Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Co-Expressed Gene Assessment Based on the Path Consistency Algorithm: Operon Detention in *Escherichia coli*

Shigeru Saito
Division of Chem&Bio Informatics
INFOCOM Corporation
Tokyo 150-0001, Japan
sh.saito@infocom.co.jp

Katsuhisa Horimoto
Computational Biology Research Center
National Institute of Advanced Industrial Science and Technology
Tokyo 135-0064, Japan
k.horimoto@aist.go.jp

***Abstract*—The assessment of co-expressed genes is an initial step to investigate the gene regulatory system in a microarray analysis. We designed a simple method to assess co-expressed genes from the expression profiles, based on the path consistency (PC) algorithm, which systematically infers a causal graph. The PC algorithm is simply modified for the frequent occurrence of high correlations between expression profiles, which causes an accidental stop of the algorithm by the violation of a numerical calculation. By utilizing the flexible features of the algorithm, biological information about co-expressed genes can be introduced into the algorithm. The performance is illustrated by the application to the gene profiles of operons: 1168 gene profiles in 137 known operons in *Escherichia coli*. By the improved algorithm, more than 80% of the operons were correctly detected, and less than one gene per operon was falsely detected. The performance promises the wide feasibility of the present method for the assessment of co-expressed genes from microarray data on a genomic scale.**

***Keywords*—gene expression profile, co-expression, gene regulation, graphical model, network inference**

I. INTRODUCTION

The assessment of co-expressed genes from microarray data is one of the useful approaches to investigate gene regulatory system [1]. For example, a transcriptional factor is expressed coordinately with the genes regulated by the factor. Furthermore, a number of recent studies demonstrated that genes co-expressed across multiple conditions are more likely to represent common functions than would be expected by chance alone [2,3]. The general tenet is that genes encoding proteins participating in a common pathway will display correlated expression levels when analyzed on a sufficient scale [4-7]. Thus, the assessment of co-expressed genes from microarray data is the basis for investigating the cellular functions of genes as well as their regulatory systems.

Most of the methods to assess co-expressed genes from microarray data begin with a similarity measure that describes the degree to which the expression levels between pairs of genes are similar across multiple conditions [1]. The matrix of similarity across the microarray, typically representing the pairwise similarity of the expression patterns of thousands of genes, is the starting point from which the genes can be organized into clusters. Clustering involves a wide variety of algorithms for dividing genes into groups with approximately similar expression patterns [8]. However, there are several important limitations to most of the clustering algorithms that are in conflict with the reality of biology. For example, the analyzed genes are assigned to only one cluster: an artificial limitation is placed on the biology under study, in that many genes play important roles in multiple, but distinct, relationships. As an alternative to assigning genes to clusters, the pairwise similarity is estimated by setting a threshold to create a graph comprised only of edges with similarities that exceed the threshold. For example, Allocco and colleagues originally described such graphs as relevance networks [9], in which both positive and negative correlation coefficients exceeding a specified threshold are displayed graphically, allowing visual recognition of highly connected subsets of genes. Recent studies have mined relevance networks to extract co-expressed genes by the pairwise relationships that could be extracted manually from the graphs in cancer cells [10,11] and myopathic muscle biopsies [12], and by the cliques in the graph that could be computationally identified to find the highly connected subsets of genes in relevance networks in the spleens of mice exposed in vivo to low-dose ionizing radiation (IR) [13]. Although the relevance network approach overcomes the pitfalls of clustering approaches, such as a single relation, the correlation coefficient, which is utilized at the initial step, is essentially a measure between two variables. As well known in statistics, Simpson's paradox illustrates the risk for the pseudo-correlation between two variables, when the two variables are correlated with the other variables [14]. Thus, in terms of the existence of multiple relationships between the genes, it is desirable that a basic correlation between two genes itself is estimated by carefully considering the influence of the remaining genes.

A partial correlation coefficient quantifies the correlation between two variables, when conditioning on one or several other variables. For example, the first order of a partial correlation coefficient, $r_{xy,z}$, between variables x and y

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

conditioning on z means the correlation between the parts of x and y that are uncorrelated with z. In other words, the partial correlation coefficient express the correlation between x and y without any influence of z. The order of the partial correlation coefficient is determined by the number of variables that it is conditioned on, and any arbitrary n-th order of a partial correlation coefficient can be formulated by using the (n-1)-th order of a partial correlation coefficient.

