A Bayesian Approach to High-Throughput Biological Model Generation

Xinghua Shi[1] and Rick Stevens[1,2⋆]

[1] Department of Computer Science, University of Chicago, Chicago, IL 60637, USA.
shi@uchicago.edu
[2] The Computing, Environment and Life Science, Argonne National Laboratory, Argonne, IL 60439, USA.
stevens@anl.gov

Abstract. With the availability of hundreds and soon-to-be thousands of complete genomes, the construction of genome-scale metabolic models for these organisms has attracted much attention. However, manual work still dominates the process of model generation and leads to the huge gap between the number of complete genomes and genome-scale metabolic models. The challenge in constructing a genome-scale models from existing databases is that usually such a directly extracted model is incomplete and contains network holes. Network holes occur when a network is disconnected and certain metabolites cannot be produced or consumed. In order to construct a valid metabolic model, network holes need to be filled by introducing candidate reactions into the network. Toward the high-throughput generation of biological models, we propose a Bayesian approach to improving draft genome-scale metabolic models. A collection of 23 types of biological and topological evidence is extracted from databases the SEED [1], KEGG [2] and BiGG [3]. Based on these pieces of evidence, 23 individual predictors are created using Bayesian approaches. Afterwards, in order to combine these individual predictors and unify their predictive results, an ensemble of individual predictors is built on majority vote and four classifiers: Naive Bayes Classifier, Bayesian Network, Multilayer Perceptron Network and AdaBoost. A set of experiments is performed to train and test individual predictors and integrative mechanisms of single predictors, and evaluate the performance of our approach.

1 Introduction

The number of annotated genomes is approaching 1000, thanks to high-throughput sequencing technology in biology and automated genome annotation tools in bioinformatics. The availability of these complete genomes provides a significantly important way of analyzing genomes at system level. One type of such analysis has been carried out through the construction of a genome-scale metabolic model for a microorganism from its genome sequence. Starting from the extraction of gene-protein-reaction

⋆ The authors would like to thank Federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Contract No. HHSN266200400042C. This work was supported in part by the U.S. Dept. of Energy under Contract DE-AC02-06CH11357.

(GPR) associations from genome and pathway databases, it is possible to build genome-scale metabolic models using constraint-based approaches such as flux balance analysis [7, 8, 10].

However, a genome-scale metabolic model generated from directly extracting GPR associations from existing databases is usually incomplete and contains network holes. Network holes are those places where certain metabolites cannot be produced or consumed as there are no reactions to connect these metabolites. In order to construct a valid model, every metabolite should have fluxes pass through it. candidate reactions need to be introduced to fill network holes. A number of factors can lead to network holes such as missing genes, wrong or missing annotations, poor mappings from functions to biochemical reactions. At present, manual search for hole-filling candidates dominates the work of filling network holes in the construction of genome-scale metabolic models. Due to the huge volume of data available to act as evidence and the large scale of metabolic networks, manual work is time-consuming and labor-intensive. Up to now, approximately 27 genome-scale metabolic models have been constructed [6]. In comparison, the number of complete genomes is approaching 1000. With the exponential growth of the number of genomes, the gap between the number of genomes and the number of genome-scale metabolic models is expanding. In order to bridge this gap and generate 1000 genome-scale metabolic models in the near future, it is desirable that computational approaches be applied to fill network holes and thus accelerate the model-building process.

2 Related Work

Computational approaches have been proposed to improve metabolic networks or metabolic pathways. Green and Karp [14] showed a Bayesian approach to identify missing enzymes for filling pathway holes in Pathway/Genome databases (PGDB) by integrating evidence from homology, operon, and metabolic pathway relationships. However, the networks generated from their approach are incomplete at network level and there are still network holes in networks.

Kharchenko et al. [15] developed a computational approach for selecting candidate genes that can be assigned to missing metabolic enzymes, based on the gene expression data and structure of partially reconstructed metabolic network. Chen and Vitkup [12] presented a method that uses local structure of a metabolic network combining with phylogenetic profiles to suggest candidate genes for enzymes without corresponding genes. Kharchenko et al. [11] expanded their methods to include multiple types of functional association evidence, including clustering of genes on the chromosome, similarity of phylogenetic profiles, gene expression, protein fusion events and a local structure of metabolic network to infer genes encoding for a specific metabolic function. However, the collection of algorithms presented in [11, 12, 15] focuses on filling network holes where a single network hole is present in a neighborhood of metabolic networks. In practice, it is common that patches of network or a collection of holes occur in a metabolic network, especially in organisms whose genomes are not well annotated and there are many network holes in their metabolic networks. Therefore, these approaches are not suitable for the massive production of genome-scale metabolic models.

