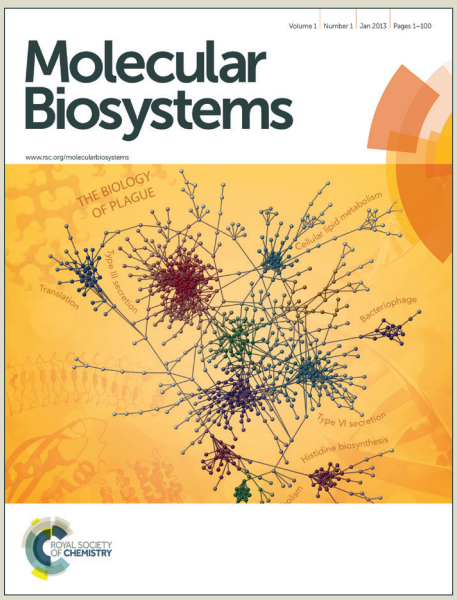
Molecular BioSystems

Accepted Manuscript



ROYAL SOCIETY OF CHEMISTRY

www.rsc.org/molecularbiosystems

Molecular BioSystems RSCPublishing

METHOD

Inferring cellular regulatory networks with Bayesian Model Averaging for Linear Regression (BMALR)

Cite this: DOI: 10.1039/x0xx00000x

Xun Huang[a] and Zhike Zi[a,b*]

[a]*BIOSS Centre for Biological Signalling Studies, University of Freiburg, 79104, Freiburg, Germany.*
[b]*Present address: Max Planck Institute for Molecular Genetics, 14195, Berlin, Germany*
[*]*Corresponding author, E-mail: zhike.zi@molgen.mpg.de*

Received ooth January 2014,
Accepted ooth January 2014

DOI: 10.1039/x0xx00000x

www.rsc.org/

Bayesian network and linear regression have been widely applied to reconstruct cellular regulatory networks. In this work, we propose a Bayesian Model Averaging for Linear Regression (BMALR) method to infer molecular interactions in biological systems. This method uses a new closed form solution to compute the posterior probabilities of the edges from regulators to the target gene within a hybrid framework of Bayesian model averaging and linear regression methods. We have assessed the performance of BMALR by benchmarking on both *in silico* DREAM datasets and real experimental datasets. The results show that BMALR achieves both high prediction accuracy and high computational efficiency across different benchmarks. A pre-processing of the datasets with the log transformation can further improve the performance of BMALR, leading to a new top overall performance. In addition, BMALR can achieve robust high performance in community predictions when it is combined with other competing methods. The proposed method BMALR is competitive compared to the existing network inference methods. Therefore, BMALR will be useful to infer regulatory interactions in biological networks. A free open source software tool for BMALR algorithm is available at https://sites.google.com/site/bmalr4netinfer/.

Introduction

With advances of high-throughput experimental technologies, plenty of network inference methods have been developed to identify regulatory interactions in cellular networks from quantitative experimental data. These network inference methods are becoming increasingly important in the field of systems biology to address many biological problems. Examples of network inference approaches include Bayesian networks[1-5], mutual information[6-8], linear regression[9-11], ordinary differential equations[12, 13] and statistical test[14, 15]. Among these approaches, Bayesian network has become popular due to the following reasons: (1) Bayesian network uses the probability theory, which is suitable for dealing with noise in biological data. (2) The prior knowledge of molecular interactions from literature or curated databases can be well encoded in the prior distribution structure of Bayesian network. In addition, linear regression based methods are also widely used in biological network inference due to their high computational efficiency.

A Bayesian network is a graphical model that describes probabilistic relationships between network variables. Such relationships are encoded within the structure of a directed acyclic graph. To infer the interactions of network variables, one strategy is to find a directed acyclic graph that most likely generates observed experimental data, which is assumed to be a steady data set for static Bayesian networks. This is performed by evaluating each possible graph with a score-based approach in the Bayesian context and subsequently search for the graph that maximizes the score[16]. The score function is defined with two common probabilistic models: linear Gaussian models and multinomial models[3]. However, it is a computationally laborious problem to evaluate all possible graphs that correspond to all possible interactions and choose the best scoring graph[17, 18]. To address this problem, heuristic search methods (e.g.: the greedy-hill climbing approach) were proposed[5]. On the other hand, given limited amounts of data, a variety of graph structures may describe the data similarly well. Therefore, network-averaging strategy was proposed to find the consensus interactions present in most of the high-scoring graphs[5, 19].

To improve the computational efficiency of network reconstruction, a decomposition technique is applied in regression-based methods. The inference of regulatory interactions targeting all N genes is decomposed into N independent sub-problems by inferring the interactions from regulators to a single target gene. Linear regression models have been proposed to solve these sub-problems. One popular method uses singular value decomposition to construct a set of candidate networks that match the observed data sets and regression approaches are employed afterwards to choose the most likely solution[20-22]. Another widely-used linear regression

algorithms is Lasso (Least Absolute Shrinkage and Selection Operator)[23], which uses L1-norm regularization to efficiently select a parsimonious set of regulators for each target gene. Different Lasso derived methods have been developed, which include LARS[24], GLASSO[10] and Inferelator[25, 26].

The combination of Bayesian network and linear regression could be a good strategy for the inference of cellular networks as it can take advantage of the strengths from both methods. For example, Rogers *et al.*[27] proposed a Bayesian regression method to reconstruct regulatory networks from gene expression data with a fast Sparse Bayesian regression algorithm of Tipping and Faul[27]. This method does not require the discretization of data and requires no setting of heuristic threshold for the predicted interactions. Moreover, Bayesian model averaging method integrated in a regression context has been applied to inferring regulatory network from time series data[28-30]. These studies have shown the advantages of Bayesian network methods in incorporating prior knowledge and the computational efficiency of regression approaches.