Owing to the nature of a partial correlation coefficient for a complex relationship between multiple variables, the partial correlation coefficient has been utilized as a basic correlation to infer gene regulatory networks from microarray data by various approaches. We have previously developed a method for inferring the association between many genes [15, 16], based on the graphical Gaussian model (GGM) [17]. In the GGM, the higher order of a partial correlation coefficient, in which this order is equal to the total number of variables, is calculated from the inverse matrix of the correlation coefficient matrix. Unfortunately, the calculation of the higher order of the partial correlation coefficient is frequently difficult when applying it to infer a large scale network between many genes. This is because the correlation coefficient matrix is not regular, due to the existence of many genes with similar expression patterns. Thus, we have designed a method to utilize the GGM to infer a framework of gene associations by a combination with the clustering of microarray data in the previous study [15, 16], although its resolution degree was naturally low, due to the clustering. In the following studies, the difficulty in the calculation of the higher order partial correlation coefficient has been carefully treated: methods for calculating the partial correlation coefficients from lower to higher orders [18] and for using a combination of the Moore–Penrose pseudo-inverse and Robbins–Efron-type inferences [19] have been designed. However, some restraints remained in the two methods: the former, sophisticated method has been designed for a sparse network, and the latter, simple method needs an *ad hoc* setting for the order of the partial correlation method to estimate the network.

We have developed a method for the assessment of co-expressed genes, along the efforts for the network inference based on the partial correlation coefficient. Our method partially exploits the path consistency (PC) algorithm [20]. The PC algorithm is one of the algorithms that infer causal relationships between variables, and is composed of the construction of an undirected independence graph (UIG), based on the partial correlation coefficient, and the estimation of a causal relationship from the constructed UIG, according to the orientation rule [21]. Although the UIG part of the PC algorithm still faces the difficulty in calculating the partial correlation coefficient in some cases, this part has merits, such as the natural completion of the algorithm in a reasonable amount of computational time, depending on the analyzed data. Thus, we utilized the UIG part to assess the co-expressed genes from the microarray data: the genes that are composed of a network are regarded as a set of co-expressed genes.

We examined the performance of our method by using a set of microarray data that was measured in *Escherichia coli*, and by detecting the constituent genes of operons as a benchmark test for the assessment of co-expressed genes. The operons are one of the most trusted gene sets, in which multiple open reading frames are transcribed from the same promoter into a single mRNA transcript, and therefore the genes are generally transcribed at the same levels. Indeed, some methods illustrated their performance by analyzing the operons in *E. coli* [22, 23]. Although the performances of their methods are extremely high, the previous methods were specialized to detect the operons, and thus depended on various data specific to the operons, such as the sequence length of the upstream regions, except for the expression degree of the genes. In contrast to the previous methods, the present method requests the least amount of information about operons, to test the performance for generally assessing the co-expressed genes.

II. MATERIALS AND METHODS

A. Path Consistency (PC) Algorithm

The path consistency (PC) algorithm [20] is an algorithm to infer a causal graph that is composed of two parts: the undirected graph inference by a partial correlation coefficient and the following directed graph by using the orientation rule. Since the present method partially exploits the first part of the PC algorithm for the assessment of the co-expressed genes, here, we briefly describe the initial steps for the corresponding part in the PC algorithm.

Step 0: preparation of graph

- Prepare a complete undirected graph of $C = (V, E)$, where $V=\{X_1, X_2, \ldots, X_m\}$.

Step 1: the zeroth-order of the partial correlation coefficient

- Select any ordered pairs of vertices $\{X_i, X_j\}$ that are adjacent in C, and calculate the zeroth-order of the partial correlation coefficient, r_{ij} (the Pearson's correlation coefficient), between X_i and X_j.
- Examine the independence between X_i and X_j, and delete an edge between X_i and X_j, if X_i and X_j are independent. The graph C is updated, and the variable pairs with the independence are restored in the subset S.
- If no ordered pairs of vertices $\{X_i, X_j\}$ that are adjacent in the updated C are found, then the algorithm stops; otherwise, go to the next step.

Step 2: the first-order of the partial correlation coefficient

- Select any ordered pairs of vertices $\{X_i, X_j\}$ that are adjacent in C, except for the variables in S, and calculate the first-order of the partial correlation coefficient, $r_{ij, k}$, between X_i and X_j given X_k.
- Examine the partial independence between X_i and X_j given X_k, and delete an edge between X_i and X_j, if X_i and X_j are partially independent, given X_k. The graph C is updated, and the variable pairs with partial independence are restored in S.
- If no ordered pairs of vertices $\{X_i, X_j\}$ that are adjacent in C are found, then the algorithm stops; otherwise, go to the next step.

