The First International Symposium on Optimization and Systems Biology (OSB'07)
Beijing, China, August 8–10, 2007
Copyright © 2007 ORSC & APORC, pp. 178–188

Missing Enzyme Identification Using Reversible-Jump Markov-Chain-Monte-Carlo Learning Approach

Bo Geng[1,2] Xiaobo Zhou[1,*] Jinmin Zhu[1]
Y. S. Hung[2] Stephen Wong[1]

[1] HCNR-CBI, Harvard Medical School and Brigham & Women's Hospital, Boston, MA 02215
[2] Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong

Abstract Computational identification of missing enzymes plays a significant role in reconstruction of metabolic network. For a metabolic reaction, given a set of candidate enzymes identified by certain biological evidences, there is a need to develop a powerful mathematical model to predict the actual enzyme(s) catalyzing the reaction. In this study, a regression model is proposed to solve the problem, in which a reversible jump Markov-chain-Monte-Carlo learning technique is used to estimate the model parameters. We evaluate the model using known reactions in *Escherichia coli*, *Mycobacterium tuberculosis*, *Vibrio cholerae*, and *Caulobacter cresentus*. It is demonstrated that the model obtains favorable results compared with several other approaches.

Keywords Metabolic network; missing enzymes identification; regression model; Markov chain Monte Carlo.

1 Introduction

The study of genomic sequencing and high-throughput biotechnologies is making metabolic network reconstruction possible [1]. Such reconstruction enables systematic comprehending of molecular mechanism of cellular metabolism [2]. Biochemical experiments in past decades have revealed metabolic functions in various organisms, which contribute considerably to the reconstruction of metabolic networks. Meantime, pathway inference methods have been developed to complement metabolic network reconstruction [3, 4]. However, there are still many metabolic reactions with enzymes unknown even in well-studied organism [5, 6], which results in partially reconstructed network. In addition to conventional biological experiments, recent study in systems biology attempt to invent computational methods to address the problem.

A number of computational strategies have been proposed. PathwayTools hole-filler [7] exploited sequence homology and pathway-based evidences to identify a set

[*] Corresponding to: zhou@crystal.harvard.edu

of candidates and subsequently predict actual enzyme(s) from the candidates using Bayesian model. Chen *et al.* [8] combined local structure of metabolic networks and phylogenetic profiles to identify missing genes. Kharchenko *et al.* [9] incorporated co-expressing properties of metabolic network, which is complementary to the sequence homology method. Recently, Kharchenko [10] proposed a novel approach based on local structure of metabolic network, gene expression, protein fusion events and other evidences. The common theme of the strategies is to determine the actual enzymes from a group of candidates. Therefore, there is a need to develop a powerful mathematical model to predict from candidates the actual enzymes catalyzing the metabolic reactions of interest.

Here we proposed a model consisting of a mixture of k radial basis functions (RBFs) and a linear regression term to predict actual enzymes, in which a reversible jump MCMC technique [11] is adopted to estimate model order and the parameters. Owing to its capability of model order selection, the reversible jump MCMC exhibits satisfactory predictive performance in our experimental results. Moreover, we compare its predictive power with Bayesian network model, which is previously used in [7] for missing enzyme identification, support vector regression (SVR) [12], perceptron, and back-propagation (BP) neural network [13].

2 Datasets

A metabolic reaction is known as chemical changes in living cells by which energy is provided for vital processes, involving substrates, products, and enzyme. While there are reactions with enzymes 'missing', some others have enzymes assigned to them based on literature, genomic sequence, or databases. These are referred to as 'known' reactions. We adopt 'known' reactions rather than 'missing' reactions to evaluate the proposed model. Two datasets are used in our study, that is, 'known' reactions in *Escherichia coli* (*E.coli*) and three other bacteria: *Mycobacterium tuberculosis* (*Mtu*), *Vibrio cholerae* (*Vch*), and *Caulobacter cresentus* (*Ccr*).