In this work, we developed a new method that uses Bayesian model averaging for linear regression (BMALR) to infer cellular regulatory networks. In this method, we applied a new closed form solution to calculate the posterior probabilities of the edges from putative regulators to the target gene, which leads to high computational efficiency and high prediction accuracy. We assessed this new method by benchmarking with *in silico* datasets from the DREAM (Dialogue on Reverse Engineering Assessment and Methods) project[31-34] and the datasets of real experiments. The results indicate that our method BMALR is competitive in terms of prediction accuracy and computational efficiency when it is compared to the best existing methods. The log transformation of datasets can further improve the performance of BMALR. In addition, we evaluated the performance of community predictions[34] with and without BMALR on DREAM benchmarks. The community prediction methods with BMALR can achieve robust high performance, which suggests that BMALR has a complementary advantage in community predictions.

Methods

The proposed method BMALR integrates Bayesian model averaging method with linear regression approach. In general, BMALR implements linear regression of the data of the target node on all combinations of other nodes. The final score of the edge from a parent node to the target node is the sum of the posterior probability of the linear regression models that contain this edge.

In the following sections, we introduce more technical details about Bayesian model averaging and a new close form solution for calculating the likelihood of each local structure in Gaussian Bayesian network. At the end, the datasets, performance metrics and network inference methods used for comparison are described.

Bayesian model averaging

To infer regulatory interactions in cellular networks, one way is to find the Bayesian network structure G that best explains the data. This is normally achieved by maximizing the likelihood of the observed dataset generated from the network structure G (maximum likelihood) or the posterior probability of the structure G given the observed data (maximum a posteriori). However, given a limited number of observed datasets, many Bayesian network models may explain the data almost equally well. It would be risky to make an inference on the interactions of network variables depending on a single optimized Bayesian network structure[5, 19].

Instead of searching for the best Bayesian network structure, Bayesian model averaging was proposed to find the edge features (f) that are present in most high-scoring Bayesian network structures[5, 19]. An edge feature (f) is the edge relation feature between network variables X_i and X_j in a Bayesian network structure (G). Such a feature can be quantified with the posterior probability of f:

$$P(f \mid D) = \sum_G f(G) P(G \mid D) \tag{1}$$

This probability shows our confidence in edge feature f given the observed dataset (D). If the Bayesian network structure G contains f, $f(G)$ equals to 1, otherwise 0.

As the number of candidate network structures increase exponentially with the number of node variables, it is not feasible to exactly compute Equation 1 from all possible candidate networks.

Local model averaging for linear regression

In order to have an efficient and accurate estimate for posterior probability of the edge feature, the global network is decomposed into a set of local networks[27, 28]. Each local network is composed of one target variable and its possible parents (regulators). The local network can be reconstructed by model averaging with the local structures composed of all the possible parent sets of a target variable[19], which is similar to the local learning approach based on the Markov blanket[35]. Here we model the local structures with a weighted linear combination of the values of its parents:

$$X_i = \sum_{j \neq i} w_{ji} X_j + \varepsilon_i \tag{2}$$

where X_i is the value of variable i. w_{ji} is a weight constant representing the influence of variable j to variable i. If w_{ji} is zero, there is no edge from j to i in the regulatory network. If w_{ji} is non-zero, j is one of i's regulators (parents). ε_i denotes the noise.

The posterior probability of each edge is the sum of posterior probabilities of all the putative local structures containing the edge. This leads to the following approximation (in comparison with Equation 1) for posterior probability of an edge feature f:

$$P(f \mid D) \approx \sum_{Pa \in S_i} f(G_{Pa}) P(G_{Pa} \mid D_{Pa, X_i}) \tag{3}$$

where the node X_i is the target of the edge feature f. S_i is the set of all possible parent sets of X_i. G_{Pa} is a local structure that is composed of the edges from the nodes in Pa, a parent set of node X_i. $D_{Pa,Xi}$ denotes the data restricted to X_i and the node variables in Pa. If the local structure G_{Pa} contains f, $f(G_{Pa})$ equals to 1, otherwise 0.

Based on Bayes theorem, Equation 3 can be written as:

$$\begin{aligned} P(f \mid D) &\approx \sum_{Pa \in S_i} f(G_{Pa}) P(G_{Pa} \mid D_{Pa, X_i}) \\ &= \frac{\sum_{Pa \in S_i} f(G_{Pa}) P(D_{Pa, X_i} \mid G_{Pa}) P(G_{Pa})}{\sum_{Pa \in S_i} P(D_{Pa, X_i} \mid G_{Pa}) P(G_{Pa})} \end{aligned} \tag{4}$$

where $P(G_{Pa})$ is the prior probability of structure G_{Pa}. For simplicity, we can assume that cardinalities of parent sets are uniformly distributed, meaning $P(G_{Pa}) \equiv 1$[19]. Here, we set a restriction on the maximum cardinalities of parent sets (denoted as *maxFanIn*) in order to reduce the computation cost. This strategy has been applied before in other Bayesian learning approaches[3, 36]. In this paper, we set the default value of *maxFanIn* as 2. It is worth noting that the inference results are not sensitive to the value of *maxFanIn*[29], which is shown in the part of results and discussion.