Step 3: the second-order of the partial correlation coefficient

- Select any ordered pairs of vertices $\{X_i, X_j\}$ that are adjacent in C, except for the variables in S, and calculate the second-order of the partial correlation coefficient, $r_{ij,kl}$, between X_i and X_j, given X_k and X_l.
- Examine the partial independence between X_i and X_j, given X_k and X_l, and delete an edge between X_i and X_j, if X_i and X_j are partially independent, given X_k and X_l. The graph C is updated, and the variable pairs with partial independence are restored in S.
- If no ordered pairs of vertices $\{X_i, X_j\}$ that are adjacent in C are found, then the algorithm stops; otherwise, go to the next step.

In general, the (m-2)-th order of the partial correlation coefficient is calculated between two variables, given (m-2) variables, i.e., $r_{ij,rest}$, between X_i and X_j, given the '*rest*' of the variables, $\{X_k\}$ for k=1, 2, …, m, and $k \neq i$, j, and after calculating the (m-2)-th order of the partial correlation coefficient, the algorithm naturally stops. However, the algorithm does not usually request the (m-2)-th order of correlation coefficient for the natural stop. This is because adjacent variables after excluding the variables in the subset S are often not found, even in the calculation of the lower orders of partial correlation coefficients.

B. *Test for the correlation from actual data in the PC algorithm*

In the application of the PC algorithm to actual data, we test the correlation between the variables from the actual data, for judging the independence between variables by using the standard statistical test. Fortunately, t test can be applied to both the zeroth-order of the partial correlation coefficient for the independence and the higher-orders of the partial correlation coefficients for the partial independence [24].

The t is obtained from the correlation coefficient as follows,

$$t = \frac{r\sqrt{df}}{\sqrt{1-r^2}} \tag{1}$$

where r and df are the correlation coefficient and the degree of freedom, respectively. In the zeroth-order of the partial correlation coefficient, r is Pearson's correlation coefficient, r_{ij}, expressed by

$$r_{ij} = \frac{\mathrm{cov}(X_i, X_j)}{\sqrt{\mathrm{var}(X_i)\,\mathrm{var}(X_j)}} \tag{2}$$

where $cov(X_i, X_j)$ and $var(X_i)$ are the covariance between X_i and X_j, and the variance of X_i. In the higher order of the partial correlation coefficients, r is the partial correlation coefficient, $r_{ij,rest}$, expressed by

$$r_{ij,rest} = \frac{-r^{ij}}{\sqrt{r^{ii} \cdot r^{jj}}} \tag{3}$$

where r^{ij} is the i-j element of inverse correlation coefficient matrix. Note that the dimension of the correlation coefficient matrix corresponds to the orders of the partial correlation coefficients. For example, the first order of the partial correlation coefficient is calculated from three variables, and the dimension of the correlation coefficient matrix is three. In general, the n th-order of the partial correlation coefficient is calculated from the (n+2) dimension of the correlation coefficient matrix. As for the degree of freedom, each degree of freedom, df, in the zeroth- and higher-order of the partial correlation coefficients is expressed by n - 2 and n - p, respectively, where n and p are the number of samples and the number of variables.

C. *A simple modification for analyzing the gene expression profiles in the PC algorithm (mPCA)*

In the actual expression profile data, many genes frequently show profiles with similar patterns. This makes the numerical calculation of correlation coefficients difficult, due to the multi-colinearity between the variables. Indeed, the correlation coefficient matrix is not always regular, due to many gene expression profiles sharing similar expression patterns on a genomic scale. The original PC algorithm accidentally stops, if only one correlation between a pair of variables shows a violation of the numerical calculation. However, in a biological sense, the gene pairs that cause the accidental stop can be interpreted as the case when they are highly associated with each other, in terms of gene expression. Thus, we will modify the PC algorithm to prevent it from accidentally stopping with the highly associated gene pairs.

We simply modify the original PC algorithm as follows. If the calculation of any order of the partial correlation coefficient between the variables is violated, then the corresponding pair of variables is regarded as being dependent. For example, if the first-order correlation coefficient, $r_{ij,k}$, cannot be calculated numerically due to the multi-colinearity between X_i and X_j, then keep the edge X_i-X_j without the statistical test. The other parts remain unchanged in the modified algorithm. Note that the above modification ensures the natural stop of the algorithm for the data including the high correlation. Hereafter, the modified algorithm above is named *mPCA*.

