

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



miniTUBA: a web-based dynamic Bayesian network analysis system and an application for host-pathogen interaction analysis

Yongqun He and Zuoshuang Xiang

*Unit for Laboratory Animal Medicine, Department of Microbiology and Immunology,
Center for Computational Medicine and Bioinformatics, and Comprehensive Cancer Center,
University of Michigan, Ann Arbor, MI 48109, USA*

1. Introduction

A Bayesian network (BN) is a representation of a joint probability distribution over a set of random variables (Friedman, 2004). A BN includes two components: (i) a directed acyclic graph (DAG) with vertices representing variables and edges indicating conditional dependence relations, and (ii) a set of conditional distributions for each variable, given its parents in the graph. A dynamic Bayesian network (DBN), an extension of BN, describes how variables influence each other over time. Mathematically, a DBN is a discrete time approximation of a stochastic differential equation or as a Markov chain model with possibly many states. DBN analysis has been considered as a powerful method to analyze and interpret heterogeneous, fluctuating time course data for many systems including biomedical data (Dean and Kanazawa, 1988; Friedman, et al., 2000; Korb and Nicholson, 2004). Compared to static Bayesian networks, DBN captures time varying parameters and predicts a time course of biological progression (*e.g.*, disease). DBNs also permit temporal cycles between variables allowing the user to interpret connections as temporal causation—a more clinically relevant definition of causation for many clinicians. A key advantage of DBNs over static Bayesian network analysis is that the relationships described in DBNs always have an unambiguous direction of causality.

DBNs have been used to analyze and interpret various data in different systems including clinical data (Li, et al., 2007; Neapolitan, 2003; Peelen, et al., 2010; Watt and Bui, 2008). Most medical/clinical inference data are dynamic dataset. However, investigators in the field largely depended upon multiple static comparisons to determine which clinical or experimental variables may represent potential targets for prediction or prevention of deleterious outcomes in patients. However, a clinical event (*e.g.*, sepsis) is typically a complex, heterogeneous, and dynamic process. Multiple factors may influence its outcome. This static comparison approach has led to numerous failed clinical trials which targeted single inflammatory mediators in patients without careful consideration of their clinical states in specific clinical courses (Remick, 2003). DBNs provide a means by which these

dynamic and often noisy datasets can be analyzed and interpreted to predict the impact of a complete network of clinical and experimental variables in an individual patient at any given time point. This then provides a clinician or researcher the ability to diagnose a disease state and intervene before overt clinical evidence is present.

DBNs have been applied to analyze gene expression data (Dojer, et al., 2006; Husmeier, 2003; Imoto, et al., 2006; Kim, et al., 2003; Ong, et al., 2002; Pe'er, et al., 2001; Rau, et al., 2010; Yu, et al., 2004; Zou and Conzen, 2005). DBNs have generated insights that could not be obtained from static Bayesian analysis. Unlike BNs which are acyclic, DBNs allow for cycles and more closely reflect the biological realities. In addition, DBNs can improve the ability to predict causal relationships based on the temporal nature of the data. For example, Ong *et al.* used DBNs to model regulatory pathways among 169 genes in *E. coli* to physiological changes that affect tryptophan metabolism (Ong, et al., 2002). Kim et al. applied DBNs to study a 45-gene subnetwork of the cell cycle system in *Saccharomyces cerevisiae*, a species of budding yeast (Kim, et al., 2003). More recently, Rau et al. introduced an iterative empirical Bayesian procedure with a Kalman filter that estimates the posterior distributions of DBN parameters (Rau, et al., 2010). This empirical method consumes considerably less computational time and was used to analyze human T-cell activation data with 58 genes over 10 time points (Rangel, et al., 2004). However, in general DBNs have not been widely used for gene expression data analysis. Its general usefulness in modeling reliable pathways and predicting testable hypotheses remain to be demonstrated (Xia, et al., 2004).

miniTUBA is a web-based dynamic Bayesian network analysis and Gibbs sampling prediction system (<http://www.miniTUBA.org>), with the goal of learning and simulating biomedical networks using temporal data from experimental and clinical investigations (Xiang, et al., 2007). The miniTUBA modelling system allows clinical and biomedical researchers to perform complex medical/clinical inference and prediction. This system is designed for easy data management, and the results are displayed in a way that is easily interpretable by a clinical or biomedical investigator. The miniTUBA implementation does not require a local installation or significant data manipulation. Using synthetic data and laboratory research data, our previous publication (Xiang, et al., 2007) demonstrates that miniTUBA accurately identifies regulatory network structures from temporal data.