Probability in Gaussian Bayesian network

To calculate $P(D_{Pa,Xi}|G_{Pa})$, the likelihood that the local structure G_{Pa} generates the data $D_{Pa,Xi}$, we assume that the noise term in the linear regression models (ε_i in Equation 2) follows multivariate normal distribution. As a result, variable i and its regulators Pa will be in multivariate normal distribution, which is corresponding to a Gaussian Bayesian network[37, 38]. In Gaussian Bayesian network, we can compute the likelihood that the local structure G_{Pa} generates the data $D_{Pa,Xi}$ as[37, 38]:

$$P(D_{Pa,X_i} | G_{Pa}) = \frac{\rho(D_{Pa,Xi})}{\rho(D_{Pa})} \quad (5)$$

Assuming a normal-Wishart prior in the Gaussian Bayesian network, the probability density of data, $\rho(D_{Pa,Xi})$ and $\rho(D_{Pa})$ can be approximated with the following formula.

$$\rho(D_W) \propto \left| E^{(W)} + M \cdot R^{(W)} \right|^{\frac{-l_W - M}{2}} \quad (6)$$

where W is a set of variables (e.g.: for the calculation of $\rho(D_{Pa})$, $W = Pa$). l_W is cardinalities of W. $E^{(W)}$ is a l_W-by-l_W identity matrix. $R^{(W)}$ is the l_W-by-l_W Pearson correlation matrix of D_W. M is the sample size.
The derivation of above equations and other technical details for calculating probability density of data are provided in the Supplementary Text.

Datasets

We evaluated the proposed method with a variety of datasets: (1) The *in silico* benchmark data from DREAM4 "*in silico* network challenge" and DREAM 5 "network inference challenge", which are available at http://wiki.c2b2.columbia.edu/dream/index.php/Challenges. (2) The real experimental data obtained in single cell flow cytometry experiments[4], which measured the expression level of 11 signaling molecules in T-cell signaling network upon various interventions.

Performance metrics

To assess the performance of BMALR and previous network inference methods, we used a number of performance metrics, which include the area under the precision-recall (AUPR) and receiver operating characteristic (AUROC) curves, the *F*-score[39], empirical *p*-values of AUPR and AUROC for the DREAM project benchmarks[34], as well as the *p*-value of the one-sided Fisher's exact test[40].
To compute AUPR and AUROC metrics, we counted true positive (*TP*), false positive (*FP*), true negative (*TN*), and false negative (*FN*) by comparing the inferred network with gold standard networks at a certain threshold setting. Accordingly, true positive rate (*TPR*, or recall), positive predictive value (*PPV*, or precision), false positive rate (*FPR*) are calculated. Then the values of AUPR and AUROC were calculated by creating the precision-recall curve (PR curve) and receiver operating characteristic curve (ROC curve) at various threshold settings.
The *F*-score is defined as:

$$F = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} \quad (7)$$

The *p*-value of the one-sided Fisher's exact test is defined as:

$$p = 1 - \sum_{i=0}^{TP-1} \frac{\binom{Np}{i}\binom{Nt - Np}{N_{TP} + N_{FP} - i}}{\binom{Nt}{Np}} \quad (8)$$

where Np is the number of edges in gold standard network. Nt is the maximum number of possible edges in the network. $Nt = n(n\text{-}1)$ for a directed network with n nodes (excluding self-loop edges). N_{TP} and N_{FP} are the number of true positive and false positive edges in the inferred network, respectively. This *p*-value represents the probability to obtain no less than N_{TP} true positive edges by randomly selecting Np edges. The *p*-value of the one-sided Fisher's exact test is useful for the evaluation of inference methods on small networks.

Log transformation of data

To test whether the preprocessing of datasets can improve the performance of BMALR, we applied log transformation to reduce the positive skew of the data so that the distributions of transformed datasets are similar to normal distributions. We first check the skewness of the variable X. If the distribution of variable X is positively skewed, log transformation is applied as the following[41]: a constant C is added to X, then an optimization procedure is applied to obtain an optimal C that makes the nonparametric skew of $\log(X+C)$ as close to zero as possible.

Network inference methods used for comparison

We compared the proposed BMALR method with 5 popular network inference methods: (1) GENIE3[42] based on feature selection with decision tree. GENIE3 was the winner of both DREAM4 and DREAM5 network inference challenges. (2) CLR[7] based on mutual information with background correction. (3) ARACNE[43, 44] based on mutual information with data processing inequality. (4) The PC-algorithm in PCALG package. We implemented pcSelect in PCALG package[45, 46], a simplified PC-algorithm with variable selection. (5) The regression based method, LARS (Least Angle Regression)[24]. (6) Bayesian model averaging method using Markov Chain Monte Carlo sampling (denoted as MCMC)[19, 47]. We ran these tools with their default settings.

Results and discussion

Performance on DREAM4 *in silico* size 100 multifactorial sub-challenge

The aim of DREAM4 *in silico* size 100 multifactorial sub-challenge is to infer the structures of five gold standard networks from given *in silico* gene expression datasets that are

simulated by 100 multifactorial perturbations of all the genes simultaneously[32, 33]. The five gold standard networks have 100 components (genes) and they have 176, 249, 195, 211 and 193 gold standard links, respectively. The performance metrics of our method (BMALR) and other applied inference methods for this sub-challenge is given in Table 1. The results show that BMALR obtained the highest overall score among all the applied network inference methods. More specifically, BMALR achieved the highest AUPR and AUROC scores in four networks and the second-highest scores in one network. The individual PR and ROC curves of the inference methods on each network are available in Supplementary Figure 1-2. The overall performance of BMALR is slightly better than GENIE3, which was the winner of this sub-challenge.