In [13], DeJongh *et al.* presented their tools for the generation of substantially complete metabolic networks for over 400 complete genome sequences currently in the SEED. Their tools are based on the notation of scenarios that represent segments of metabolic pathways with connected reactions accompanied by input and output compound sets. Assembling scenarios by connecting their input and output compound sets together, an organism-specific reconstruction of metabolic network can be then constructed. Although the work in [13] enhances the curation of associations between functional roles and reactions and thus generates better GPR associations for the reconstruction of genome-scale metabolic networks in the SEED. However, the reconstructed networks produced by their tools are still partially complete and include many network holes. Therefore, in order to obtain a complete metabolic network and then build a valid model, further work is needed to fill network holes in these draft metabolic networks.

3 Our Approach and Tools

The Rapid Annotation using Subsystem Technology (RAST) server [18] provides a rapid and fully automated annotations for bacterial genomes. Users can submit their new genomes and normally, RAST will complete annotations and make the annotated genome available within 12–24 hours. The SEED [1, 17] provides an environment and tools that curate function assignments based on subsystems. The work in [13] generates a more accurate reaction set for an annotated genome using the technology of metabolic scenarios tightly coupled with subsystems. The work in [13] also provides hundreds of draft metabolic models that are can be further improved.

Following the efforts in [13], we design a set of computational tools to improve draft metabolic models that have generated, with an aim of the high-throughput generation of complete genome-scale metabolic models. This toolset includes mainly four parts: parsers to integrate and reconcile data from different databases, network hole detectors that analyze network connectivity and identify network holes, evidence extractors that mine through integrated data and extracts pieces of evidence out of the data, and a set of predictors and the ensemble of predictors that use evidence to suggest candidate hole-filling reactions. In this paper, we focus on the later two sets of tools for extracting evidence and construction of predictors. The hole-filling mechanism proposed in this paper looks at the genome-scale metabolic model gained from an organism's genome annotations and all pieces of evidence mining extensively through known information in existing databases. By expanding a draft model exhaustively in every direction, we seek to improve genome-scale metabolic models and enable mass production of metabolic models.

Without any prior knowledge, a biochemically possible reaction can be assumed to be randomly distributed. In other words, the probability of including any reaction in a model is treated as $\frac{1}{|R|}$ if all biochemically possible reaction in KEGG, R, is considered. However, we know that, in fact, the probability of each reaction is highly skewed in different datasets. It is natural that some reactions happen more often than others. Therefore, the probabilities of reactions should be adjusted after seeing certain datasets. And these adjusted probabilities can be viewed as priors to pump into the calculation of

adjusted reaction probabilities after seeing other datasets. In this paper, reaction probabilities are adjusted by using Bayes' rule.

3.1 Evidence Extraction

In order to generate a candidate reaction list that can be incorporated to complete a partial metabolic model, a set of evidence should be extracted from available data. As shown in Table 1, there are totally 23 pieces of evidence, within seven different types, that extracted from different data sources. They are (a) reaction priors, (b) the co-occurrence of reaction pairs, (c) segment priors, and (d) the co-occurrence of reaction segment pairs for datasets in two reconstructed BiGG models, $iJR904$ and $iSB619$; KEGG reference pathway map; KEGG modules; KEGG organism maps; and all draft models in the SEED; as well as (e) the co-occurrence of gene pairs in all organisms in the SEED, (f) the co-occurrence of gene pairs in the SEED, and (g) the co-occurrence of gene-genes pairs in the same clusters on chromosomes in the SEED.

