Inference of Gene Regulatory Networks via Multiple Data Sources and a Recommendation Method

Makbule Gulcin Ozsoy
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
makbule.ozsoy@ceng.metu.edu.tr

Faruk Polat
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
polat@ceng.metu.edu.tr

Reda Alhajj
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
alhajj@ucalgary.ca

***Abstract*—Gene regulatory networks (GRNs) are composed of biological components, including genes, proteins and metabolites, and their interactions. In general, computational methods are used to infer the connections among these components. The computational methods should take into account the general features of the GRNs, which are sparseness, scale-free topology, modularity and structure of the inferred networks. In this work, observing the common features of recommendation systems and GRNs, we used a known recommendation method to predict the gene relationships, e.g., which molecules regulate others. The method we used is based on Pareto dominance and collaborative filtering. For the experiments, we used a combination of two datasets, namely microarray data and transcription factor (TF) binding data. The results show that using information from multiple sources improves the performance. Also, we observed that employing an approach from the recommendation systems domain revealed good performance.**

Keywords*—*Gene regulatory networks, Multiple Data Sources, Pareto Dominance, Collaborative Filtering

I. Introduction

Gene regulatory networks (GRNs) are composed of biological components; such as genes, proteins and metabolites, and their interactions. Since directly observing the gene relationships by experiments is very costly, recently computational methods are used to infer the connections among these components. These methods should consider the properties of GRNs, which are sparseness, scale-free topology, modularity and structure of the inferred networks [2]. GRNs are sparse, i.e., the number of connection among genes are limited; GRNs follow the power distribution function for the connectivity; they are structurally decomposable into network motifs and genes can form clusters of being highly co-expressed and/or having similar functions ([2], [6]).

We observed that GRNs has some features common to the domains usually investigated by recommendation systems and this may allow us to study GRNs by employing a recommendation system. For example, both of them are sparse and have a topology which usually follows the power distribution function [11]. Also, recommendation systems usually cover clusters of users and/or items which are mostly used to predict future preferences of the users by the recommendation methods. Observing the similarities between GRNs and recommendation systems and with the purpose of constructing GRNs using information from multiple datasets, we applied a known Pareto dominance and collaborative filtering based approach to infer GRNs. The employed method was originally used in [5] to make recommendations. For the experiments, we used microarray and transcription factor (TF) binding data ([8], [3]) as it was done in [9]. We anticipate that other recommendation approaches, which are well-known in the information retrieval and data mining literature, may be used in the future for the GRN inference problem.

The rest of this paper is organized as follows. In Section III, we present details of the method that we use to infer GRNs. In the Section IV, we will cover the evaluation process and the results. The related work is presented in Section II. Section V is the conclusions.

II. Related Work

The methods to infer (reverse engineer) GRNs commonly use Boolean networks, Bayesian networks, relevance networks, differential and difference equations [6]. In Boolean networks the gene interactions are represented as a boolean function. The aim of the reverse engineering is to find out the related boolean function for each gene [6]. [4] is an example approach using Boolean networks. The Bayesian networks are the most commonly used model to infer GRNs [6]. These networks are based on the conditional dependence of the nodes (e.g. genes), where the conditional probabilities are based on the parent nodes only. Using this feature the probability of the graph can be calculated by a joint probability distribution, which is dependent on the probability of existence of edges between nodes. [12] and [9] are example approaches that use Bayesian networks.

The use of differential and difference equations can be appropriate to infer GRNs, since the concentration of biological components changes over time [10]. The equations are based on the input gene expression data, the time, the model parameters and external effects. It aims to find the changes in the gene expression data and the relations among the genes. [10] is an example approach that is based on differential and difference equations. In relevance networks, using the similarity metrics such as Pearson coefficient or mutual information the connection among genes are decided. [7] is an example approach that is based on relevance networks.

Recently, the integration of prior knowledge and multiple types of data to the GRN inference process gained attention in the literature. [12] and [9] are example approaches that combine multiple data sources to infer the GRNs more accurately.

III. Inference of Gene Regulatory Networks

In this work we aim to predict gene bindings, such as the structure of a GRN and the direction of edges, such that which gene(s) regulates which other gene(s). Even though there can exist many different components in a GRN, such as biological markers, genes or proteins, in this paper we refer to all of them as genes. Observing the similarities between GRNs and recommendation systems, we decided to use a known Pareto dominance and collaborative filtering based recommendation approach [5], to infer GRNs. In the original approach, target users are recommended with the items, which are predicted to be preferred by the user in the future. In this work, we mapped the target users into genes and the output is mapped to the predicted genes that the target user interacts with (i.e., is expected to regulate). We decided to use this method since it is able to combine information from multiple features from multiple datasets. The method is composed of 3 main steps: Similarity calculation, neighbor selection and regulated genes (item) selection.