Gaussian Bayesian models assume the datasets of variables have normal distributions. However, if the datasets of the variables have highly skewed distributions, the assumption of normal distribution of the data would be invalid. To deal with the skewed data, we applied the log transformation for the datasets[41] and tested whether such a strategy could improve the performance of BMALR. As it is shown in Table 1, a preprocessing of the datasets with the log transformation improves the performance of BMALR (16% increase of overall score in this benchmark).

Table 1 Performance metrics of inference methods on DREAM4 *in silico* size 100 multifactorial sub-challenge. The bold numbers indicate the best value in each performance metric. BMALR* denotes the results of BMALR with the log transformation of the datasets.

Method	NET1		NET2		NET3		NET4		NET5		Score	Time (second)
	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC		
BMALR	0.155	**0.745**	**0.166**	**0.737**	**0.231**	**0.792**	**0.234**	**0.808**	**0.214**	0.778	**39.4**	1.8E+00
GENIE3	0.154	**0.745**	0.155	0.733	**0.231**	0.775	0.208	0.791	0.197	**0.798**	37.4	3.5E+01
CLR	**0.157**	0.733	0.137	0.693	0.199	0.745	0.185	0.738	0.192	0.737	31.6	**2.1E-01**
ARACNE	0.145	0.690	0.129	0.688	0.186	0.742	0.167	0.721	0.152	0.750	28.6	1.7E+00
PCALG	0.134	0.714	0.109	0.663	0.200	0.712	0.166	0.702	0.184	0.725	27.1	3.2E+00
LARS	0.136	0.619	0.123	0.619	0.205	0.673	0.206	0.656	0.209	0.644	25.7	2.0E+01
BMALR*	0.213	0.772	0.188	0.746	0.274	0.799	0.257	0.812	0.242	0.810	45.7	1.8E+00

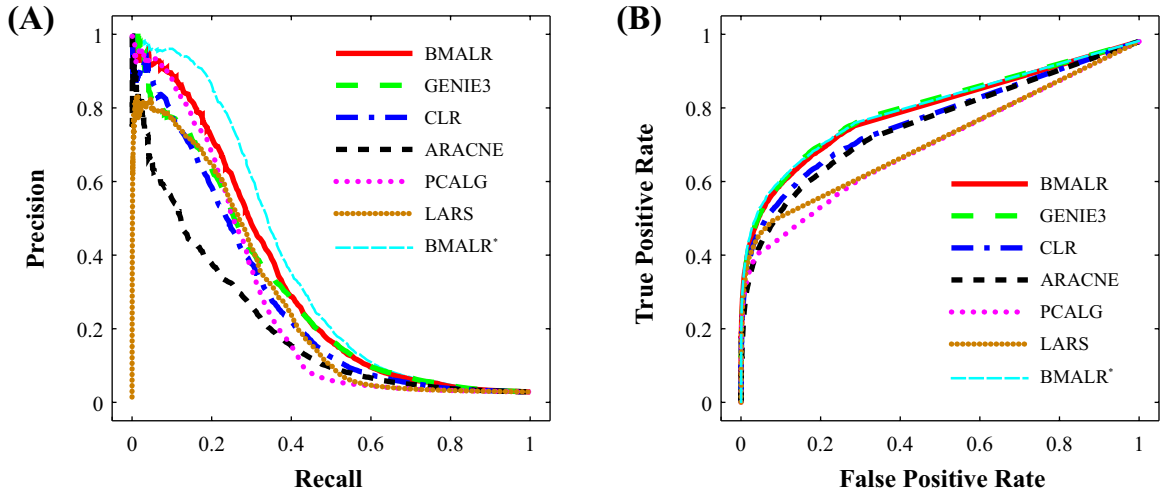


Fig. 1 PR and ROC curves of different methods for DREAM5 *in silico* sub-challenge. BMALR* denotes the results of BMALR with the log transformation of the datasets. (A) Precision-recall (PR) and (B) Receiver operating characteristic (ROC) curves.

Performance on DREAM5 network inference sub-challenge

To evaluate the performance of BMALR method on predicting the interactions of large networks, we compared BMALR with other methods by applying them to the DREAM5 network inference sub-challenge with the dataset simulated from an *in silico* network, which has 1643 genes, 195 transcription factors (TFs) and 805 chips. The top 100,000 edge predictions are used for the DREAM5 evaluation[34]. *In silico* DREAM5 network model assumes that the mRNAs are directly translated into proteins without any further regulations. The simulated data for TFs' protein abundances are highly correlated with corresponding mRNA abundances data. Therefore, we use TFs' mRNA abundances as proxies for their protein abundances.

Table 2 shows the performance metrics of BMALR and other inference methods on the DREAM5 network inference sub-challenge. The results indicate BMALR obtained the best performance with the highest AUPR score and the second highest AUROC score, as well as best optimal *F*-score. The log transformation of datasets can further increase the performance of BMALR, leading to a best overall performance.

According to the individual PR and ROC curves of the inference methods in this benchmark (Figure 1), BMALR tends to have higher precision than other methods at low recalls in the PR curve, while the ROC curve profile of BMALR is almost the same as GENIE3. In addition, it is worth noting that the computational cost of BMALR in this sub-challenge is much less than other competitive methods such as GENIE3, PCALG and ARACNE (Table 2).