Table 1. Summary of Evidence

Type	BiGG Models	KEGG Ref Map	KEGG Modules	KEGG Org Maps	SEED Orgs
(a)	[1] 634	[2] 4,953	[3] 434	[4] 3,324	[5] 1,318
(b)	[6] 893	[7] 5,358	[8] 1,765	[9] 10,178	[10] 3,036
(c)	[11]5,175	[12]36,145	[13]11,607	[14] 237,903	[15] 70,326
(d)	[16]16,421	[17]101,101	[18]36,831	[19]1,027,801	[20] 326,834
(e)	/	/	/	/	[21] 376,880
(f)	/	/	/	/	[22] 139,183
(g)	/	/	/	/	[23] 302,664

Table 1 lists the 23 pieces of evidence and their corresponding statistics. Each row is a type of evidence and each column is a dataset.The content of each cell represents the index of an evidence, followed by the number of data points in the dataset. For example, "$[1]634$" means that for the first ($[1]$) evidence, which is reaction priors in BiGG Models, there are 634 reactions in the dataset. The details of each type of evidence is described as follows.

(a) *Reaction Priors:* For any reaction r in the KEGG (all reactions in KEGG are denoted as R), if it has been seen in existing pathway maps with some prior probability, then these priors can be used to infer the probability of including r as a hypothetical reaction in a model. Reaction prior $Pr(r)$ for any reaction r in any dataset, from , is calculated using the ratio between the frequency of r and the dataset size.

(b) *Co-Occurrence of Reaction Pairs:* For any reaction $r \in R$, if it co-occurs with another reaction in existing pathway maps, then the probabilities of their co-occurring can be used to infer the probability of including r as a candidate reaction. A pair of reactions, (r_1, r), is called to co-occur if there is a set of one or multiple common compounds, noted as compound set C, among primary products of reaction r_1 and primary substrates of reaction r. The conditional co-occurrence of a pair of reaction

(r_1, r), $Pr(r|r_1)$, is defined as the frequency of reaction pair $r_1 - C - r$, divided by the frequency of reaction r_1. This co-occurrence indicates the probability of inferring r after seeing r_1 in the dataset.

(c) *Segment Priors:* A pathway segment is a linear sequence of reactions connected by common compounds. Every pathway map is decomposed into a set of pathway segments including from two to six reactions. For any reaction $r \in R$, if it has been seen in pathway segments of existing maps with some prior probability, then these priors can be used to infer the probability of including r as a hypothetical reaction. Segment priors are calculated by dividing the frequency of segments by the size of dataset.

(d) *Co-Occurrence of Reaction-Segment Pairs:* For any reaction $r \in R$, if it co-occurs with a pathway segment in existing maps, then the probabilities of their co-occurring can be used to infer the probability of including r as a candidate reaction. A reaction-segment pair, (r, s), is said to co-occur if the products of reaction r have one or multiple common compounds with the substrates of the first reaction in the segment s, or the opposite, the substrates of reaction r have one or more common compounds with the products of the last reaction in the segment s. The conditional co-occurrence of a reaction-segment pair (r, s), $Pr(r|r_1)$, is defined as the frequency of segment $r - C - s$ or $s - C - r$, divided by the frequency of segment s. This co-occurrence indicates the probability of inferring r after seeing s in the dataset.

(e) *Gene Co-Occurrence in Complete SEED Organisms:* For any gene g, if it co-occurs with another gene g_1 in known genomes, then the probabilities of their co-occurrence can be used to infer the probability of including g as a candidate gene that may encode for some reaction to fill a network hole. The co-occurrency of gene pairs is calculated by dividing the number of organisms that a pair of genes occur by the number of organisms that one gene occur. In account of homology across many organisms, protein families in the SEED, denoted as FIGfams, are used in the calculations of gene co-occurrence.

(f) *The Co-Occurrence of Gene-Genes Pairs on Gene Clusters in SEED Organisms:* A gene cluster is a group of genes that sit close to each other on chromosomes of an organism. Metabolic genes sitting on the same cluster tend to encode reactions that also construct a pathway segment. The co-occurrence of gene-genes pairs in clusters is calculated by the number of organisms that a gene-genes pair occur divided by the number of organisms that the set of genes occurs.

(g) *Gene Co-Occurrence on Gene Clusters in SEED Organisms:* The co-occurrency of gene pairs in is calculated by the ratio of the number of organisms that a pair of genes sitting on the same cluster and the number of organisms that one gene occur.

3.2 Predictor Construction

Faced with the challenge of searching for hole fillers in large volume of data, it is desirable to build computational predictors that infer plausible candidate reactions to reconcile network holes, based on known knowledge. Specifically, the 23 pieces of evidence extracted above are used, and 23 individual predictors are built according to each evidence. These predictors are of seven types according to the seven types of evidence they use.