Escherichia coli. For the sake of explanation, we arbitrarily choose 100 known reactions (can use more) in *E.coli*. The reactions and enzyme information in *E.coli* are available on the website of System Biology Research Group at UCSD, EcoCyc [14], and Kyoto Encyclopedia of Genes and Genomes (KEGG) database [15]. We use 100 reactions that can be found at all the three information resources.

Three other bacteria. We utilize mixed 60 known reactions (20 for each organism) taking placing in *Mycobacterium tuberculosis* (Mtu), *Vibrio cholerae* (Vch), and *Caulobacter cresentus* (Ccr). The reaction information can be found at both KEGG and MetaCyc database [16].

For each reaction in our datasets, first, a group of candidate proteins are identified. Second, a feature vector is calculated for each candidate. Third, the proposed model is used to predict whether the candidates are actual enzymes catalyzing the reactions. Finally, we compare prediction results with prior knowledge of reactions and their enzymes. It is noted that a few reactions fail to obtain any candidates by the candidate identification method adopted, due to low E-value resulting from insufficient sequence homology. In this case, we do not include them into our datasets.

3 Identification of candidate enzymes

The approach used by pathway-hole filler module in PathwayTools software [17] is adopted to identify candidate enzymes. We define the protein catalyzing a particular reaction in our dataset as has-function enzyme and otherwise no-function enzyme.

For each metabolic reaction, the procedure of candidate identification proceeds in the following manner. First, query from KEGG database other organisms, in which the reaction is present. We retrieve the desired organisms from 497 organisms that KEGG provides. Then both organism names and KEGG entry IDs of the genes encoding the actual enzymes are obtained. While in most cases one gene encodes one enzyme, sometimes there are more than one gene encoding one enzyme in certain organisms. Second, retrieve from KEGG protein sequences corresponding to the genes obtained from the previous step. Here we also use 'isozymes' to refer to those proteins with the same function in a variety of organisms as [7]. Third, search the whole target genome for sequences homologous to the query isozymes. BLAST [18, 19] is employed to do the homology search. Generally, the more frequent a candidate sequence is a hit; the more credible it is actual enzyme catalyzing the reaction. Finally, consolidate all BLAST hits into a final set of candidate proteins. A parameter vector is then calculated from the consolidation result for each candidate as its feature used for downstream prediction. The vector has d = 7 elements, composed of Shotgun-score, best E-value, average rank, average fraction aligned, pathway direction, adjacent-reactions, and average BLAST score, among which the definition and calculation of the first six parameters are described in detail in [7]. Since BLAST score is a direct measurement of sequence similarity and larger BLAST score usually indicates higher homology, therefore the average BLAST score is also introduced as one feature of each candidate.

4 Prediction Method

4.1 Problem formulation

Assume we have M known reactions. The reaction r_m has $q^{(m)}$ enzymes candidates, $m \in \{1,2,...,M\}$. Let the vector $[x_{t,1}\ x_{t,2}\ ...\ x_{t,d}] \overset{\Delta}{=} \mathbf{x}_t \in \mathrm{R}^d$denote the d-dimension parameter vector of the t-th sample, $t \in \{1,2,...,N\}$, then the overall data becomes as below, in which the underlining candidates represent actual enzymes catalyzing the reactions. The missing enzyme prediction is essentially a mapping problem in the field of pattern recognition. The aim is to find a function $f(\mathbf{x}_t)$ to approximate the mapping formulated as following:

$$f(\mathbf{x}_t) = \begin{cases} 1, & \text{if } \mathbf{x}_t \text{ is actual enzyme} \\ 0, & \text{if } \mathbf{x}_t \text{ is not actual enzyme} \end{cases} \tag{1}$$

We will use variable y_t to denote the output values $\{0,\ 1\}$ in the next subsection.