This chapter will describe many updated features of the miniTUBA system, and provide general instructions and pitfalls involved in miniTUBA DBN modeling. The study of host-pathogen interactions is critical to understand microbial pathogenesis and host immune responses against pathogen infections. To our knowledge, dynamic Bayesian network analysis has not been applied to study host-pathogen interactions using microarray gene expression data. In this chapter, we will report how to apply miniTUBA DBN analysis to understand the immune networks in murine macrophages responded to different *Brucella* infections.

2. miniTUBA design and implementation

2.1 Overall miniTUBA system design

The miniTUBA software allows users to continuously update their data, fill out missing data, choose different analysis settings (e.g., data discretization, Markov lags and prior topology), perform DBN, and visualize results (Fig. 1). miniTUBA can also make temporal predictions using Gibbs sampling to suggest interventions based on an automated learning process pipeline using all data provided. Different graphic supports are provided (Fig. 1).

miniTUBA currently runs on two Dell Poweredge 2580 servers running the Redhat Linux operating system (Redhat Enterprise Linux ES 4) and Apache HTTP Server. Data are stored in miniTUBA using a MySQL database and the interface is constructed using a variety of scripts including PHP and Perl.

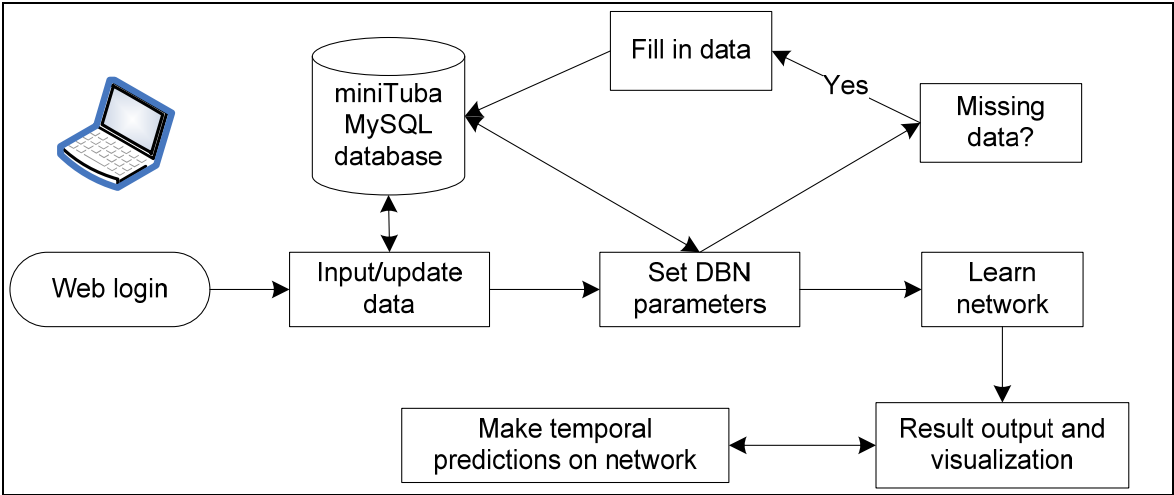


Fig. 1. miniTUBA system design.

The detailed tutorial for running miniTUBA is available at: <http://www.minituba.org/docs/tutorial.php>. We provide a miniTUBA Sandbox Demo (<http://www.minituba.org/sandbox/index.php>). This Sandbox Demo is developed for first time users to get familiar with the system by exploring the features using built-in user account and some simple data.

miniTUBA is a project-oriented web-based system. One or more projects can be created by a registered user. Currently, each project needs to go through an internal review process. This review process ensures that the computational resource is properly used since an approved project can run analyses that take up to 144 hrs – representing a significant computational investment. Once approved, a user can submit/update data, set up DBN settings and run each analysis. For each project, a user can run multiple analyses and these analyses will be stored in miniTUBA for later use.

By June 14 2010, miniTUBA has 66 registered users and hosted 82 projects including 32 testing projects.

2.2 miniTUBA DBN parameter selections

Optimal DBN parameter selections are critical to successful DBN analysis. Different DBN parameters can be set in miniTUBA to specify how the data are pre-processed and how constraints are set on the DBN learning algorithm. The DBN settings for each analysis will be stored in the miniTUBA database and can be reused for future analyses. Different DBN settings may result in different results. Default DBN settings follow the best practices described elsewhere (Yu, et al., 2004), but can be changed by the user if desired.