(a) *Predictors Using Reaction Priors:* Five individual predictors, P_1 – P_5, are constructed to reflect reaction priors from five different datasets of BiGG models, KEGG reference pathway map, KEGG network modules, KEGG organism pathway maps, and SEED draft models. The underlying idea of these five predictors is that reactions with higher priors in known datasets are more likely to be selected as candidate reactions than those with lower priors. Each predictor based on priors from different datasets generates a set of candidate reactions. A scoring function $S(r)$ of a predictor in this group, for any r in one dataset, is set to be $Pr(r)$, which is the prior of reaction r. With scores assigned to all reactions r in the dataset, these reactions are ranked by their scores $S(r)$, and only those reactions with high scores are selected as candidate reactions, noted as R_c.

(b) *Predictors Using Co-Occurrence of Reaction Pairs:* Five predictors, with index of P_6 to P_{10}, are built to include the five pieces of evidence extracted from co-occurrence of reaction pairs in the five datasets. A scoring function $S(r)$ is defined in Equation 1 for each of these five predictors. Let's denote all reactions in one of the five datasets as R_d and all reactions in the draft model as R_m. For each reaction pair (r, r_1), where $r_1 \in R_d, r_1 \notin R_m$ and $r \in R$, the co-occurrence of (r, r_1), $Pr(r|r_1)$ is calculated . In order to capture all the possible reactions co-occur with reaction r in a local neighborhood of r, the process of calculating $Pr(r|r_1)$ is proceeded in 5 runs. In each run, all the reactions r that co-occur with any reaction r_1 that is in the dataset but not in the model are considered. The reaction r with highest co-occurrency with r_1 is inserted into the reaction list R_m and removed from R_d. At the same time, this highest co-occurrency is recorded. Then with the updated R_m and R_d, the probability of $Pr(r|r_1)$ is re-calculated. After 5 runs, the product of all the five highest co-occurrency scores is obtained for any reaction r. And the greatest co-occurrency product that contains r is chosen as the score of r and the pathway of reactions that is corresponding to this score is considered as a candidate hole filler. All reactions in such a pathway are considered as candidate reactions if this pathway has a high score. This makes sure that reactions in five steps away from r are considered. The number 5 is selected since a pathway segment with at most 6 reactions are used in this work, which in turn is chosen between the tradeoff of computational cost and the predictive capability of neighbor steps away. After sorting all reactions $r \in R$ by their scores $S(r)$, a list of candidate reactions R_c that have high scores is selected.

$$S(r) = \max \prod_{i=1}^{5} \max_{r_1 \in R_d, r_1 \notin R_m} Pr(r|r_1) \tag{1}$$

(c) *Predictors Using Segment Priors:* Five predictors based on segment priors in the five datasets are constructed, with abbreviations of P_{11} to P_{15}. For all each reaction $r \in R$, search for all the segments s in the dataset that contain reaction r, sort these segments s by their priors $Pr(s)$ and assign the maximal probability of these segments, to the score of r.

(d) *Predictors Using Co-Occurrence of Reaction-Segment Pairs:* Five predictors, with index from P_{16} to P_{20}, are built according to reaction-segment co-occurrence in the five datasets. Each of such predictor infers a set of candidate reactions that co-occur and connect, via a common compound set, with some segments in the datasets.

(e) *Predictors Using Gene Co-Occurrence in Complete SEED Organisms:* The gene co-occurrence in all complete genomes of the SEED is applied build a predictor P_{21}. A scoring function is designed to assign a score, $S(r)$, to each reaction $r \in R$ based on the corresponding gene's co-occurrence with other genes in the model. Just as the mechanism of calculating scores for reaction co-occurrency, the genes within five steps away from a given gene are considered to cover the local neighborhood.

(f) *Predictors Using The Co-Occurrence of Gene-Genes Pairs on Gene Clusters in SEED Organisms:* Predictor P_{22} is constructed using the co-occurrency of gene-genes pairs in the same gene clusters of all SEED organisms. The assumption here is that if one gene g sits in the neighborhood of a collection of genes g_s on the chromosomes in many organisms, then when seeing a set of neighbor genes g_s in a new organism, we can propose that g is also present in the new organism.

(g) *Predictors Using Gene Co-Occurrence on Gene Clusters in SEED Organisms:* Predictor P_{23} is built based on the gene co-occurrency on the same gene clusters in SEED organisms. The score of r, $S(r)$ can be calculated from the co-occurrence of any gene it's associated with other genes in the same gene clusters.