$$
\begin{array}{llcccccc}
 & & \multicolumn{4}{c}{\overbrace{\hphantom{x_{1,1}\quad x_{1,2}\quad \cdots \quad x_{1,d}}}^{\text{input parameter vector}}} & \overbrace{\hphantom{0}}^{\text{output}} \\
\text{Reaction 1} & \left\{\begin{array}{l} \underline{\text{candidate 1}} \rightarrow \mathbf{x}_1 : \\ \text{candidate 2} \rightarrow \mathbf{x}_2 : \\ \cdots \\ \text{candidate } q^{(1)} \rightarrow \mathbf{x}_{q^{(1)}} : \end{array}\right. & \begin{array}{c} x_{1,1} \\ x_{2,1} \\ \cdots \\ x_{q^{(1)},1} \end{array} & \begin{array}{c} x_{1,2} \\ x_{2,2} \\ \cdots \\ x_{q^{(1)},2} \end{array} & \begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} & \begin{array}{c} x_{1,d} \\ x_{2,d} \\ \cdots \\ x_{q^{(1)},d} \end{array} & \begin{array}{c} 1 \\ 0 \\ \cdots \\ 0 \end{array} \\
\text{Reaction 2} & \left\{\begin{array}{l} \underline{\text{candidate 1}} \rightarrow \mathbf{x}_{q^{(1)}+1} : \\ \text{candidate 2} \rightarrow \mathbf{x}_{q^{(1)}+2} : \\ \cdots \\ \text{candidate } q^{(2)} \rightarrow \mathbf{x}_{q^{(1)}+q^{(2)}} : \end{array}\right. & \begin{array}{c} x_{q^{(1)}+1,1} \\ x_{q^{(1)}+2,2} \\ \cdots \\ x_{q^{(1)}+q^{(2)},1} \end{array} & \begin{array}{c} x_{q^{(1)}+1,2} \\ x_{q^{(1)}+2,2} \\ \cdots \\ x_{q^{(1)}+q^{(2)},2} \end{array} & \begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} & \begin{array}{c} x_{q^{(1)}+1,d} \\ x_{q^{(1)}+2,d} \\ \cdots \\ x_{q^{(1)}+q^{(2)},d} \end{array} & \begin{array}{c} 1 \\ 0 \\ \cdots \\ 0 \end{array} \\
\cdots & \qquad \cdots & & \cdots & & & \\
\text{Reaction M} & \left\{\begin{array}{l} \underline{\text{candidate 1}} \rightarrow \mathbf{x}_{N-q^{(M)}+1} : \\ \text{candidate 2} \rightarrow \mathbf{x}_{N-q^{(M)}+2} : \\ \cdots \\ \text{candidate} q^{(M)} \rightarrow \mathbf{x}_N : \end{array}\right. & \begin{array}{c} x_{N-q^{(M)}+1,1} \\ x_{N-q^{(M)}+2,1} \\ \cdots \\ x_{N,1} \end{array} & \begin{array}{c} x_{N-q^{(M)}+1,2} \\ x_{N-q^{(M)}+2,2} \\ \cdots \\ x_{N,2} \end{array} & \begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} & \begin{array}{c} x_{N-q^{(M)}+1,d} \\ x_{N-q^{(M)}+2,d} \\ \cdots \\ x_{N,d} \end{array} & \begin{array}{c} 1 \\ 0 \\ \cdots \\ 0 \end{array} \\
 & & \multicolumn{5}{c}{\underbrace{\hphantom{x_{1,1}\quad x_{1,2}\quad \cdots \quad x_{1,d}\quad 0}}_{\text{data}}}
\end{array}
$$

4.2 Reversible jump MCMC learning of regression model

As stated in subsection 4.1, given a list of candidate enzymes identified according to certain biological evidences, missing enzyme prediction is basically a mapping formulated by equation (1). The mapping problem can be written in general notation as f:$\mathbf{x} \rightarrow y$. Suppose we have a group of N input-output observations (candidates):

$$O = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N; y_1, y_2, ..., y_N\} \tag{2}$$