Depending on the purpose of a particular DBN simulation, in many cases only parts of experimental data and/or variables may be used (Fig. 2). A subset of the available experiment units has different meanings in different use cases. For microarray experiments, this may mean inclusion of selected microarray chips. In a clinical setting, it may mean analysis of only some of the patients. Certain variables can be excluded in a particular analysis. Those variables included in a study can be set as parents only (i.e., they do not act as children) or children only (i.e., they cannot be parents) (Fig. 2). For example, a drug treatment is usually considered as the start point of an experiment and hence can only have children, and a survival status is usually the final outcome of an event and hence cannot have any other child variable.

In many cases, not all required data points have experimental data. For example, most clinical data are not gathered at a consistent sampling interval. Therefore, a data fitting method is needed to fill in predicted data for these missing time points. Spline fitting is a general method of generating new data points within the range of a discrete set of known data points. Such a spline fitting approach has been shown by Yu et al to yield good behavior for reasonably smooth temporal data (Yu, et al., 2004). miniTUBA uses the R function `splinefun` (Forsythe GE, et al., 1977) to interpolate missing data across time.

Discretization is the process of transferring continuous variables into discrete counterparts. For efficient learning, the experimental data are discretized into a finite number of bins. Two basic discretization methods are equal interval method and equal number of variables (or quantile) method. The miniTUBA discretization allows 2 to 10 bins of interval discretization or 2 to 10 bins of quantile discretization. Alternatively, the users can make 2 to 10 customized bins. For example, an assignment of "2, 5" represents three bins with values <2 , $2-5$, and >5 , respectively.

One advantage of DBN analysis is that structural priors can be easily defined and enforced. In some cases, a user may know that some edges between variables must or must not be present. These constraints can be included in the analysis in miniTUBA as structural priors (Fig. 2).

For DBN execution, a miniTUBA user can specify the time of DBN execution ranging from 1 minute to 144 hours. To speed up the searching process, a DBN execution job can be performed in parallel by using 1-16 analysis instances, each to be run on a separate node of the backend cluster.

Since the identification of the highest-scoring Bayesian network model for a given set of data is known to be NP-complete (Chickering, 1996), heuristic rather than exhaustive search strategies are used. Two optimization algorithms are available in miniTUBA for users to

choose to learn the underlying DBN: simulated annealing and greedy learning. Simulated annealing is a learning process based on successive update steps (either random or deterministic). The update step length is proportional to an arbitrarily set parameter which can play the role of a temperature. In analogy with the annealing of metals, the temperature is made high in the early stages of the process for faster minimisation or learning, is then reduced for greater stability (Ispolatov and Maslov, 2008). The greedy random algorithm makes the locally optimal choice at each stage with the hope of finding the global optimum (Diniz-Filho, et al., 2005). Simulated annealing was found to consistently find the highest scoring Bayesian network models while greedy random algorithm does not (Hartemink, et al., 2002). Therefore, the simulated annealing approach is set as the default.

Current setting of DBN analysis parameters:

(1) Which experimental units [?] do you wish to include in the analysis? [Select all](#) [Deselect all](#)

1 ☒ 2 ☒ 3 ☒

(2) Inclusion and characteristics of experimental variables: [?]

Variables to include?	Allow parents or children? [?]	Spline Fitting [?]	Discretization Policy [?]
Select all Deselect all	Change all to Both	Change all to No Fitting	Change all to No Discretization
A <input checked="" type="checkbox"/>	Both	No Fitting	No Discretization
B <input checked="" type="checkbox"/>	Parents only	Natural Fitting	Quantile: 2 bins
C <input checked="" type="checkbox"/>	Children only	No Fitting	Interval: 3 bins
D <input checked="" type="checkbox"/>	Both	No Fitting	Customized: 3 bins 2, 5

(3) Network structure properties

Settings	Edges in the Setting	Option
Must Be Excluded Edges	Please list any relationships which should NOT be allowed in the proposed network. For example: Variable2->Variable1 Variable3->Variable1 Variable4->Variable1	<input type="radio"/> Use this option <input checked="" type="radio"/> Ignore this option
Must Be Present Edges	Please list any relationships which must be present in the proposed network. For example: Variable2->Variable3 Variable1->Variable4	<input type="radio"/> Use this option <input checked="" type="radio"/> Ignore this option
Initial Structure	Do you want to include the previous computed network structure in the current analysis? This will save the computation time.	<input type="radio"/> Use this option <input checked="" type="radio"/> Ignore this option

(4) Searcher Choice [?]