D. *Introduction of biological priors into mPCA (mPCA+)*

One of the merits of the present method is that biological information can be easily introduced into the *mPCA* as the prior. As for the operon detection in the present study, we can introduce three types of information about the operons as the priors. The first type of biological information is that the number of constituent genes in one operon is limited, from the fact that the observed operon is composed of less than twenty genes [25]. According to the definition of an operon, the second is that all of the constituent genes in each operon are coded in the same strand, and the third is that all constituent genes are adjacent to each other. Note that the above three types of information are general features of operons. Thus, the present introduction needs no preliminary survey, such as the sequence length distribution of upstream regions in operons, depending on the bacterial genomes analyzed.

The first type of information is realized by partially replacing the observed correlation coefficients with zero

values, which means that the corresponding edges are removed at any level of significance probability. In the present study, the elements that were 20 elements apart from the diagonal in the correlation coefficient matrix were set to zero values, i.e.,

$$\begin{cases} c_{ij} = r_{ij} & \text{when } |i-j| \le 20 \\ c_{ij} = 0 & \text{when } |i-j| > 20 \end{cases}$$

where c_{ij} represents the *i-j* elements of the correlation coefficient matrix generated for calculating the higher order of correlation coefficients. In the introduction of the second type of information, first, the two data sets of genes coded in the respective strands were generated, and then the two outputs separately obtained from the data sets were combined into one output. The third type of information is realized by simply excluding the disordered genes from the outputs; if the genes in one inferred network are not adjacent, then the corresponding edges are removed. Note that the introduction of the three types of information into the algorithm is ordered when operating the algorithm. The first two priors are set before operating the algorithm, while the last prior is set after doing it. Hereafter, the algorithm, in which the three priors were further introduced into *mPCA*, is referred to as *mPCA+*.

E. Gene expression and operon data

The gene expression profile data analyzed in the present study are listed in TABLE I. We compiled the expression profiles measured for all genes in Escherichia coli from eight experiments [26-33], and in total, the numbers of genes and measured points are 4289 and 178, respectively, in the present data set. To correct for the different magnitudes of the profiles between the measured conditions, the profiles were standardized by the average and the standard deviation for each condition.

TABLE I. EXPRESSION PROFILE DATA SETS USED IN THE PRESENT STUDY

Condition	Number of measued conditions	Reference
M9+glucose and LB medis	16	26
Degradosome mutants	78	27
UV irradiation	15	28
Tryptophan regulation	27	29
DNA gyrase and topoIV regulation	21	30
RraA regulation	7	31
RnaG regulation	10	32
Adaptation to famine	4	33

TABLE II. GENE FREQUENCY IN OPERONS OF THE PRESENT DATA SET

No. of genes in operon	2	3	4	5	6	7	8	9	10	11	12	13	15
No. of operons	169	81	50	35	13	8	6	5	3	2	1	1	3

The information on the operons was cited from EcoCyc [25], and the operons that are composed of more than two genes with profiles within the data set were selected. The number of operons thus selected was 377, and the total number of constituent genes was 1163. The distribution of the constituent gene numbers of each operon is listed in TABLE II. As seen in the table, about half of the operons are composed of two genes, and the maximum number of constituent genes is 15 in one operon. Although the maximum number in the present data set is less than that set as the prior in the preceding section, the present prior is set to detect a co-expressed gene set composed of more genes than those of known operons.

III. RESULTS AND DISCUSSION

A. Application of the PC algorithm

As expected from the high correlations between the expression profiles, as usual, the UIG part in the PC algorithm stopped during its application to the data set analyzed in the present study. This is because highly similar profiles show multi-colinearity between the expression profiles. Indeed, the correlation coefficients could not be calculated in 111 cases (data not shown). Thus, the modification for the high correlation shown in 2.2 was needed for detecting co-expressed genes from the expression profiles in the present study. The word “data” is plural, not singular.