We postulate the following multivariate-input, single-output mapping:

$$y_t = f(\mathbf{x}_t) + n_t \tag{3}$$

Where $\mathbf{x}_t \in \mathrm{R}^d$denotes a set of d-dimension input vectors, $y_t \in \mathrm{R}$ is single variable output, n_t stands for noise, $t \in \{1, 2, ..., N\}$. The purpose of the learning is to compute an approximation to the function f and estimate the characteristics of the noise process. We consider a mixture model M, consisting of a mixture of J RBFs and a linear term [24], represented as:

$$M_0: \; y_t = b + \beta^T \mathbf{x}_t + n_t, \; J = 0 \tag{4}$$

$$M_J: \; y_t = \sum_{j=1}^{J} a_j \phi(||\mathbf{x}_t - \mu_j||) + b + \beta^T \mathbf{x}_t + n_t, \;\; 1 \le J \le J_{\max} \tag{5}$$

where J_{max} is the maximum number of basis functions (J_{max} is set as 40), $\mu_j \in \mathrm{R}^d$ denotes the j-th RBF center, $a_j \in \mathrm{R}$ the amplitude of the j-th RBF, $b \in \mathrm{R}$ and $\beta \in \mathrm{R}^d$ the linear regression parameters, and the noise $n_t \in \mathrm{N}(0, \sigma^2)$ is assumed to be i.i.d. Gaussian. $||.||$ is the Euclidean distance metric. $\phi(\rho) = \exp(-\rho^2)$ is chosen as the basis function in our experiments. The space of the radial basis centers Ω_J is defined as:

$$\Omega_J \overset{\Delta}{=} \{\mu = [(\mu_{1,1}, \cdots, \mu_{1,d});\ \cdots;(\mu_{J,1}, \cdots, \mu_{J,d})];\ \mu_{j,i} \in [\min x_{l,i} - 0.1, \max x_{l,i} + 0.1], \\ j = 1, \cdots, J;\ i = 1, \cdots, d;\ l = 1, \cdots, N\} \tag{6}$$

And define$\Omega \overset{\Delta}{=} \cup_{J=0}^{J_{\max}} \{J\} \times \Omega_J$. For notational convenience, equations (4) and (5) can be expressed in a vector-matrix fashion:

$$\begin{bmatrix} y_1 \\ y_2 \\ \\ \vdots \\ \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} \ldots x_{1,d} & \phi(||\mathbf{x_1} - \boldsymbol{\mu_1}||) \cdots \phi(||\mathbf{x_1} - \boldsymbol{\mu_J}||) \\ 1 & x_{2,1} \ldots x_{2,d} & \phi(||\mathbf{x_2} - \boldsymbol{\mu_1}||) \cdots \phi(||\mathbf{x_2} - \boldsymbol{\mu_J}||) \\ & & \\ \vdots & \vdots & \vdots \\ & & \\ 1 & x_{N,1} \ldots x_{N,d} & \phi(||\mathbf{x_N} - \boldsymbol{\mu_1}||) \cdots \phi(||\mathbf{x_N} - \boldsymbol{\mu_J}||) \end{bmatrix} \begin{bmatrix} b \\ \beta_1 \\ \vdots \\ \beta_d \\ a_1 \\ \vdots \\ a_J \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \\ \vdots \\ \\ n_N \end{bmatrix} \tag{7}$$

That is,

$$\mathbf{y}_{N\times 1} = \mathbf{D}_{N\times(1+d+J)} . \alpha_{(1+d+J)\times 1} + \mathbf{n}_{N\times 1} \tag{8}$$

We assume that the number J of RBFs and the parameters $\theta_J \overset{\Delta}{=} \{\alpha, \mu, \sigma^2\}$ are unknown. Given a set of observations O, our goal is to estimate J and θ_J. Bayesian inference is used to estimate the unknown parameters J and θ_J. Hyper-parameter $\Lambda, \delta^2 \in \mathrm{R}^+$ are introduced and presumed to be independent of each other. Moreover, $\sigma^2$and δ^2 are assumed to have inverse-Gamma distribution, *i.e.*$\sigma^2 \sim IG(0,0)$, $\delta^2 \sim IG(2,10)$, and Λ has Gamma distribution, *i.e.* $\Lambda \sim Ga(0.5,0)$. According to Bayes theorem, the joint posterior distribution can be formalized as:

$$p(J, \alpha, \mu, \sigma^2, \Lambda, \delta^2 | \mathbf{x}, \mathbf{y}) \propto p(\mathbf{y} | J, \alpha, \mu, \sigma^2, \Lambda, \delta^2, \mathbf{x}) . p(J, \alpha, \mu, \sigma^2, \Lambda, \delta^2) \tag{9}$$