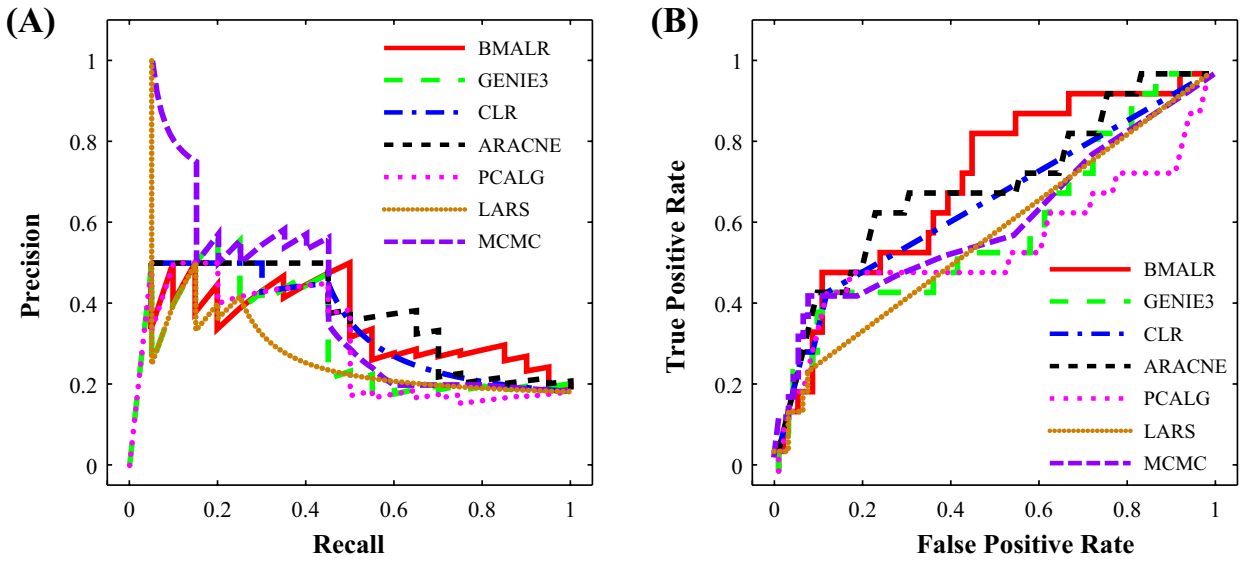


Fig. 2 PR curve and ROC curve for T-cell signaling network. (A) PR curves (B) ROC curves

Table 2 The performance metrics of inference methods on DREAM5 network inference sub-challenge with *in silico* dataset. The bold numbers indicates the best values among all methods. BMALR* denotes the results of BMALR with the log transformation of the datasets.

Method	AUPR	AUROC	Optimal *F*-score	Time (second)
BMALR	**0.320**	0.809	**0.372**	**1.3E+02**
GENIE3	0.291	**0.815**	0.346	1.2E+04
CLR	0.266	0.786	0.332	4.1E+02
ARACNE	0.192	0.771	0.280	2.5E+04
PCALG	0.265	0.714	0.340	7.0E+03
LARS	0.263	0.726	0.353	5.0E+02
BMALR*	0.362	0.815	0.403	1.3E+02

Table 3 The performance metrics of inference methods on the benchmark of T-cell signaling network. The bold numbers indicates the best values among all methods.

Method	AUPR	AUROC	Optimal F-score	*p*-value of Fisher's exact test	Time (second)
BMALR	0.343	**0.720**	**0.500**	2.70E-04	**1.3E-01**
GENIE3	0.304	0.607	0.462	1.14E-03	3.7E+02
CLR	0.344	0.666	0.450	1.86E-03	2.3E-01
ARACNE	0.372	0.709	0.481	5.39E-04	2.7E+02
PCALG	0.300	0.558	0.450	1.86E-03	1.9E+00
LARS	0.275	0.585	0.313	4.09E-02	8.6E-01
MCMC	**0.391**	0.632	**0.500**	**1.97E-04**	3.8E+02

Performance on the benchmark of T-cell signaling network with real experimental data

In this section, we evaluated the performance of BMALR and other methods on the benchmark of T-cell signaling network with 11 components and 20 interactions. The experimental data consists of 11 phosphorylated T-cell signaling molecules under 9 perturbation conditions, which were simultaneously measured with single cell flow cytometry[4]. We first applied z-score normalization for each dataset, so that all the proteins have a mean of 0 and a standard deviation of 1.

The results shown in Table 3 suggest that BMALR is among the top performing approaches, with the highest AUROC score. The optimal *F*-score of BMLAR is the same as that obtained with Bayesian model averaging method using Markov Chain Monte Carlo sampling (MCMC)[19, 47]. The individual PR and ROC curves of each network inference method on this benchmark are shown in Figure 2.

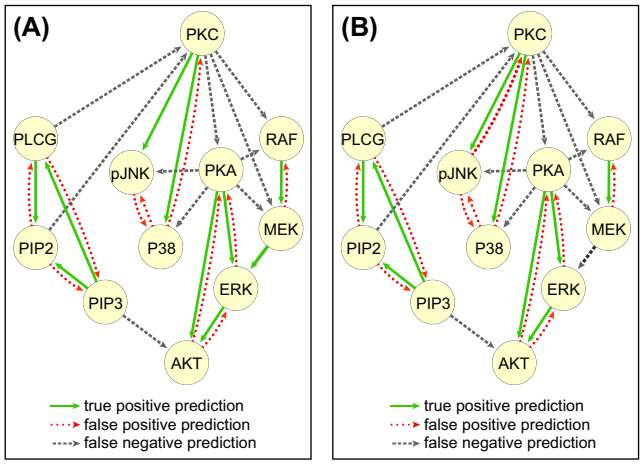


Fig. 3 The predicted T-cell signaling networks with top 20 edges by (A) BMALR and (B) MCMC network inference method, respectively.