B. Numbers of connected edges by mPCA

We counted the numbers of edges detected by *mPCA* at the different levels of significance probabilities, before evaluating the detection accuracy of the present method. Fig. 1 is the plot of the numbers of edges by *mPCA*, against the orders of correlation coefficients. By the simple modification described in 2.2, *mPCA* naturally stopped by completing the calculation for higher orders of correlation coefficients. In the significance levels ranging from 10^{-1} to 10^{-8}, the calculations in the respective levels stopped from the third-order of correlation to the seventh-order of correlation. As seen in the figure, the number of edges decreased most drastically from the zeroth (the range from $10^{4.6}$ (≈40,000) to $10^{5.6}$ (≈400,000)) to the first-order correlations (the range from $10^{2.8}$ (≈700) to $10^{4.5}$ (≈36,000)), while the decrease in the degree of edge numbers from all connected cases to the zeroth-order correlation is relatively small (675,703 (=1163*1162/2) to about 550,000 edges). In addition, the numbers of edges in the first-order correlation depend on the significance levels in the range from $10^{2.8}$ (≈700) to $10^{4.5}$ (≈36,000), while those in the zeroth-order correlation ranged from $10^{4.6}$ (≈40,000) to $10^{5.6}$ (≈380,000). Thus, the calculation of the partial correlation coefficient is highly effective to estimate the degree of gene associations. In the orders higher than the second order of correlations, the numbers of edges decrease gradually. Finally the calculations stopped, depending on the significance level: the third order at 10^{-8}, 10^{-7}, 10^{-6} and 10^{-5}, the fourth order at 10^{-4} and 10^{-3}, the sixth order at 10^{-2}, and the seventh order at 10^{-1}.

Here, we can estimate a possible range for the edge numbers in the constructed graphs, with reference to the numbers of constituent genes in the analyzed operons in TABLE II. Assuming that only the constituent genes in the operons are connected with each other, the maximum number of edges is calculated to be 2477, when the constituent genes in each operon are mutually connected (complete graph), and the minimum number of edges is calculated as 928, when the

constituent genes are linearly connected (chain graph). As seen in Fig. 1, the numbers of edges when the algorithm naturally stopped were in the ranges of possible numbers of connected edges at the significance levels of 10^{-2} to 10^{-5}. Thus, we will consider these significance probability levels in the following study.

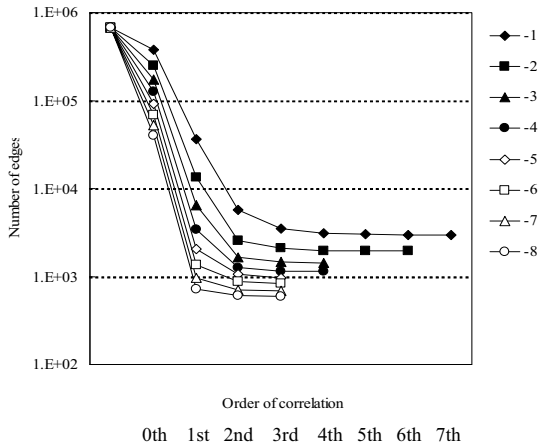

Figure 1. Number of edges established by *mPCA* until natural stops of estimation. We performed *mPCA* at significance levels from 10^{-1} to 10^{-8}. The calculations stop in various orders of correlations, depending on the above levels. The number of possible edges connected between all constituent genes, 675,703 ($=_{1163}C_2$), is also plotted at the start points of each graph.

C. Detection accuracy by mPCA

We evaluated the accuracy for detecting the operons by *mPCA*. For this purpose, first, the correctly detected operons were counted in the significance probability levels from 10^{-2} to 10^{-5}. In this count, an operon was regarded as a 'correctly detected' operon, when the edges were connected between more than two genes in the constituent genes of each operon. Then, the fraction of the number of correctly detected operons to the total number of operons was calculated as the accuracy for the correctly detected operons. In addition to this operon accuracy, two values of gene detection accuracy were further estimated in the correctly detected operons: one is the average ratio of the number of correctly detected genes to the number of constituent genes per operon, and the other is the average number of genes that were falsely connected with the operon genes.

TABLE III. DETECTION ACCURACY IN APPLICATION TO KNOWN OPERONS IN *E. COLI*

	Significance level			
	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Fraction of correctly detected operons	0.63	0.57	0.54	0.51
Ratio of correctly detected genes	0.85	0.82	0.81	0.79
Number of falsely detected genes	8.41	5.75	4.33	3.51

TABLE III shows the three values of detection accuracy. As seen in the table, the fractions of correctly detected operons ranged from 0.51 to 0.63 at four levels of significance probabilities. As for the gene accuracy in the correctly detected operons, the average ratios of the number of correctly detected genes showed high values, about 0.80, and the average numbers of falsely detected genes for each operon ranged from 3.5 to 8.5 at the four levels. While the ratios of the number of correctly detected genes were relatively high, the numbers of falsely detected genes were large; the constituent genes of one operon were falsely connected with the genes of the other operons. This situation causes the relatively low accuracy of operon detection. Thus, the reduction of the falsely connected operons will be effective to raise the operon detection accuracy, by considering some priors.