where $p(\mathbf{y}|J, \alpha, \mu, \sigma^2, \Lambda, \delta^2, \mathbf{x})$ is the likelihood and $p(J, \alpha, \mu, \sigma^2, \Lambda, \delta^2)$ is the prior distribution. The likelihood for model (8) is:

$$p(\mathbf{y}|J, \theta_J, \Lambda, \delta^2, \mathbf{x}) = (2\pi\sigma^2)^{-N/2} \exp(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{D}.\alpha)'(\mathbf{y} - \mathbf{D}.\alpha)) \tag{10}$$

The prior distribution $p(J, \alpha, \mu, \sigma^2, \Lambda, \delta^2)$ is:

$$\begin{aligned} & p(J, \alpha, \mu, \sigma^2, \Lambda, \delta^2) = \\ & p(\alpha | J, \mu, \sigma^2, \Lambda, \delta^2) . p(\mu | J, \sigma^2) . p(J | \sigma^2, \Lambda, \delta^2) . p(\sigma^2) . p(\Lambda) . p(\delta^2) \end{aligned} \tag{11}$$

After standard probability marginalization and transformation, the joint posterior distribution (9) can be obtained as the following expression:

$$\begin{aligned} & p\left(J, \alpha, \mu, \sigma^2, \Lambda, \delta^2 | \mathbf{x}, \mathbf{y}\right) \\ & \propto \left[(2\pi\sigma^2)^{-N/2} \exp\left(-\tfrac{1}{2\sigma^2}\left(\mathbf{y} - \mathbf{D}.\alpha\right)'\left(\mathbf{y} - \mathbf{D}.\alpha\right)\right)\right] \left[\left|2\pi\sigma^2\Sigma\right|^{-1/2} \exp\left(-\tfrac{1}{2\sigma^2}\alpha'\Sigma^{-1}\alpha\right)\right] \\ & \times \left(-\tfrac{1}{\sigma^2}\right)\left(-\tfrac{1}{\Lambda^2}\right)\left[\tfrac{\mathrm{II}_{\Omega}(J,\mu)}{\zeta^J}\right]\left[\tfrac{1}{\delta^2}\exp(-\tfrac{10}{\delta^2})\right]\left[\tfrac{\Lambda^J/J!}{\Sigma_{j=0}^{J_{\max}}\Lambda^j/j!}\right] \end{aligned} \tag{12}$$

Where $\Sigma^{-1} = \delta^{-2}\mathbf{D}'\mathbf{D}$ and $II_{\Omega} = (J,\mu)$ is the indicator function of the set Ω (1 if $(J,\mu) \in \Omega$, 0 otherwise). One might select model order J by $\arg\max p(J|\mathbf{x},\mathbf{y})$ with J$\in\{0,\ldots,$ $J_{max}\}$, and also can perform parameter estimation by computing the conditional expectation $\mathrm{E}(\theta_J|J,\mathbf{x},\mathbf{y})$ based on (12). However, it is difficult to obtain these quantities analytically, as it involves integrals of high-dimension of nonlinear functions. Therefore, the reversible jump MCMC method was proposed to perform necessary Bayesian computation in [20]. The principle of MCMC is to draw random samples from an ergodic Markov chain $(J^{(i)},\theta_J^{(i)},\Lambda^{(i)},\delta^{2(i)})_{i\in\mathrm{N}}$ whose equilibrium distribution is the target posterior distribution. The initial values of $\mu_1,...,\mu_J$ are randomly chosen according to (6) and initial value of J is J_{max}. The Markov chain generates$L \gg 1$ sampling points, asymptotically convergent to the posterior distribution. We discard the points resulted from the initial steps, which is so-called in-birth period, and keep the last P steps for the computation. Here we set L = 2000 and P = 1000. Given a test sample $\mathbf{x}_{N+1}$, y_{N+1} can be then evaluated by:

$$\begin{aligned}&\hat{y}_{N+1} = \hat{E}(y_{N+1}|\mathbf{x}_1,\cdots,\mathbf{x}_{N+1},y_1,\cdots,y_N) = \\ &T.\tfrac{1}{P}\textstyle\sum_{i=1}^{P}\mathbf{D}(\mu^{(i)},\mathbf{x}_{N+1}).E(\alpha|J^{(i)},\mu^{(i)},\sigma_J^{2(i)},\delta^{2(i)},\mathbf{x}_1,\cdots,\mathbf{x}_N,y_1,\cdots,y_N)\end{aligned} \tag{13}$$

where T is a threshold for determining whether $\hat{y}_{N+1}$ is nearer to 1 or 0. If $|\hat{y}_{N+1}$-$1| \leq |\hat{y}_{N+1}|$, the test sample is predicted as has-function enzyme. The reversible jump MCMC sampler is able to sample directly from the joint distribution and jump between subspaces of different dimensions. A Metropolis-Hasting (MH) algorithm is performed, in which candidates are proposed based on proposal distributions. The candidates are randomly accepted according to an acceptance ratio that ensures reversibility and invariance of Markov chain with respect to the posterior distribution.

5 Results and Discussion

To evaluate the predictive power of the proposed model, we compare its performance with several other models: SVR, Bayesian network model, perceptron, and BP neural network, among which Bayesian network was previously adopted by Green *et al* [7].

5.1 Candidate identification

For each reaction in the dataset, we performed the procedure depicted in section 3. For the *E.coli* data, totally 3349 candidates are identified for 100 reactions and 121 out of the 3349 actually catalyze corresponding reactions. For *Vch*, *Mtu* and *Ccr* data, 2592 candidate proteins are identified for the overall 60 reactions and 72 out of the 2592 are actual. Note that the amount of no-function candidates is considerably larger than that of has-function candidates. Table 1 shows an example of candidate identification results. A group of seven sequences from the *E.coli* genome, i.e., b1850, b1581, b2247, b2871, b3686, b4477, and b4478 are identified as candidate proteins possessing 2-dehydro-3-deoxy-6-phosphogalactonate aldolase activity (E.C. 4.1.2.21), among which the one shown in bold (b4477) has been experimentally identified [21].

Table 1: Candidate identification result for 2-dehydro-3-deoxy-6-phosphogalactonate aldolase

GeneID	Shotgun score	Best E-value	Average score	Average Fraction aligned	Average rank	Pathway direction	Adjacent rxns
b1850	15	2e-14	63.18	0.27	2	0	0
b1581	4	1e-46	170.25	0.29	2	0	0
b2247	4	2e-20	87.83	0.28	3	0	0
b2871	4	0.65	28.3	0.27	3.25	0	0
b3686	1	0.29	29.6	0.41	3	0	0
b4477	15	5e-105	213.33	0.59	1	1	1
b4488	4	0	587.75	0.77	1	1	1

5.2 Cross validation

The cross validation herein is basically a binary classification test. For *E.coli* data, we randomly partitioned all the candidates into five separate groups and five-fold cross validation was then performed to evaluate the predictive power of five different models, i.e. reversible jump MCMC, SVR, perceptron, BP neural network, and Bayesian model. For *Vch*, *Mtu* and *Ccr* data, three-fold cross validation was applied.

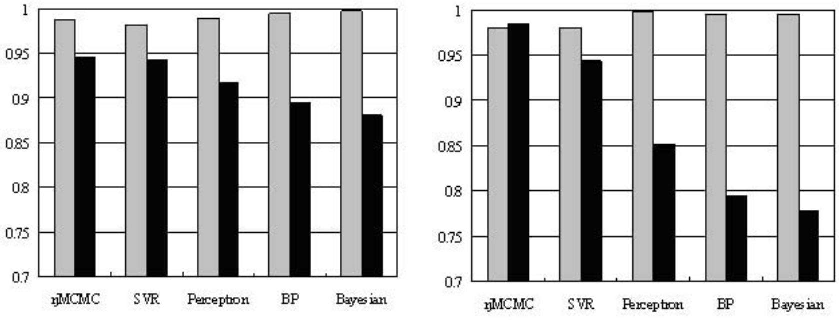
For a binary classification test, specificity and sensitivity are usually used for performance assessments. Specificity indicates the ability to correctly predict negative cases, i.e. true negative (TN) or no-function enzymes. Ssensitivity indicates the ability to correctly predict positive cases, i.e. true positive (TP) or has-function enzymes. In our experiments, both specificity and sensitivity are measured. However, we consider that sensitivity is more important for missing enzyme identification problem because the cost of a false positive is much less than the cost of a false negative.