In summary, 23 reaction predictors of seven types are constructed to suggest candidate genes to fill network holes.

3.3 Ensemble of Predictors

Individual predictors based on various evidence may produce inconsistent or incorrect predictions. In order to improve the predictive accuracy and resolve inconsistence in individual predictors, ensemble methods need to be incorporated to integrate individual predictors. One simple assembly of individual predictors is to retrieve the results of each predictor and pick the reactions that are predicted by most predictors. Assume there are non-selfish and nonbiased behaviors among all individual predictors.

An alternative approach of integrating individual predictors is treating the selection of candidate hole-filling reactions as a classification problem. In this scenario, two classes, class 0 and class 1 are considered. Any reaction in KEGG ($r \in R$) is assigned to class 1 if it is a hypothetical reaction that should be included in a network, and dedicated to class 0 otherwise. Each individual predictor generates a score that represents a corresponding attribute; hence 23 individual predictors produce 23 attributes. Every instance is a reaction that includes all the 23 attribute values and one extra flag that indicates the class this reaction belongs to. After reducing the hole-filler problem to a classification problem, many classifiers in machine learning can be applied. Four such classifiers—naive Bayes classifier, Bayesian network, multilayer perceptron network, and boosting mechanism—are used in this paper.

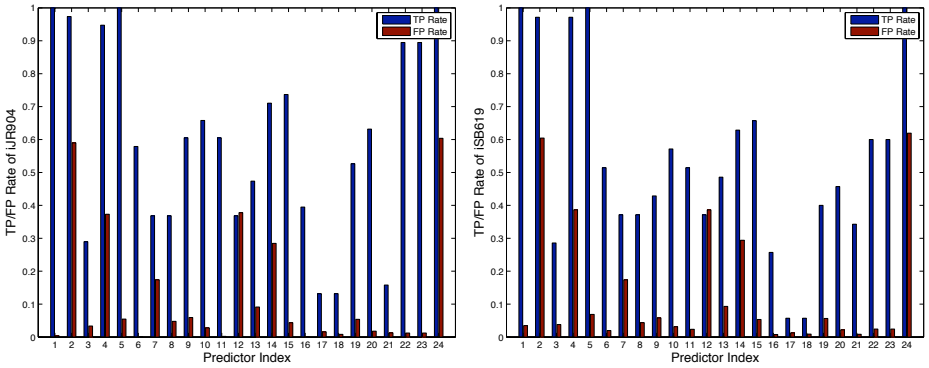
4 Experiments and Results

To evaluate the set of computational tools, two groups of experiments are designed. The first group of experiments is for a self-consistency check on two reconstructed models of $iJR904$ [9] for *Escherichia coli K-12* and $iSB619$ [19] for *Staphylococcus aureus N315*. A set of core metabolic genes selected from [16] is removed from these two

models and examined to see how well the predictors would fill network holes caused by the knockout of these core genes. The second set of experiments starts with removing 10% of the reactions in a model at a time, eventually removing 80% of the reactions in the model and see the change of recovery rates.

4.1 Results of Core Knockouts

4.1.1. Results of Individual Predictors and Majority Vote Integrator



(a) $iJR904$ (b) $iSB619$

Fig. 1. TP/FP Rates of Individual Predictors and Majority Vote on Reconstructed iJR904 and iSB619 Model

Figure 1 shows the true positive (TP) rate and false positive (FP) rate of each individual predictor and the majority vote of individual predictors on the reconstructed $iJR904$ model (Figure 1(a)) and $iSB619$ model (Figure 1(b)). The x axis shows the predictor indices, with 1 to 23 representing individual predictors and 24 representing the majority vote integrator of all individual predictors. The details of all individual predictors indexed from 1 to 23 are explained in Table 1. The y axis indicates the true positive rate (shown as the first bar, in blue, in each group) or false positive rate (shown as the second bar, in red, in each group) for each predictor or majority vote integrator.