Similarity calculation: Similarity among genes are calculated using the features available in dataset(s). Features don't have to exist in a single dataset and features from multiple datasets can be used. In the literature there are various similarity or correlation calculation methodologies, such as Euclidean distance, Pearson correlation and Cosine similarity. In this work we preferred to use Cosine similarity as in Equation 1.

In Equation 1, genes are shown as *A* and *B*. Genes can have multiple features and each feature is indicated by the subscript *i*. The subscript *j* indicates the values of each feature.

$$sim(A_i, B_i) = \frac{\sum_{j=1}^{n} A_{ij} \times B_{ij}}{\sum_{j=1}^{n} A_{ij}^2 \times \sum_{j=1}^{n} B_{ij}^2} \tag{1}$$

Neighbor selection: The neighbors are the ones that behave most similar to the target gene. Knowing the neighbor genes and their connections in the graph, the connections of the target gene can be predicted. In order to decide on the most representative neighbors, the similarity values calculated in the previous step and the Pareto dominance relation are used (Equation 2). In the equation g_i and g_j represent genes and f indicates the different 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. At the end, the non-dominated genes are assigned as the neighbors of the target gene. In order to collect as many neighbors as predefined, an iterative process of neighbor collection is applied, as explained in [5]

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

Regulated genes (Item) selection: The genes to which the target gene has a connection are decided by using collaborative filtering. In this process, the known connections of neighbor genes are used to decide on the best matching gene to predict for the target gene to regulate, such that the target gene regulates the predicted genes. The genes which are already known to be regulated by the neighbor genes are assigned as the candidate genes. For each candidate gene, a connection score is calculated by Equation 3. A higher connection score indicates that the candidate gene is more promising to be regulated by the target gene. In Equation 3, the *score* represents the connection score, *t* represents the target gene, *n* represents the neighbor gene and *c* represents the candidate gene. In the calculation the similarity among the target and the neighbor genes ($sim(t, n)$) and the binding probabilities of the neighbor and candidate genes ($b(n, c)$) are used.

$$score(c) = \sum sim(t, n) \times b(n, c) \tag{3}$$

In each step, different settings can be used. The explanations and the abbreviations of these settings are given as follows:

Multi-Objective Optimization Type (MOT): This setting is related to the step of neighbor selection. The number of neighbors to select can be decided by different settings. Only_Dominates (OD): Find non-dominated neighbors 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 neighbors by running multiple iterations and pruning when necessary. At_Least_N_Dominates (AND): Find at least N neighbors by running multiple iterations. Unlike the N_Dominates setting, no pruning is applied in this setting.

Regulated genes (Item) Selection Method Type (IST): This setting is related to the regulated genes (item) selection step. Different settings may assign different values to the parameters in Equation 3. Sum (SUM): The similarities between the target and the neighbor genes are not considered, such that $sim(t, n) = 1$ for all neighbors. Average (AVG): After summing up values -as done in the SUM method-, the result is divided by the number of neighbors 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 neighbor genes. Weighted Average (WAVG): AVG method is performed by additionally using the similarities among the target and the neighbor genes. So instead of dividing the summation into the number of neighbor genes, it is divided into the summation of the similarities between the target and the neighbor genes. In the application, we used binding probabilities between the target and neighbor gene, instead of calculating similarities, such that $sim(t, n) = b(t, n)$.

IV. Testing and Evaluation

In order to evaluate the performance of the described methods we used precision@k, recall@k and f1-measure, which are the commonly used metrics in the literature. For the evaluation we used the same datasets that were used in [9], which are microarray data from Spellman et al. [8] and transcription factor (TF) binding data from Lee et al. [3]. The first dataset [8] contains time series gene expression data, in which there are 6178 genes and 77 time steps. In [1] these time steps are divided into three phases. In this work, we used each

phase as a different feature, rather than using each time step as a feature. The second dataset [3] contains binding location data of 6270 genes and 106 TFs. Even though the datasets contain many more genes, in [9] 25 of them are chosen based on the studies described in [1]. Following that work, we also worked on the same 25 genes. Also we executed the same pre-processing steps performed in [9]: Filling the missing values in the microarray dataset, converting p-values in the binding data into probability values and filling missing probabilities in the binding data. In [9] a commercial tool is used to collect the golden data. Unlike them, we preferred to use a public tool named as GeneMANIA. GeneMANIA provides various information based on interaction types: Genetic interactions, Co-localization, Co-expression, Physical interactions, Shared protein domains, other. We collected information for the selected 25 genes for all the interaction types from GeneMANIA on March 10, 2015. If not explicitly stated otherwise, we presented the average of all interaction types as the evaluation results.

