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data', *Proceedings of the Pacific Symposium on Biocomputing*, Vol. 10, pp.459–470.

Dream4 (2015) *In Silico Network Challenge*.

Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000) 'Using bayesian networks to analyze expression data', *Journal of Computational Biology*, Vol. 7, Nos. 3/4, pp.601–620.

Gao, H., Tang, J. and Liu, H. (2012) 'Exploring social-historical ties on location-based social networks', *Proceedings of the 6th International Conference on Weblogs and Social Media*, 4–7 June, Dublin, Ireland.

Gebert, J., Radde, N. and Weber, G-W. (2007) 'Modeling gene regulatory networks with piecewise linear differential equations', *European Journal of Operational Research*, Vol. 181, No. 3, pp.1148–1165.

GeneMANIA(2015) *Genemania*.

Grzegorczyk, M. and Husmeier, D. (2008) 'Improving the structure mcmc sampler for bayesian networks by introducing a new edge reversal move', *Machine Learning*, Vol. 71, Nos. 2/3, pp.265–305.

Hecker, M., Lambeck, S., Toepfer, S., Someren, E. and Guthke, R. (2009) 'Gene regulatory network inference: data integration in dynamic models-a review', *Biosystems*, Vol. 96, No. 1, pp.86–103.

Ideker, T., Dutkowski, J. and Hood, L. (2011) 'Boosting signal-to-noise in complex biology: prior knowledge is power', *Cell*, Vol. 144, No. 6, pp.860–863.

Jaskowiak, P.A., Campello, R.J. and Costa, I.G. (2014) 'On the selection of appropriate distances for gene expression data clustering', *BMC Bioinformatics*, Vol. 15, No. 2, p.S2.

Kwon, Y-K. and Cho, K-H. (2007) 'Analysis of feedback loops and robustness in network evolution based on boolean models', *BMC Bioinformatics*, Vol. 8, No. 1, p.430.

Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I. et al. (2002) 'Transcriptional regulatory networks in saccharomyces cerevisiae', *Science*, Vol. 298, pp.799–804.

Li, J. and Zhang, X-S. (2007) 'An optimization model for gene regulatory network reconstruction with known biological information', *Optimization and Systems Biology. Lecture Notes in Operations Research*, Vol. 7, pp.35–44.

Liang, S., Fuhrman, S. and Somogyi, R. (1998) 'REVEAL, a general reverse engineering algorithm for inference of genetic network architectures', *Pacific Symposium on Biocomputing*, Vol. 3, pp.18–29.

Marbach, D., Prill, R.J., Schaffter, T., Mattiussi, C., Floreano, D. and Stolovitzky, G. (2010) 'Revealing strengths and weaknesses of methods for gene network inference', *PNAS*, Vol. 107, No. 14, pp.6286–6291.

Marbach, D., Schaffter, T., Mattiussi, C. and Floreano, D. (2009) 'Generating realistic in silico gene networks for performance assessment of reverse engineering methods', *Journal of Computational Biology*, Vol. 16, No. 2, pp.229–239.

Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D.and Califano, A. (2006) 'Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context', *BMC Bioinformatics*, Vol. 7, No. 1, p.S7.

Michailidis, G. and d'Alché-Buc, F. (2013) 'Autoregressive models for gene regulatory network inference: sparsity, stability and causality issues', *Mathematical Biosciences*, Vol. 246, No. 2, pp.326–334.

Nicolau, M. and Schoenauer, M. (2009) 'On the evolution of scale-free topologies with a gene regulatory network model', *Biosystems*, Vol. 98, No. 3, pp.137–148.

Ozsoy, M.G., Polat, F. and Alhajj, R. (2014) 'Multi-objective optimization based location and social network aware recommendation', *Proceedings of the 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2014)*, 22–25 October, Miami, Florida, USA, pp.233–242.

Ozsoy, M.G., Polat, F. and Alhajj, R. (2015) 'Inference of gene regulatory networks via multiple data sources and a recommendation method', *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2015 (Short Paper)*, pp.661–664.

Prill, R.J., Marbach, D., Saez-Rodriguez, J., Sorger, P.K., Alexopoulos, L.G., Xue, X., Clarke, N.D., Altan-Bonnet, G. and Stolovitzky, G. (2010) 'Towards a rigorous assessment of systems biology models: the dream3 challenges', *PLoS ONE*, Vol. 5, No, 2, p.e9202.

Ristevski, B. (2013) 'A survey of models for inference of gene regulatory networks', *Nonlinear Anal Model Control*, Vol. 18, pp.444–465.

Schafer, J. and Strimmer, K. (2005) 'Learning large-scale graphical gaussian models from genomic data', *Science of Complex Networks From Biology to the Internet and WWW*, Vol. 776, pp.263–276.

Shermin, A. and Orgun, M.A. (2009) 'Using dynamic bayesian networks to infer gene regulatory networks from expression profiles', *Proceedings of the 2009 ACM Symposium on Applied Computing* (SAC'09), ACM, New York, NY, USA, pp.799–803.

Shmulevich, I., Dougherty, E.R. and Zhang, W. (2002) 'From boolean to probabilistic boolean networks as models of genetic regulatory networks', *Proceedings of the IEEE*, Vol. 90, No. 11, pp.1778–1792.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. (1998) 'Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization', *Molecular Biology of the Cell*, Vol. 9, No. 12, pp.3273–3297.

Tan, M., Alshalalfa, M., Alhajj, R. and Polat, F. (2008) 'Combining multiple types of biological data in constraint-based learning of gene regulatory networks', *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, (CIBCB '08)*, pp.90–97.

Wang, Y.X.R., Jiang, K., Feldman, L.J., Bickel, P.J. and Huang, H. (2014) 'Inferring gene association networks using sparse canonical correlation analysis', *Annals of Applied Statistics Methodology*, Vol. 9, No. 1, pp.300–323.

Wessels, L.F.A., van Someren, E.P., Reinders, M.J.T. et al. (2001) 'A comparison of genetic network models', *Proceedings of the Pacific Symposium on Biocomputing*, Vol. 6, pp.508–519.

Ye, M., Yin, P., Lee, M-C. and Lee, D.L. (2011) 'Exploiting geographical influence for collaborative point-of-interest recommendation', *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, 25–29 July, Beijing, China, pp.325–334.

Zhang, W., Zang, J., Jing, X., Sun, Z., Yan, W., Yang,D., Guo, F. and Shen, B. (2014) 'Identification of candidate mirna biomarkers from mirna regulatory network with application to prostate cancer', *Journal of Translational Medicine*, Vol. 12, No. 66, p.28.

Zhang, Y., Deng, Z., Jiang, H. and Jia, P. (2007) 'Inferring gene regulatory networks from multiple data sources via a dynamic bayesian network with structural', In Sarah Cohen-Boulakia and Val Tannen, editors, *Data Integration in the Life Sciences*, Volume 4544 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, pp.204–214.

Note

1 There is another and more recent data set in Dream Challenge, namely Dream5 Network Inference Challenge. In this data set, there are four different networks and for each network only a single feature data is explicitly provided. If only a single feature is used, our proposed method reduces into traditional collaborative filtering method. In order to show the proposed method's ability to combine multiple features from multiple data sources, we preferred to use Dream4 data set.