From the results of the reconstructed $iJR904$ model in Figure 1(a), 12 predictors are considered as good predictors, as their true positive rates are greater than 0.5 and the false positive rates are smaller than 0.1. However, predictor P_1, P_6, P_{11} and P_{16} should be excluded. The datasets these four predictors use are reaction priors, reaction co-occurrency, segment prior, reaction-segment co-occurrency in two reconstructed $iJR904$ and $iSB619$ models in the BiGG. Hence, these four predictors are biased largely. Then the remaining 8 predictors have good performance in this set of experiments. They are P_5, P_9,P_{10}, P_{15}, P_{19}, P_{20}, P_{22}, and P_{23}. Another set of predictors are considered to have fair performance since they have high true positive (with

$TP\ rate \geq 0.5$) and relatively high false positive rates (with false positive rates between 0.1 and 0.4) at the same time. This set of predictors includes P_4 using reaction priors in KEGG organism maps,P_{13} using segment priors in KEGG modules, and P_{14} using segment priors in KEGG organism maps. The majority vote integrator, P_{24} can recover all knockout reactions if one reaction is said to be recovered by this integrator if it's recovered by any single predictor. When P_{24} is set to recover a knockout reaction if more than half of individual predictors have voted for that reaction, then the true positive rate is reduced to 0.45 and the false positive rate decreases to a very small ratio of 0.008.

The results of the reconstructed $iSB619$ model in Figure 1(b) show a similar pattern for the performance of all predictors. Especially, if majority vote integrator P_{24} is said to recover a knockout reaction when more than half of individual predictors have voted for that reaction, then the true positive rate is 0.57 and the false positive rate decreases to a very small ratio of 0.001. This shows that majority vote is a good since it is capable to detect approximately 57% of knockout reactions while keeps the false positive rate very low. What's more, we can also conclude that predictors use SEED information, such as predictors P_5, P_{10}, P_{15}, P_{20}, P_{22}, and P_{23}, perform well. In addition, the results show the evidence from gene clusters, used by P_{22} and P_{23}, provides strong strength to suggest candidate hole-fillers.

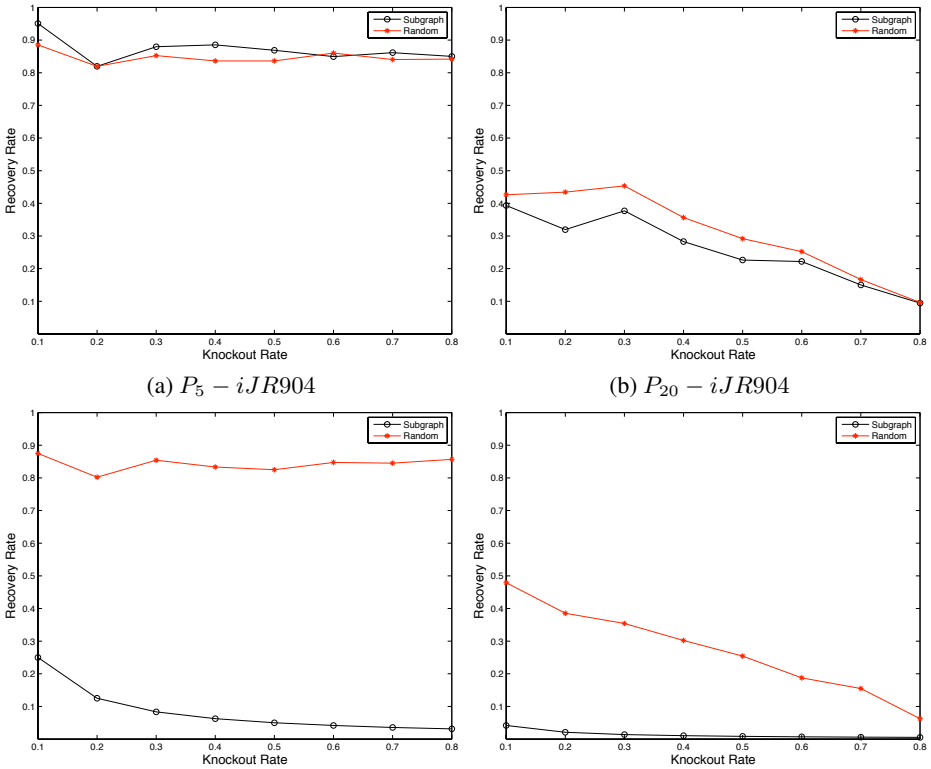
4.1.2. Classifier Results

Weka [5], a data mining software is used to train and test four classifiers including naive Bayes classifier, Bayes network, multilayer perceptron network, and AdaBoostM1. Table 2 summarizes the performance of these four classifiers on the class of 1, which is a class of candidate hole-filling reactions. The classifiers are abbreviated as "NB," "BN," "MLP," and "AB," respectively. The first part is the training error, and the second part is the performance on stratified cross-validation. In each part, three rows represent three different measurements of the performance of classifiers and there are accuracy, which is the ratio of correctly classified instances in the test dataset, the true positive rates and false positive rates. The results in Table 2 show that four classifiers have high accuracy in both training and cross-validation test. Also note that for any of the four classifiers, the cross-validated accuracy is close to the training set accuracy. We thus conjecture that the classifiers do not overfit the training set [5].