D. Detection accuracy by mPCA+

We evaluated the accuracy for detecting the operons by *mPCA*+, in which the three priors described in 2.3 were introduced into *mPCA*. In addition to the detection accuracy, the detection accuracies by introducing each of the three priors were also calculated.

Fig. 2A shows the fractions of correctly detected operons by *mPCA*+. The fractions of correctly detected operons by *mPCA*+ were much higher than those detected by *mPCA* in TABLE III. Indeed, the fractions detected by *mPCA*+ were in the range from 0.83 to 0.92, at the four significance probabilities, while those by *mPCA* ranged from 0.51 to 0.63. Thus, the introduction of the three biological priors in *mPCA*+ was strikingly effective to improve the accuracy for detecting the constituent genes in operons. Furthermore, the respective effects of the three priors on the accuracy in operon detection were also evaluated in Fig. 2A. Among the three priors, the number of constituent genes was the most effective for the accuracy. Indeed, the fractions rose from ca. 0.6 to more than 0.8, by the introduction of the constituent gene number into *mPCA* as a prior. The introduction of the coding strand was also effective for the operon detection; the fraction was raised by ca. 0.05. In contrast, the adjacency of constituent genes had little effect on the detection accuracy. Thus, the three priors showed different effects on the operon detection accuracy. The respective effects will be further investigated below, by evaluating the gene detection accuracy in the correctly detected operons.

Fig. 2B shows the average ratios of the correctly detected genes at different levels of significance probabilities. The average ratios by *mPCA*+ reached ca. 0.95, indicating the striking effects of prior introduction on the detection accuracy. As seen in the figure, the introduction of the three priors also showed distinctive effects in raising the accuracy. Interestingly, the overall effects of the different priors on the gene detection accuracy were very similar to those on operon detection accuracy in Fig. 3A. Indeed, the effect increased in the order of the number of constituent genes, the coding strand, and the continuity of the constituent genes. As for the effect on the correctness, the prior on the number of constituent genes was highly effective, while in contrast, the prior on the adjacent genes was minimally effective.

The detection accuracy by *mPCA+* was drastically improved with the falsely detected genes. The average numbers of false genes detected by *mPCA+* were suppressed to about 0.5 genes per operon in the four significance probabilities in Fig. 2C, while those by *mPCA* ranged from ca. 4 to 8 genes per operon. Interestingly, the effects of the three priors on the false gene number in Fig. 2C are different from those on the correctly detected genes in Fig. 2B. The adjacency of the constituent genes was the most effective prior to exclude the falsely detected genes, while the number of constituent genes and the coding strand were much less effective. Thus, the prior for the operon structure is effective for the exclusion of falsely detected genes.

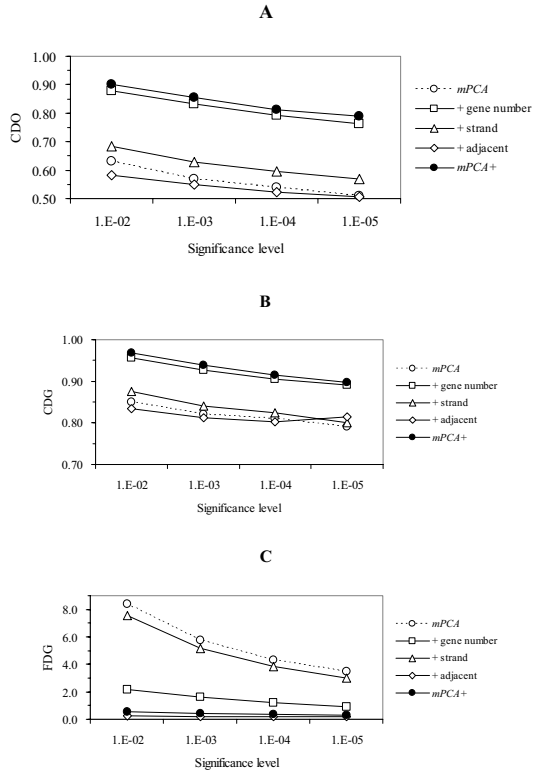

Figure 2. Detection accuracy by *mPCA+*, together with the effects of the three priors on the detection accuracy. The detection accuracies in the three criteria were calculated in the significance probabilities from 10^{-2} to 10^{-5}, respectively: A, the fraction of correctly detected operons (CDO); B, the ratio of correctly detected genes (CDG) per operon; and C, the number of falsely detected genes (FDG) per operon. For reference, the detection accuracies by *mPCA* are also plotted.