Simulated annealing

(5) Number of Instances [?]

2

(6) Markov Lags [?]

3

(7) Maximum Query Time [?]

0 Hours 10 Minutes

(8) Note

Fig. 2. An example setting for a miniTUBA DBN analysis. This example contains four nodes (A-D) and data of three experiments (i.e., three experimental units). Different nodes can have different settings.

Markov lag is the time interval (or lag) between the start of an event and its effect. For example, for a project with hourly data sets, Markov lag 1 implies that perturbations made now will have a measurable effect in one hour, while a Markov lag of 2 means that the effect will be observable after two hours. Depending on the nature of the experimental units and purpose of the experiment, a user may need to try different Markov lags to find out the optimal Markov lag. This time interval or lag can be easily changed in miniTUBA to examine influential relationships at different time intervals of interest. Although not used here, it is also possible to create models that cover a range of Markov lags. These more complete models are not included in miniTUBA as the results can be difficult to interpret mechanistically.

miniTUBA uses a modified version of the software package BANJO (<http://www.cs.duke.edu/~amink/software/banjo/>) developed under the direction of Dr. Alexander J. Hartemink at Duke University for dynamic Bayesian network learning for DBN learning (Smith, et al., 2006). DBN analysis does not require as much time as static BN requires due to its time restriction. However, generally BDN still requires much computational power. In miniTUBA, parallel computation is implemented for DBN execution. The learning jobs are distributed to a 44 node cluster of Apple G5 computers. In parallel jobs, each processor begins network learning from either a random DBN topology or uses a different random seed when learning with a stochastic method such as simulated annealing. Due to the embarrassingly parallel nature of network structure learning, this approach results in a nearly linear decrease in computing time as additional nodes are added. Because initial network learning can take hours to days to complete, miniTUBA alerts registered users by email when a job completes.

2.3 Temporal predictions based on Gibbs sampling in miniTUBA

It is possible to predict the values of future time points given a DBN, conditional probabilities generated from experimental data, and initial values. A prediction module was written that combines a Gibbs sampler to sample future values and a bootstrapping step to de-discretize the predictions. To perform the Gibbs sampling prediction, the data are first discretized (e.g. low, medium, high) and a conditional probability table was generated for each variable. The associated observations for each condition are also recorded. Gibbs sampling is then used to predict future states for each variable by sampling from the conditional probability distribution (Korb and Nicholson, 2004). Bootstrapping is used to de-discretize the states to continuous numerical values by sampling from the associated observations of the predicted states. In prediction mode, miniTUBA repeats this process of sampling and bootstrapping 10,000 times. For numerical variables, the mean and the standard error calculated from the 10,000 predictions are plotted along with the initial values. In miniTUBA, a probability table is provided for variables with nominal values and a probability curve is shown for every such variable.

This feature can suggest interventions based on an automated learning process pipeline using all data provided. This is very useful in a clinical setting. Based on previous data, a doctor may suggest some interventions to stop a disease trend. Similar cases may occur in laboratory research.

2.4 miniTUBA output and visualization

The top scoring and consensus networks generated by the DBN learning process are visualized using Graphviz (<http://www.graphviz.org/>) (Gansner and North, 2000). The top 10 scoring network graphs are shown in the results page. A consensus network among the top 10 scoring networks can be generated to show edges that are present in all 10 networks, indicating relationships that are present with high confidence. A simple edge confidence is calculated based on the frequency of the edge present among the top 10 scoring networks. While other metrics for edge confidence are possible, such as p-values and probability of conservation, we have found from user studies that these more quantitative metrics tend to overwhelm most non-computational users and end up making the result less useful.

Once a node in a miniTUBA top network is clicked, a conditional probability table calculated based on the input dataset is displayed, together with proposed causal relationships associated with the node. To assess how much better or worse a network is than the others among the top 10 scoring networks, a plot of the Bayes score distribution for these networks can also be displayed in the results page. To simply and intuitively interpret the relationships predicted by the DBN engine, a module is developed to allow user to generate 2D/3D scatter plots by clicking on a variable node with 1 or 2 other variable nodes as parents. The R “plot” command and the LiveGraphics3D package (<http://www.vis.uni-stuttgart.de/~kraus/LiveGraphics3D/>) are used to draw 2D and 3D plots respectively. The 3D scatter plot can be rotated or zoomed in/out for users to find better angle or resolution.