Table 2. Preformance of Different Classifiers on Core Knockout Results of $iJR904$ Model

Classifier		NB	BN	MLP	AB
Training	Accuracy(%)	99.1487	99.236	99.8254	99.9127
	TP	0.863	0.605	0.816	0.921
	FP	0.007	0.004	0	0
Cross Val.	Accuracy(%)	99.0177	99.0613	99.6726	99.8035
	TP	0.816	0.632	0.684	0.868
	FP	0.008	0.006	0.001	0.001

Table 3 shows that all four classifiers have good testing results. Each classifier has accuracy greater than 98%, the TP rate is relatively high and FP rate is low. This obser-



(a) $P_5 - iJR904$ (b) $P_{20} - iJR904$

(c) $P_5 - iSB619$ (d) $P_{20} - iSB619$

Fig. 2. Recovery Rate of Random and Subgraph-Based Knockouts

vation demonstrates that classifiers trained on core knockouts of $iJR904$ is capable of recovering core metabolic reactions in $iSB619$ model.

Table 3. Preformance of of $iSB619$ Model for Classifiers Training on $iJR904$ Model

Classifier		NB	BN	MLP	AB
Test	Accuracy(%)	98.7532	99.2261	99.1617	98.9467
	TP	0.343	0.543	0.743	0.914
	FP	0.008	0.004	0.006	0.01

4.2 Results of Random and Subgraph-Based Knockouts

Figure 2 shows the recovery rate of random and subgraph-based knockouts for two predictors on the reconstructed $iJR904$ and $iSB619$ models. The two predictors are P_5 using reaction priors and P_{20} which uses reaction-segment co-occurrence in SEED draft models. In each figure, the x axis is the knockout rate, which is the fraction of the number of knockout reactions and the total number of reactions in the original model. The y axis is the recovery rate of a predictor for corresponding knockout. The black

line with circled points represents the change of recovery rate over subgraph-based knockouts. The red line with starred points denotes the change curve of recovery rate on random knockouts.

In all of the four subgraphs (a) – (d), we see an overall tendency of declining recovery rate with the increase of knockout rate. This shows that usually the more reactions are removed from a model, the fewer of them can be recovered given the information of remaining reactions in the model. One exception is that in Figure 2(a), the recovery rate of predictor P_5 on $iJR904$ model stays rather steady as the knockout rate goes from to 0.1 to 0.8. This is due to the fact that P_5 is based on the frequency of a reaction in all draft models in SEED, while reactions in these draft models cover the majority of reactions in $iJR904$ model. Therefore, P_5 can recover the majority of the network even though a large proportion of $iJR904$ model is removed. In the cases of $iJR904$ model in Figure 2(a) and Figure 2(b), the difference between two lines is trivial. This observation shows that the predictors perform almost equally well for the random knockouts and subgraph-based knockouts in $iJR904$ model. However, in Figure 2(a) and Figure 2(b), both predictors recover much more reactions in the case of random knockouts than for subgraph-based knockouts on $iSB619$ model. This is due to the fact that $iSB619$ model is not only smaller but also sparser than the $iJR904$ model. There are 812 reactions in the reconstructed $iJR904$ model while there are 590 reactions in the reconstructed $iSB619$ model. Meanwhile, there are $8,826$ pathway segments with length smaller or equal to 6 in the reconstructed $iSB904$ model, while the number is $3,054$ for reconstructed $iSB619$ model. In summary, the results shown in Figure 2 demonstrates that the results from our computational predictors agree with the property of evidence and models.