In summary, the introduction of the three priors into *mPCA* greatly improved the detection accuracy. Furthermore, the prior for the number of constituent genes was the most effective to detect the operons correctly, and to exclude the falsely detected genes, the prior on the adjacency of the genes in the operons was the most effective. More generally, the prior on the space for searching the relationship between the genes is responsible for the correctness, and that on the structural features for the target is for the falseness, in terms of the co-expressed gene assessment.

IV. Conclusions

We designed a simple method to assess co-expressed genes from the expression profiles, based on the PC algorithm. We modified the algorithm for the high correlation due to accidental stops, and the modification enabled us to utilize the beneficial properties of the PC algorithm, such as the selection of direct associations from indirect associations and the natural stop depending on the intrinsic properties of the analyzed data. Furthermore, the flexibility for introducing the priors generated higher accuracy for detecting the operon structure, with a low rate of false structure detection. Indeed, the algorithm with priors attained about 90% in the benchmark test and 70% in the genes on the genomic scale, and a small number of false genes (less than 1.0 gene per operon), for the known operons in E. coli. Thus, the present method will be useful to survey co-expressed genes for initially investigating gene regulatory systems from microarray data.

Acknowledgment

K.H. was partly supported by a Grant-in-Aid for Scientific Research on Priority Areas "Systems Genomics" (grant 20016028) and by a Grant-in-Aid for Scientific Research (A) (grant 19201039), from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] D. K. Slonim, "From patterns to pathways: gene expression data analysis comes of age," *Nat. Genet*. **32**, 502–508 (2002).

[2] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, et al., "Functional discovery via a compendium of expression profiles," *Cell* **102**, 109–126 (2000).

[3] S. A. McCarroll, C. T. Murphy, S. Zou, S. D. Pletcher, C. S. Chin, et al., "Comparing genomic expression patterns across species identifies shared transcriptional profile in aging," *Nat. Genet.* **36**, 197–204 (2004).

[4] C. J. Wolfe, I. S. Kohane and A. J. Butte, "Systematic survey reveals general applicability of ''guilt-by-association'' within gene coexpression networks," *BMC Bioinformatics* **6**, 227 (2005).

[5] M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA* **95**, 14863–14868 (1998).

[6] L. F. Wu, T. R. Hughes, A. P. Davierwala, M. D. Robinson, R. Stoughton, et al., "Large-scale prediction of *Saccharomyces cerevisiae* gene function using overlapping transcriptional clusters," *Nat. Genet.* **31**, 255–265 (2002).

[7] J. M. Stuart, E. Segal, D. Koller and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," *Science* **302**, 249–255 (2003).

[8] A. Jain, *Algorithms for clustering data*. Prentice-Hall, Englewood Cliffs, 1988.

[9] D. J. Allocco, I. S. Kohane and A. J. Butte, "Quantifying the relationship between co-expression, co-regulation, and gene function," *BMC Bioinformatics* **5**, 18 (2004).

[10] M. Moriyama, Y. Hoshida, M. Otsuka, S. Nishimura, N. Kato, et al., "Relevance network between chemosensitivity and transcriptome in human hepatoma cells," *Mol. Cancer. Ther*. **2**, 199–205 (2003).

[11] M. Bredel, C. Bredel, D. Juric, G. R. Harsh, H. Vogel, et al., "Functional network analysis reveals extended gliomagenesis pathway maps and three novel MYC-interacting genes in human gliomas," *Cancer Res*. **65**, 8679–8689 (2005).

[12] D. Sanoudou, J. N. Haslett, A. T. Kho, S. Guo, H. T. Gazda, et al., "Expression profiling reveals altered satellite cell numbers and

glycolytic enzyme transcription in nemaline myopathy muscle," *Proc. Natl. Acad. Sci. USA* **100**, 4666–4671 (2003).

[13] B. H. Voy, J. A. Scharff, A. D. Perkins, A. M. Saxton, B. Borate, E. J. Chesler, L. K. Branstetter and M. A. Langston, "Extracting Gene Networks for Low-Dose Radiation Using Graph Theoretical Algorithms," *PLoS Comput. Biol.* **2**, e89 (2006).

[14] E. H. Simpson, "The interpretation of interaction in contingency tables," *J. Roy. Statist. Soc. B*, **13**, 238-241 (1951).