3. Application of miniTUBA in analysis of macrophage responses to *Brucella* infections

Brucella is a facultative intracellular Gram-negative bacterium that causes a zoonotic disease called brucellosis in swine, cattle, wild life, other animals and “undulant fever” in humans. (Schurig, et al., 2002). Human brucellosis remains the most common zoonotic disease worldwide with more than 500,000 new cases annually (Pappas, et al., 2006). *B. suis* cause brucellosis mainly in swine and humans (Pappas, et al., 2006). The interaction between macrophages and *B. suis* is critical for the establishment of a chronic *Brucella* infection. Smooth virulent *B. Suis*, which contains intact lipopolysaccharide (LPS), prevents macrophage cell death. However, rough and attenuated strain *B. suis* strain VTRS1, which is deficient in the O antigen of LPS due to a *wboA* gene mutation (Winter, et al., 1996), was able to induce strong programmed cell death of infected macrophages. To further investigate the mechanism of VTRS1-induced cytotoxicity in infected macrophages, microarrays were used to analyze temporal transcriptional responses of murine macrophage-like J774.A1 cell line following infection with strain 1330 or VTRS1.

In this section, we demonstrate the application of miniTUBA dynamic Bayesian network analysis in analyzing the immune network in macrophages infected with *Brucella suis* live virulent strain 1330 or attenuated vaccine candidate strain VTRS1 (Winter, et al., 1996). The results indicate that miniTUBA can be used to predict novel targets in a programmed cell death pathway.

3.1 Microarray experimental design

This study contained a DNA microarray experiment using the Affymetrix 430 2.0 array technology. In total, 42 microarray chips were included with seven time points. The basic protocol is as follows: J774.A1 mouse macrophages were plated in T75 at 8×10^6 cells per flask one day prior to infection, and then infected with *B. suis* S1330 or VTRS1 at a MOI of 200:1. Total RNAs were isolated by TRIzol and further purified using Qiagen RNeasy Mini Kit (Qiagen, Valencia, CA) at 0 h, 1 h, 2 h, 4 h, 8 h, 24 h, and 48 h post infection. The RNA samples were stored at -80 °C until an Agilent 2100 BioAnalyzer (Agilent Technologies, Palo Alto, CA) was used to assess the concentrations and quality of RNA samples. Total RNA (20 µg) per sample was used for hybridization with Affymetrix mouse GeneChip 430 2.0 array. Preparation of cDNA, hybridization, quality controls and scanning of the GeneChip 430 2.0 arrays were performed according to the manufacturer's protocol (Affymetrix, Santa Clara, CA) (He, et al., 2006).

The data has been deposited in the GEO database with the accession number GSE21117.