Figure 3 shows the networks predicted by the top 2 performing methods (BMALR and MCMC), which have the same number of edges as the gold standard network derived by Sachs *et al.*[4]. The two predicted networks are very similar. They have only 2 different edges: MCMC predicted network has a false edge pJNK→PKC that is not shown in the predicted network by

BMALR and a true positive edge MEK→ERK is missed in MCMC predicted network, while it is present in the predicted network by BMALR. Interestingly, most of the false positive edges in the two predicted network have correct pairs of molecular interactions, but the directions of these false positive edges are wrong, which indicates that both BMALR and MCMC methods are difficult to determine the directions of molecule interactions in this benchmark.

Similarity of predictions from different inference methods

Recent work has shown that different network inference methods provide different predicted interactions on the same regulatory system. To address this problem, it was proposed to make community predictions by integrating the results from multiple inference methods because limitations of different methods tend to be cancel out[33, 34]. Therefore, it is important to investigate the similarity and difference of the predictions from different network inference methods[15, 34].

In order to compare the overlap and the difference of predicted interactions between our method and others, we compared the binary networks from each method at a certain cutoff, which is set to ensure that the binary networks generated from each method have the same number of edges (4012 edges for the DREAM5 sub-challenge) with the gold standard network. To quantify the similarity between the predictions of different methods, we performed single linkage cluster analysis[48] with a distance metric based on spearman correlations of the ranks of predicted edges from each method. As shown in Supplementary Figure 3, the interactions predicted by our method BMALR are most similar to those predicted by LARS and PCALG. This could be explained by the fact that linear regression is used in BMALR, PCALG and LARS.

To test whether BMALR could help to achieve higher performance by making community predictions with other individual network inference methods, we performed community predictions with combinations of every two individual methods for gold standard network 1 (NET1) of DREAM4 *in silico* size 100 multifactorial sub-challenge following the approach proposed by Marbach *et al.*[34]. As shown in Figure 4, community methods with BMALR can achieve robust high performance in the predicted networks. Similar high performance of community methods with BMALR is obtained with the application for other gold standard networks of DREAM4 sub-challenge (Supplementary Figure 4). In all the cases, the highest performance predictions were obtained in the community predictions with the combination of BMALR plus other competition methods such as CLR, GENIE3 and ARACNE. These results suggest that BMALR is complementary to other inference methods and could be a good candidate method for community predictions.

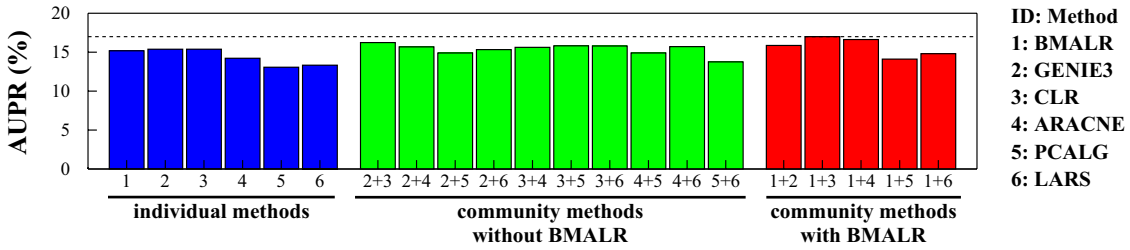


Fig. 4 The performance (area under precision-recall curve, AUPR) of individual methods and community methods with combinations of every two individual methods for gold standard network 1 (NET1) of DREAM4 *in silico* size 100 multifactorial sub-challenge. The dashed horizontal line denotes the highest performance level in all the methods.

Fan-in error analysis

We next checked whether BMALR is resistant to the fan-in error stemming from the difficulties in predicting combinatorial regulations[33]. We performed indegree based network analysis on DREAM5 sub-challenge to study the robustness of BMALR and other methods to the fan-in error. Specifically, we investigated the overall performance of each inference method on predicting the regulatory input(s) of genes with one transcription factor (indegree = 1), two transcriptional factors (indegree = 2), etc. As it is described in[33], the prediction confidence for different regulatory inputs of genes is quantified with their ranks in the corresponding list of edge predictions (the first edge has a scaled prediction confidence of 100% and the last edge has confidence 0%).

The data shown in Figure 5 indicates that most inference methods have high prediction confidence for the edge of genes with 1 transcription factor input. In general, the prediction confidence is reduced as the indegree of genes increases, but the prediction confidence of BMALR and GNEIE3 seems more robust on high indegrees.

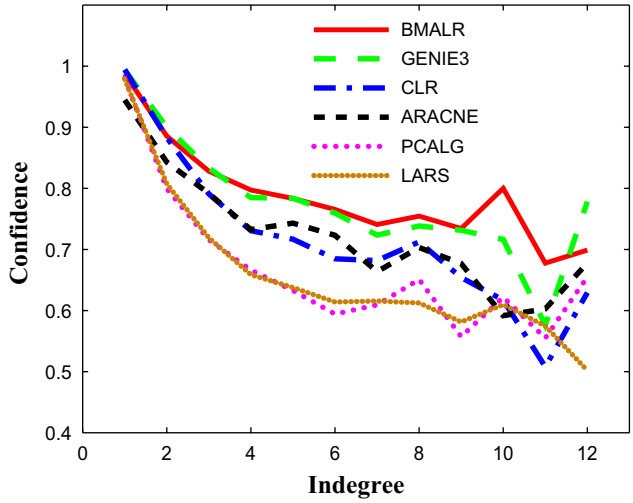


Fig. 5 Fan-in error analysis of the inference methods on DREAM5 network inference sub-challenge with *in silico* datasets.