5 Conclusion

In this paper, we study the problem of accelerating the process of constructing a genome-scale models by suggesting a set of reactions to fill network holes. A Bayesian approach is proposed to take into account of all information gained in databases and mine through large volume of data. A set of computational tools is built to extract biological and topological evidence from existing data, construct predictors using different pieces of evidence, and design an ensemble of predictors to integrate and unify individual predictors. By suggesting a collection of candidate hole-fillers computationally, it improves the model-building process by speeding up the process of finding hole-filling reactions. A series of experiments is preformed in order to evaluate the performance of the approach and computational tools.

These computational tools that support the improvement of large-scale metabolic models are shown to be able to recover a large proportion of removed reactions in the reconstructed $iJR904$ and $iSB619$ models. Moreover, a collection of experiments generate new and informative results that shed light on the properties of metabolic networks and various data and evidence. For example, high true positive rates and low false positive rates for the two predictors using evidence from gene clusters in SEED organisms show that gene clusters are helpful in searching for candidate hole-fillers.

By providing computational tools that support the improvement of draft metabolic models, we expect to generate thousands of genome-scale metabolic models in a high-throughput way. These tools can be integrated into our efforts of developing a scientific workflow [20] to eventually automate the construction of genome-scale metabolic models. These models can be analyzed as a system and insights can be obtained about properties of organisms such as the genotype-phenotype relationships. With the availability of thousands of biological models, it is then possible to perform comparative analysis of these models and a new generation of experimental hypotheses can be achieved.

References

1. The SEED. http://www.theseed.org/wiki/Main_Page.
2. KEGG: Kyoto Encyclopedia of Genes and Genomes. http://www.genome.jp/kegg/.
3. BiGG: A Biochemical Genetic and Genomic Database of Large Scale Metabolic Reconstructions. http://bigg.ucsd.edu/.
4. CellDesigner: http://www.systems-biology.org/cd/.
5. Weka: Data mining software in Java. http://www.cs.waikato.ac.nz/ ml/weka/.
6. Feist A.M., Herrgard M.J., Thiele I., Reed J.L., and Palsson B.O. Reconstruction of biochemical networks in microbial organisms. *Nat. Rev. Microbiol.*, 2008.
7. Palsson BO. *Systems Biology: Properties of reconstructed networks.* Cambridge University Press, 2006.
8. Reed JL and Palsson BO. Minireview thirteen years of building constraint-based in silico models of escherichia coli. *Journal of Bacteriology*, pages 2692–2699, 2003.
9. Reed JL, Vo TD, Schilling CH, and Palsson BO. An expanded genome-scale model of escherichia coli k-12 (ijr904 gsm/gpr). *Genome Biol.*, 4(9):R54, 2003.
10. Edwards JS and Palsson BO. Robustness analysis of the Escherichia coli metabolic network. *Biotechnology Prog.*, 16:927–939, 2000.
11. Peter Kharchenko, Lifeng Chen, et. al. Identifying metabolic enzymes with multiple types of association evidence. *BMC Bioinformatics*, 7(1):177, 2006.
12. Chen L and Vitkup D. Predicting genes for orphan metabolic activities using phylogenetic profiles. *Geno. Biol*, 7(R17), 2006.
13. DeJongh M, Formsma K, et. al. Toward the automated generation of genome-scale metabolic networks in the SEED. *BMC Bioinformatics.*, 8(139), 2007.
14. Green ML and Karp PD. A bayesian method for identifying missing enzymes in predicted metabolic pathway databases. *BMC Bioinformatics*, 5(76), 2004.
15. Kharchenko P., Vitkup D., and Church G.M. Filling gaps in a metabolic network using expression information. *Bioinformatics*, 20(Suppl 1):I178–I185, 2004.
16. Gil R., Silva F.J., Pereto J., and MoyaA. Determination of the core of a minimal bacterial gene set. *Microbiology and Molecular Biology Reviews*, 68(3):518537, 2004.
17. Overbeek R, Begley T, et. al. The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. *Nucleic Acids Res.*, 33(17):5691–5702, 2005.
18. Aziz RK, Bartels D, et. al. The rast server: rapid annotations using subsystems technology. *BMC Genomics.*, 9(75), 2008.
19. Becker SA and Palsson BO. Genome-scale reconstruction of the metabolic network in staphylococcus aureus n315: an initial draft to the two-dimensional annotation. *BMC Microbiol.*, 5(8), 2005.
20. Shi X. and Stevens R. SWARM: a scientific workflow for supporting bayesian approaches to improve metabolic models. *Proceedings of the 6th international workshop on challenges of large applications in distributed environments(CLADE)*, 2008.