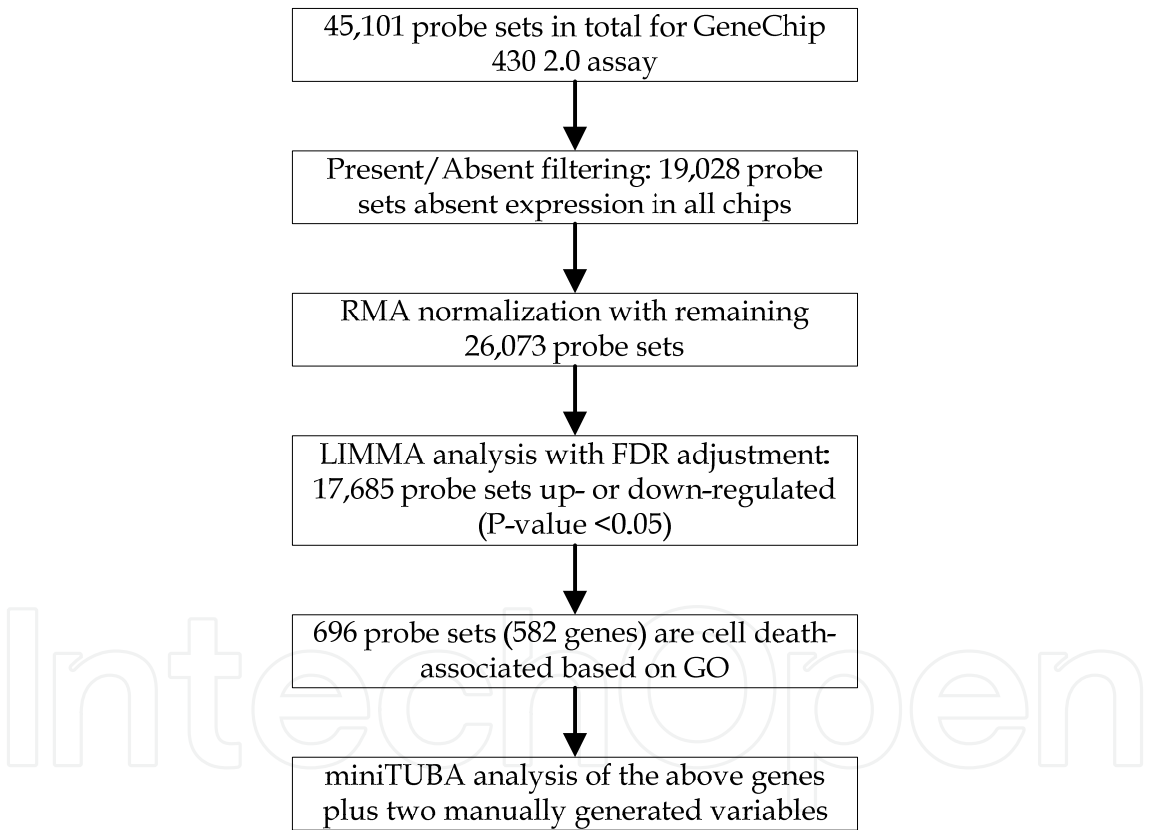


Fig. 3. Microarray data analysis workflow.

3.2 Data preprocessing prior to miniTUBA DNB analysis

For the probe sets that passed the Present/Absent filtering criterion, Robust Multi-array Average (RMA) normalization procedure was performed (Irizarry, et al., 2003). The log2 based gene expression values that have been background adjusted, normalized, and summarized were collected in the process. LIMMA (Linear Models for Microarray Data) with a false discovery rate (FDR) adjustment was further used to analyze up- or down-

regulated genes (Smyth, 2005). The moderated F-statistic obtained from LIMMA was used to determine any effects of strain, time, or possible interaction between strain and time. Cell death-associated genes were determined using the Gene Ontology (GO) annotation (Ashburner, et al., 2000). Those genes that are related to cell death and significantly regulated were used for Bayesian network analysis with an internally developed software program miniTUBA (Xiang, et al., 2007). The microarray data analysis workflow and relevant results are shown in Fig. 3.

3.3 miniTUBA DBN analysis of macrophage responses to *Brucella* infections

A miniTUBA DBN analysis was performed to analyze the immune network with a key purpose to determine why strain 1330 prevents programmed cell death of infected macrophages, while strain VTRS1 induced a cell death. The simulation used those genes that significantly regulated and belong to different cell death pathways (Fig. 3).