Table 4 Performance metrics of Bayesian network based methods on DREAM4 *in silico* size 100 multifactorial sub-challenge. The bold numbers indicate the best value in each performance metric.

Method	NET1		NET2		NET3		NET4		NET5		Score	Time (second)
	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC		
BMALR	**0.155**	**0.745**	**0.166**	**0.737**	**0.231**	**0.792**	**0.234**	**0.808**	**0.214**	**0.778**	**39.4**	**1.8E+00**
PCALG	0.134	0.714	0.108	0.665	0.209	0.714	0.173	0.705	0.181	0.722	27.5	3.2E+00
BNfinder	0.125	0.584	0.0799	0.601	0.191	0.663	0.171	0.652	0.123	0.602	19.3	1.4E+03
SBM	0.111	0.635	0.100	0.600	0.199	0.668	0.182	0.680	0.172	0.663	22.8	4.0E+02

Comparing BMALR with other Bayesian network based methods

BMLAR uses the framework of Bayesian model averaging that integrates linear regression approach. Therefore, we investigated whether the combination of Bayesian model averaging with linear regression method could improve prediction performance compared with other Bayesian network methods. We applied previous reported Bayesian network methods such as the PC-algorithm method (PCALG)[45, 46], BNFinder[2] and the SBM method with sparse Bayesian model[27, 49] to the DREAM4 *in silico* size 100 multifactorial sub-challenge. The results shown in Table 4 indicate that BMLAR achieves the highest prediction accuracy over the other comparable Bayesian network methods. In addition, the computational efficiency of BMLAR is among the top method, with similar magnitude of used computational time as the PC-algorithm method with pcSelect (feature selection version of PCALG). It's worth noting that both BMALR and pcSelect method combine Bayesian network with linear regression method. The difference is that BMALR uses model averaging to calculate the posterior probability of each edge, while pcSelect uses partial correlation based independence test to score each edge.

Similar as the method proposed by Geiger and Heckerman (referred as Geiger-Heckerman in the following)[37], BMALR calculates the probability of local linear regression models using Gaussian directed acyclic graphical (DAG) models. The major difference between BMALR and Geiger-Heckerman is that BMALR integrates the idea of model averaging to compute the edge posterior probabilities, which has been applied to infer regulatory network from time series data by Hill *et al.*[29]. However, Hill *et al.* computed the probability of local linear regression models using Bayesian linear models with interaction terms, which is different from the Gaussian DAG models used in BMALR. Therefore, BMALR is a hybrid method for network inference, which is based on the method frameworks developed by Hill *et al.* and Geiger-Heckerman.

The influence of maximum number of parents (*maxFanIn*) on the performance of BMALR

To analyze whether the performance of BMALR depends on the setting of maximum number of parents (*maxFanIn*), we applied BMALR to the benchmarks of DREAM4 and DREAM5 *in silico* sub-challenges, and the T-cell signaling network with different settings of maximum number of parents (*maxFanIn* was set between 2-7). The results shown in Supplementary Tables 1-3 suggest that the performance metrics of BMALR are robust to the *maxFanIn* parameter (when *maxFanIn* $\geq$ 2). Biological networks are usually sparsely connected, in which most network nodes have a few upstream regulatory links[50]. Therefore, the probability of local structures with large numbers of parents is relatively small, which might explain the observed insensitivity of BMALR method to the *maxFanIn* parameter.

Conclusions

In this work, we propose a Bayesian Model Averaging for Linear Regression (BMALR) method to reconstruct cellular regulatory networks. This method used a new closed form solution to compute the posterior probabilities of the edges from putative regulators to the target gene with the integration of Bayesian model averaging and linear regression methods. We compared the performance of BMALR with other network inference methods by applying to a variety of benchmarks including both *in silico* datasets in DREAM projects and real experimental datasets. BMALR shows high performance across these benchmarks with different performance metrics. We have also shown that the log transformation of the datasets can further improve the performance of BMALR in different benchmarks. In addition, BMALR is competitive in terms of computational efficiency, especially for large-scale network inference. Our results indicate that BMALR is complementary to other inference methods as it can achieve robust high performance when it is used in combination with other methods for community predictions. Last but not least, BMALR seems to be resistant to the fan-in error stemming from the difficulties in predicting combinatorial regulations. Therefore, BMALR is expected to be useful to infer regulatory interactions in biological networks.

Acknowledgements

This work was supported by the grants to Z.Z. from BMBF GerontoSys II - NephAge (031 5896A) and the Excellence Initiative of the German Federal and State Government (EXC 294). We would like to thank Schuyler Lee and Xuedong Liu for proofreading of this manuscript. In addition, we would like to thank the anonymous reviewers for their valuable comments and suggestions, which helped us to improve this work.

References

1. D. Pe'er, A. Regev, G. Elidan and N. Friedman, *Bioinformatics*, 2001, 17 Suppl 1, S215-224.
2. B. Wilczynski and N. Dojer, *Bioinformatics*, 2009, 25, 286-287.
3. N. Friedman, M. Linial, I. Nachman and D. Pe'er, *J Comput Biol*, 2000, 7, 601-620.