5.3 Performance comparison

We tuned model parameters and chose the best performance in cross validation. The Fig. 1 shows the performance comparison of the five models. It can be observed that reversible jump MCMC model outperforms the other four in both datasets. The results are as expected because reversible jump MCMC contains both linear and non-linear term, it can be seen as a model between perceptron and back-propagation neural network. Moreover, SVR also exhibits favorable prediction power and it can be considered when model complexity is emphasized more than predictive performance.

Tables 2 shows the details of performance obtained from five models.

Moreover, we draw receiver operation characteristic (ROC) curves for reversible jump MCMC, SVR and Bayesian models to compare their prediction capability. Each ROC curve is created by plotting the number of TPs against that of FPs obtained by 50 gradually increasing thresholds for classification of has-function or no-function



(a) The *E.coli* data

(b) The *Mtu*, *Vch*, and *Ccr* data

Figure 1: Performance comparison, gray bars represent specificity and black represents sensitivity.

Table 2: The performance of five prediction models in *E.coli* and three other bacteria data.

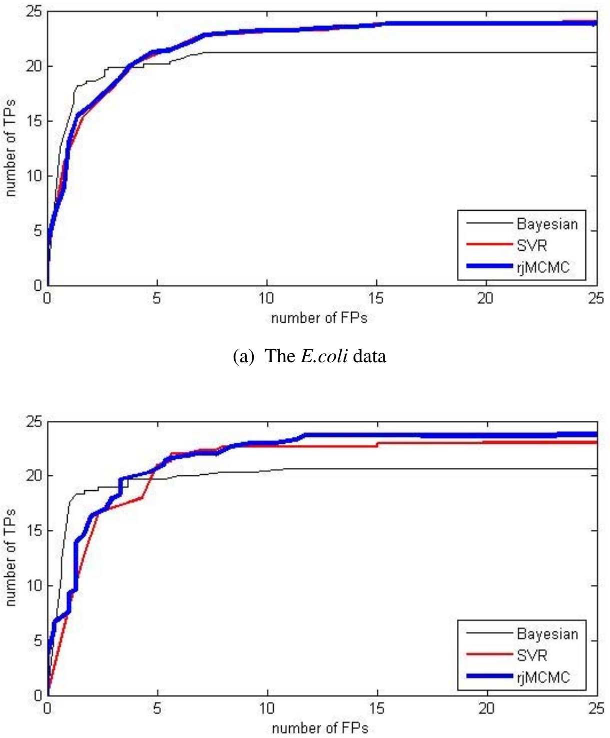
Models	*E.coli*		*Three other bacteria*	
	Specificity	Sensitivity	Specificity	Sensitivity
Reversible jump MCMC	0.9873	0.9456	0.9817	0.9841
SVR	0.9821	0.9422	0.9803	0.9441
Perceptron	0.9893	0.9183	0.9972	0.8518
BP neural network	0.9954	0.8946	0.9954	0.7947
Bayesian network	0.9975	0.8811	0.9945	0.7786

enzymes. The ROC comparison results of both datasets are shown in the Fig. 2.

Although the ROC curves indicate that Bayesian method performs better in lower number of FPs, reversible jump MCMC and SVR outperform Bayesian at higher number of FPs. As explained in section 5.2, we prefer a higher number of TPs at the cost of relatively high number of FP in this problem due to the more expensive experimental cost of a FN than a FP.