We combined the microarray data [8] and the binding data [3] in two different ways. In the first, we used the three phases extracted from the microarray data are used for the similarity calculations step, and the binding data is used for the regulated genes (item) selection step. In the second method, we added the binding data to the similarity calculations step too. As a result, for the first experimental setting, we used three features and for the second settings we used four features. In the following paragraphs we will refer these setting as 3F_Experiment and 4F_Experiment, respectively. In the experiments, we need two variables to be assigned; neighbors count (N) and the output list size (k). The performance of the methods may differ based on these parameters. We performed tests by assigning different values to these parameters: For N, we assigned the range to [1,25], where 25 is the total number of the genes. Similarly for k, we assigned the range to [1,25], where 25 is the total number of the genes. We present the best results for each regulated genes (item) selection method type (IST), using 3 or 4 features (3F_Experiment or 4F_Experiment). We collected information of the best methods and parameters for each evaluation metric;precision, recall and f1-measure. From the experimental results we observed that the f1-measure and the recall favor the selection of many neighbor genes; e.g. 22 out of the 25 genes; and predicting too many, nearly all, genes as being regulated by the target gene. Since it is known that the GRNs are sparse, this tendency does not seem to be correct. So, we decided to use precision as our main objective in the rest of this paper.

In Table I, we present the best results for the precision with the 3F_Experiment and 4F_Experiment, in the order of the sections seen in the table.. According to the table, the best performing method for 3F_Experiment is the one that chooses N many neighbors (ND) when using different gene selection approaches. In the best setting for 3F_Experiment the number of neighbor genes to be selected is 12 and the output list size is set as 2. For 4F_Experiment, the best performing method in terms of precision is the one that chooses N many neighbors (ND) when using MAX as the item selection approach. In this setting, the number of neighbor genes to be selected is 3 and the output list size is set as 1. For recall and f1-measure, the best performing method is the one that chooses at least N many neighbors (AND) when using WAVG as the item selection approach. In this setting the chosen N and k values are 5 and 2, respectively. Comparing the performances of 3F_Experiment and 4F_Experiment, we observe that adding the binding data for the similarity calculations increases the performance slightly.

TABLE I: The best results for the precision and with the 3F_Experiment and 4F_Experiment

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

Our method provides the directed graph, such that it predicts which gene regulates the others. However, in [9] the only graph provided in the paper is undirected. Since we want to compare our result to theirs, we also converted our directed graph into undirected by adding the reverse directions of the edges to the graph. In Table II, we present the results for 3F_Experiment, 4F_Experiment and [9], in the order of the sections seen in the table. According to the table, the best method in terms of precision is the one that uses N many neighbors with MAX as the item selection method. For recall the best method is the one that chooses N neighbors by using SUM or AVG as the item selection method. For f1-measure, the best method uses ND with weighted average approach. For all of the measures, the best results belong to the 4F_Experiment setting. We can conclude that adding the binding data to the similarity calculations step increases the performance.

TABLE II: The results for the undirected graph

N	k	MOT	OLT	IST	Prec.	Recall	F1
1	1	ND	F	SUM	0.275	0.124	0.163
1	1	ND	F	AVG	0.275	0.124	0.163
5	1	AND	F	MAX	0.333	0.098	0.146
1	1	ND	F	WAVG	0.275	0.124	0.163
6	4	AND	F	SUM	0.248	**0.470**	0.299
6	4	AND	F	AVG	0.248	**0.470**	0.299
10	1	ND	F	MAX	**0.342**	0.085	0.132
6	4	ND	F	WAVG	0.250	0.464	**0.300**
-	-	-	-	-	0.213	0.193	0.203

We observed from the tables that for directed graphs the weighted average (WAVG) method for choosing genes works better when we consider all the measures; i.e. precision, recall, f1-measure. For undirected graph, there is no single winner for the item selection method. Also we observed that using exactly N many neighbors (ND) mostly performed better than the other approaches.