4. K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger and G. P. Nolan, *Science*, 2005, 308, 523-529.
5. D. Pe'er, *Sci STKE*, 2005, 2005, pl4.
6. A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera and A. Califano, *BMC bioinformatics*, 2006, 7, S7-S7.
7. J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins and T. S. Gardner, *PLoS Biol*, 2007, 5, e8.
8. P. E. Meyer, F. Lafitte and G. Bontempi, *BMC bioinformatics*, 2008, 9, 461-461.
9. R. Opgen-Rhein and K. Strimmer, *BMC Systems Biology*, 2007, 1, 37-37.
10. J. Friedman, T. Hastie and R. Tibshirani, *Biostatistics*, 2008, 9, 432-441.
11. A. C. Haury, F. Mordelet, P. Vera-Licona and J. P. Vert, *BMC Syst Biol*, 2012, 6, 145.
12. M. Bansal, G. Della Gatta and D. di Bernardo, *Bioinformatics*, 2006, 22, 815-822.
13. D. Lai, X. Yang, G. Wu, Y. Liu and C. Nardini, *Bioinformatics*, 2011, 27, 232-237.
14. R. Kuffner, T. Petri, P. Tavakkolkhah, L. Windhager and R. Zimmer, *Bioinformatics*, 2012, 28, 1376-1382.
15. J. Qi and T. Michoel, *Bioinformatics*, 2012, 28, 2325-2332.
16. M. Bansal, V. Belcastro, A. Ambesi-Impiombato and D. di Bernardo, *Mol Syst Biol*, 2007, 3, 78.
17. D. M. Chickering, D. Geiger and D. Heckerman, *Learning Bayesian networks is NP-hard*, Citeseer, 1994.
18. D. M. Chickering, D. Heckerman and C. Meek, *J Mach Learn Res*, 2004, 5, 1287-1330.
19. N. Friedman and D. Koller, *Mach Learn*, 2003, 50, 95-125.
20. M. K. S. Yeung, J. Tegnér and J. J. Collins, *Proceedings of the National Academy of Sciences*, 2002, 99, 6163-6168.
21. Y. Wang, T. Joshi, X. S. Zhang, D. Xu and L. Chen, *Bioinformatics*, 2006, 22, 2413-2420.
22. N. S. Holter, A. Maritan, M. Cieplak, N. V. Fedoroff and J. R. Banavar, *Proc Natl Acad Sci U S A*, 2001, 98, 1693-1698.
23. R. Tibshirani, *J Roy Stat Soc B Met*, 1996, 58, 267-288.
24. B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, *Ann Stat*, 2004, 32, 407-451.
25. A. Madar, A. Greenfield, E. Vanden-Eijnden and R. Bonneau, *Plos One*, 2010, 5.
26. A. Greenfield, A. Madar, H. Ostrer and R. Bonneau, *Plos One*, 2010, 5.
27. S. Rogers and M. Girolami, *Bioinformatics*, 2005, 21, 3131-3137.
28. K. Y. Yeung, K. M. Dombek, K. Lo, J. E. Mittler, J. Zhu, E. E. Schadt, R. E. Bumgarner and A. E. Raftery, *Proc Natl Acad Sci U S A*, 2011, 108, 19436-19441.
29. S. M. Hill, Y. Lu, J. Molina, L. M. Heiser, P. T. Spellman, T. P. Speed, J. W. Gray, G. B. Mills and S. Mukherjee, *Bioinformatics*, 2012, 28, 2804-2810.
30. K. Lo, A. E. Raftery, K. M. Dombek, J. Zhu, E. E. Schadt, R. E. Bumgarner and K. Y. Yeung, *BMC systems biology*, 2012, 6, 101.
31. G. Stolovitzky, D. Monroe and A. Califano, *Ann N Y Acad Sci*, 2007, 1115, 1-22.
32. D. Marbach, T. Schaffter, C. Mattiussi and D. Floreano, *Journal of computational biology : a journal of computational molecular cell biology*, 2009, 16, 229-239.
33. D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano and G. Stolovitzky, *Proc Natl Acad Sci U S A*, 2010, 107, 6286-6291.
34. D. Marbach, J. C. Costello, R. Kuffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, M. Kellis, J. J. Collins and G. Stolovitzky, *Nat Methods*, 2012, 9, 796-804.
35. C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani and X. D. Koutsoukos, *J Mach Learn Res*, 2010, 11, 171-234.
36. S. Y. Kim, S. Imoto and S. Miyano, *Brief Bioinform*, 2003, 4, 228-235.
37. D. Geiger and D. Heckerman, *Ann Stat*, 2002, 30, 1412-1440.
38. R. E. Neapolitan, *Learning bayesian networks*, Pearson Prentice Hall Upper Saddle River, 2004.
39. G. Altay, F. Emmert-Streib and others, *Biology direct*, 2011, 6, 1-16.
40. R. A. Fisher, *Journal of the Royal Statistical Society*, 1935, 98, 39-82.
41. C. Feng, H. Wang, N. Lu and X. M. Tu, *Stat Med*, 2013, 32, 230-239.
42. V. A. Huynh-Thu, A. Irrthum, L. Wehenkel and P. Geurts, *Plos One*, 2010, 5.
43. A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera and A. Califano, *BMC bioinformatics*, 2006, 7 Suppl 1, S7.
44. A. A. Margolin, K. Wang, W. K. Lim, M. Kustagi, I. Nemenman and A. Califano, *Nat Protoc*, 2006, 1, 662-671.
45. M. Kalisch and P. Buhlmann, *J Mach Learn Res*, 2007, 8, 613-636.
46. P. Buhlmann, M. Kalisch and M. H. Maathuis, *Biometrika*, 2010, 97, 261-278.
47. A. V. Werhli, M. Grzegorczyk and D. Husmeier, *Bioinformatics*, 2006, 22, 2523-2531.
48. R. Sibson, *The Computer Journal*, 1973, 16, 30-34.
49. M. E. Tipping and A. C. Faul, 2003.
50. R. D. Leclerc, *Mol Syst Biol*, 2008, 4, 213.