[15] H. Toh and K. Horimoto, "Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling," *Bioinformatics* **18**, 287– 297 (2002).

[16] S. Aburatani, K. Goto, S. Saito, H. Toh and K. Horimoto, "ASIAN: a web server for inferring a regulatory network framework from gene expression profiles," *Nucleic Acids Res*. **33**, W659–W664 (2005).

[17] J. Whittaker, Graphical Models in Applied Multivariate Statistics. Wiley, New York, 1990.

[18] A. de la Fuente, N. Bing, I. Hoeschele and P. Mendes, "Discovery of meaningful associations in genomic data using partial correlation coefficients," *Bioinformatics* **20**, 3565-3574 (2004).

[19] J. Schafer and K. Strimmer, "An empirical Bayes approach to inferring large-scale gene association networks," *Bioinformatics* **21**, 754-764 (2005).

[20] P. Spirtes, C. Glymour and R. Scheines Causation, Prediction, and Search (Springer Lecture Notes in Statistics, 2nd edition, revised). MIT Press, Cambridge, 2001.

[21] J. Pearl, Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann Publishers, Palo Alto, 1988.

[22] H. Salgado, G. Moreno-Hagelsieb, T. F. Smith and J. Collado-Vides, "Operons in *Escherichia coli*: genome analyses and predictions," *Proc. Natl. Acad. Sci. USA* **97**, 6652-6657 (2000).

[23] C. Sabatti, L. Rohlin, M. K. Oh and J. C. Liao, "Co-expression patterm from DNA microarray experiments as a tool for operon prediction," *Nucleic Acids Res.* **30**, 2886-2893 (2002).

[24] T. W. Anderson, An Introduction to Multivariate Statistical Analysis. 2nd Edition, John Wiley & Sons, New York, 1984.

[25] I. M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I. T. Paulsen, M. Peralta-Gil and P. D. Karp, "EcoCyc: a comprehensive database resource for *Escherichia coli,*" *Nucleic Acids Res*. **33**, D334–D337 (2005).

[26] J. A. Bernstein, A. B. Khodursky, P. H. Lin, S. Lin-Chao and S. N. Cohen, "Global analysis of mRNA decay and abundance in *Escherichia coli* at single-gene resolution using two-color fluorescent DNA microarrays," *Proc. Natl. Acad. Sci. USA* **99**, 9697–9702 (2002).

[27] J. A. Bernstein, P. H. Lin, S. N. Cohen and S. Lin-Chao, "Global analysis of *Escherichia coli* RNA degradosome function using DNA microarrays," *Proc. Natl. Acad. Sci*. USA **101**, 2758–2763 (2004).

[28] J. Courcelle, A. Khodursky, B. Peter, P. O. Brown and P. C. Hanawalt, "Comparative gene expression profiles following UV exposure in wild-type and SOS-deficient *Escherichia coli*," *Genetics* **158**, 41–64 (2001).

[29] A. B. Khodursky, B. J. Peter, N. R. Cozzarelli, D. Botstein, P. O. Brown and C. Yanofsky, "DNA microarray analysis of gene expression in response to physiological and genetic changes that affect tryptophan metabolism in *Escherichia coli*," *Proc. Natl. Acad. Sci. USA* **97**, 12170–12175 (2000).

[30] A. B. Khodursky, B. J. Peter, M. B. Schmid, J. DeRisi, D. Botstein, P. O. Brown and N. R. Cozzarelli, "Analysis of topoisomerase function in bacterial replication fork movement: use of DNA microarrays," *Proc. Natl. Acad. Sci. USA* **97**, 9419–9424 (2000).

[31] K. Lee, J. A. Bernstein and S. N. Cohen, "RNase G complementation of rne null mutation identifies functional interrelationships with RNase E in *Escherichia coli*," *Mol. Microbiol*. **43**, 1445–1456 (2002).

[32] K. Lee, X. Zhan, J. Gao, J. Qiu, Y. Feng, R. Meganathan, S. N. Cohen and G. Georgiou, "RraA. a protein inhibitor of RNase E activity that globally modulates RNA abundance in *E. coli*," *Cell* **114**, 623–634 (2003).

[33] T. H. Tani, A. Khodursky, R. M. Blumenthal, P. O. Brown and R. G. Matthews, "Adaptation to famine: a family of stationary-phase genes revealed by microarray analysis," *Proc. Natl. Acad. Sci. USA* **99**, 13471–13476 (2002).