6 Conclusion

Computational identification of missing enzymes is important in metabolic network reconstruction. In this study, first, we adopted the approach of Green *et al* [7] to identify a list of candidate enzymes for each reaction and a feature vector is calculated for each candidate. Then, a regression model is proposed to predict whether these candidates are actual or not, in which a reversible jump MCMC technique is used to learn the model parameters. To evaluate the model, we applied it into known reactions occurring in *E.coli* and three other bacteria. We compared the method with



(a) The *E.coli* data

(b) The *Mtu*, *Vch*, and *Ccr* data

Figure 2: The ROC curves of reversible jump MCMC, SVR, and Bayesian model.

four comparable models: Bayesian network, SVR, perceptron, and BP neural network. The results indicate that the proposed method performs favorable.

Acknowledgment

The authors would like to acknowledge the collaboration with their collaborators in this effort, and, in particular, Dr. Santosh Kesari of the Dana-Farber Cancer Institute, Harvard Medical School. They would like to thank other members of the Center for Bioinformatics, Harvard Center for Neurodegeneration and Repair (HCNR) and Brigham and Women's Hospital, HMS for their technical comments. This research is partly sponsored by the Center for Bioinformatics Program Grant from HCNR, Harvard Medical School.

References

[1] J.A. Papin, T. Hunter, B.O. Palsson, S. Subramaniam. Reconstruction of cellular signaling networks and analysis of their properties. *Nature Rev. Mol. Cell Biol.*, 6, 99-111 ,2005.

[2] T. Finkel, J.S. Gutkind. Signal transduction and human disease. Wiley-Liss, Hoboken, New Jersey, 2003.

[3] H. Bono, H. Ogata, S. Goto, M. Kanehisa. Reconstruction of amino acid biosynthesis path-ways from the complete genome sequence. *Genome Res.*, 8, 203-210, 1998.

[4] T. Dandekar, R. Sauerborn. Comparative genome analysis and pathway reconstruction. *Pharmacogenomics*, 3, 245-256, 2002.

[5] A. Osterman, R. Overbeek. Missing genes in metabolic pathways: a comparative genomics approach. *Curr. Opin. Chem. Biol.*, 7, 238-51, 2003.

[6] S.J. Cordwell. Microbial genomes and 'missing' enzymes: redefining biochemical path-ways. *Arch. Microbiol.*, 172, 269-279, 1999.

[7] M.L. Green, P.D. Karp. A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. *BMC Bioinformatics*, 5:76, 2004.

[8] L. Chen, D. Vitkup. Predicting genes for orphan metabolic activities using phylogenetic profiles. *Genome Biol.*, 7:R17, 2006.

[9] P. Kharchenko, D. Vitkup, G.M. Church. Filling gaps in a metabolic network using exp-ression information. *Bioinformatics*, 20 suppl., i178-i185, 2004.

[10] P. Kharchenko, L. Chen, Y. Freund, D. Vitkup, G.M. Church. Identify metabolic enzymes with multiple types of association evidence. *BMC Bioinformatics*, 7:177, 2006.

[11] P.J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82, 711-732, 1995.

[12] A.J. Smola, B. Scholkopf. A tutorial on support vector regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, Univeristy of London, UK, 1998b.

[13] B. Widrow, M.A. Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proc. IEEE*, 78, 1415-1442, 1990.

[14] **Encyclopedia of Escherichia coli K12 Genes and Metabolism**: http://www.ecocyc.org.

[15] M. Kanehisa, S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, 28: 27-30, 2000.

[16] Caspi *et al.* MetaCyc: A multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Res.*, 34, D511-D516, 2006.

[17] P.D. Karp, S. Paley, P. Romero. The Pathway Tools software. *Bioinformatics*, 18, S225-232, 2002.

[18] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman. Basic Local alignment search tool. *J. Mol. Biol.*, 215, 403-410, 1990.

[19] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nuleic Acids Res.*, 25, 3389-3402, 1997.

[20] C. Andrieu, N. De Freitas, A. Doucet. Robust full Bayesian learning for radial basis networks. *Neural comput.*, 13, 2359-407, 2001.

[21] E.C.C. Lin. Dissimilatory Pathways for Sugars, Polyols, and Carboxylates. In Escherichia coli and Salmonella Cellular and Molecular Biology. ASM Press, 1996.