Lastly, we decided the best values for N and k. In Figures 1 and 2, we present the plot of precision values for different N and k values for the experiments using three or four features. For both experiments, we observe that the increase in k decreases the precision. For the 3F_Experiment, the best

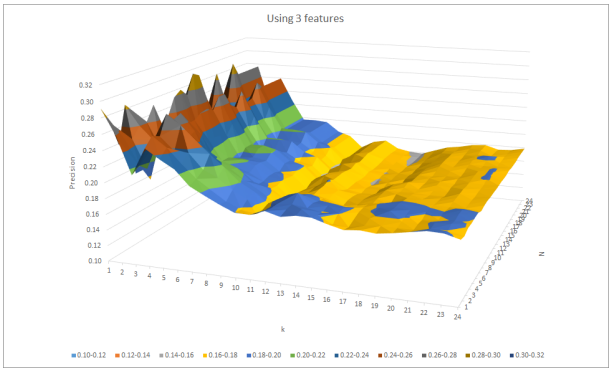

Fig. 1: The precision results for ND and WAVG for different N and k (3F_Experiment)

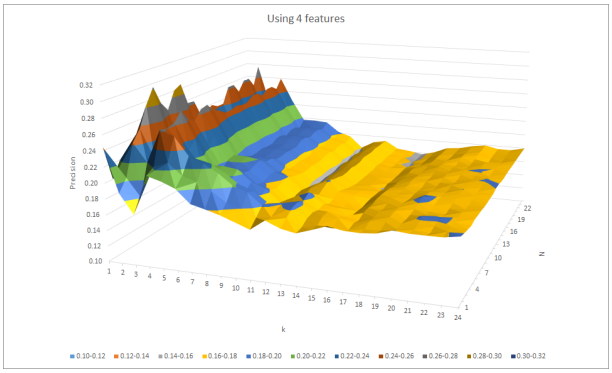

Fig. 2: The precision results for ND and WAVG for different N and k (4F_Experiment)

precision is obtained when N is set to 12 and k to 2. For the 4F_Experiment, the best precision is obtained when N is set to 6 and k to 2. Even though the performance results of both experiments are similar, adding the binding data to the similarity calculations (i.e. 4F_Experiment) helps the system to reduce the calculations by decreasing the necessary number of neighbors to choose. Note that for both experiments the number of genes to be regulated by the target gene (i.e., k) is found to be 2, which is a small value. This observation matches with the sparsity feature of GRNs.

V. Conclusion

Gene regulatory networks (GRNs) are composed of biological components; such as genes, proteins and metabolites, and their interactions. There are many approaches in the literature that aim to infer (reverse engineer) these interactions computationally. These methods should take into account the general features of the GRNs, which are sparseness, scale-free topology, modularity and structurality of the inferred networks [2].

In this work, observing the common features of recommendation systems and GRNs, we used a known recommendation method to predict the gene relationships, such as which genes regulate the others. The method we used is based on Pareto dominance and collaborative filtering and it was originally presented in [5] to give recommendations to the target users; thus it has been proved successful in a domain other than GRN reconstruction. For the experiments, we used a combination two different datasets. The results show that using information from multiple sources improves the performance. Also, we observed that the use of an approach from recommendation systems performs well. We anticipate that other recommendation approaches can be used for handling the GRN inference problem in the future. In the future, we want to apply the method presented in this paper on other biological datasets. Also, we want to use other known recommendation methods on GRN inference problem.

Acknowledgment

This work is supported by TUBITAK-BIDEB 2214-A.

References

[1] A. Bernard, A. J. Hartemink *et al.*, "Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data." in *Pacific symposium on biocomputing*, vol. 10. World Scientific, 2005, pp. 459–470.

[2] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke, "Gene regulatory network inference: Data integration in dynamic modelsa review," *Biosystems*, vol. 96, no. 1, pp. 86 – 103, 2009.

[3] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon *et al.*, "Transcriptional regulatory networks in saccharomyces cerevisiae," *science*, vol. 298, no. 5594, pp. 799–804, 2002.

[4] S. Liang, S. Fuhrman, and R. Somogyi, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures," in *Pacific Symposium on Biocomputing*, vol. 3, 1998, pp. 18–29.

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

[6] B. Ristevski, "A survey of models for inference of gene regulatory networks," *Nonlinear Anal Model Control*, vol. 18, pp. 444–465, 2013.

[7] J. Schafer and K. Strimmer, "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, 2005.

[8] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "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, 1998.

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

[10] L. F. Wessels, E. P. van Someren, M. J. Reinders *et al.*, "A comparison of genetic network models." in *pacific Symposium on Biocomputing*, vol. 6, no. 4, 2001, pp. 508–519.

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

[12] Y. Zhang, Z. Deng, H. Jiang, and P. Jia, "Inferring gene regulatory networks from multiple data sources via a dynamic bayesian network with structural em," in *Data Integration in the Life Sciences*, ser. Lecture Notes in Computer Science, S. Cohen-Boulakia and V. Tannen, Eds. Springer Berlin Heidelberg, 2007, vol. 4544, pp. 204–214.