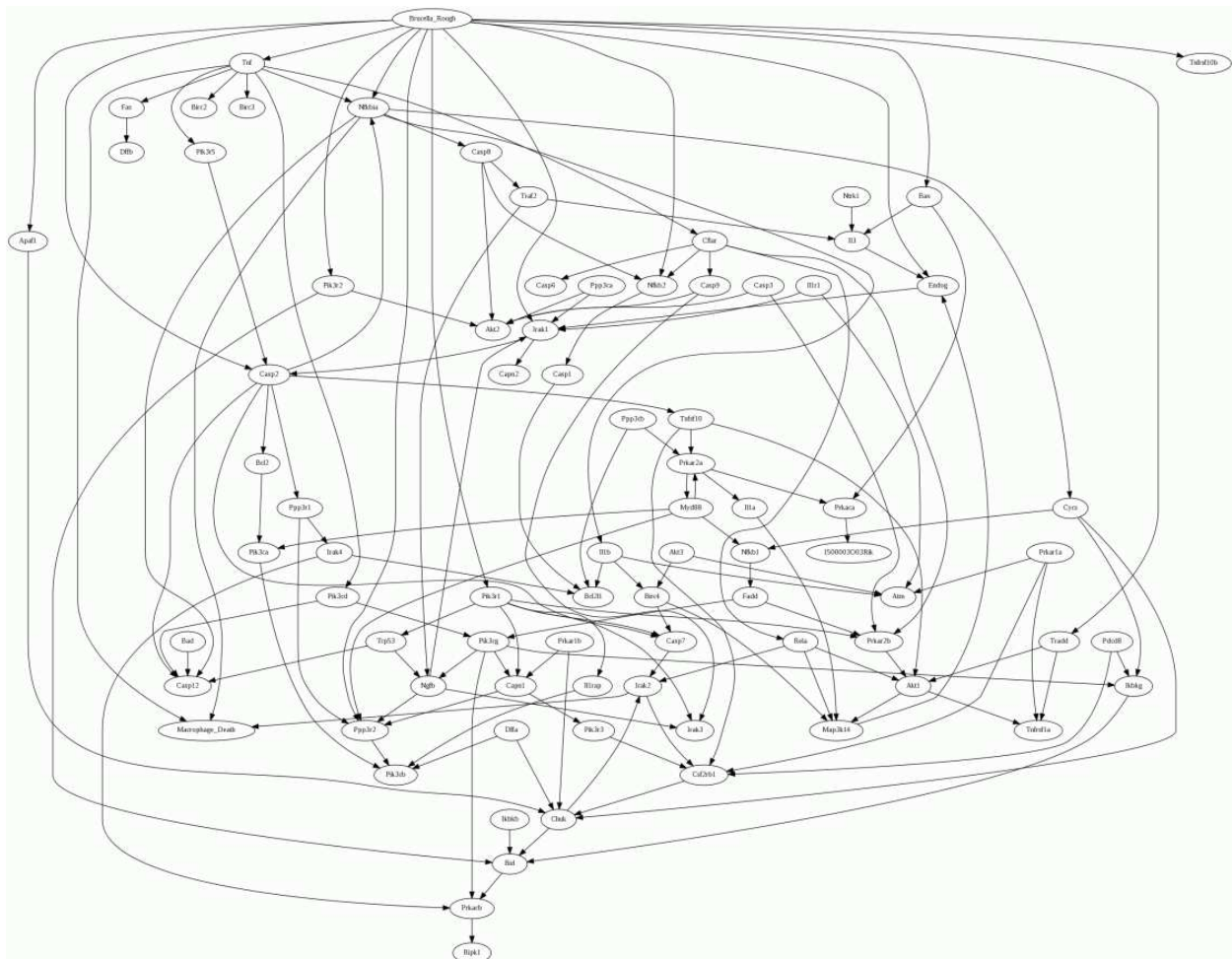


Fig. 4. Conserved network among top 10 DBN results. Four genes were screened out after the conservation. Ninety two edges were screened out for top 1 scoring network.

To perform miniTUBA analysis, the transcriptional data of those genes associated with cell death were extracted and formatted into miniTUBA input data format. The time points 0 h, 1 h, 2 h, 4 h, and 8 h post infection were used for the simulation. The Markov lag is set as 1

hour. Since the time points 3 h, 5 h, 6 h, and 7 h were missing, spline fitting was used to fill in the missing data. We did not use 24 h and 48 h since it would require filling in too many missing time points between 8 h and 24 h and between 24 h and 48 h. Using so many missing data would make the DBN analysis not reliable. The setting of quantile 3 bins was used for variable data discretization. It means that the gene expression values of each gene are seperated into three bins (low, medium, and high), and each bin contains one third of values. The two manually generated variables are “Brucella_Rough” (*i.e.*, rough or smooth *Brucella* strain) and “Macrophage_Death” (*i.e.*, live or dead macrophages). These two variables were used to represent the bacterium strains and the cell death phenotypes. The variable “Brucella_Rough” was set to have no parent, meaning no other variable pointing to this variable. The variable “Macrophage_Death” was set to have no child, meaning this variable cannot point to any other variable. In total, 16 instance runs were performed using 16 nodes. Each run took one hour. In total 16 hours of execution time was used.

After the miniTUBA DBN simulations, 10 top networks were saved. Fig. 4 demonstrates the conserved network among all top 10 networks. Those edges that are not conserved in all top 10 networks are removed. Those variables that do not have any connection to other variables are also removed from the conserved network.

Each node in a top network can be clicked to show more details about this node (Fig. 5-6). For each node, its conditional probability table is displayed in details with the edges between the node and the parent nodes. When the node has only one parent (Fig. 5) or two parents (Fig. 6), a 2D or 3D scatter plot is displayed, respectively. The data and result visualization provide users direct information for a specific variable and its relations with other associated variables.

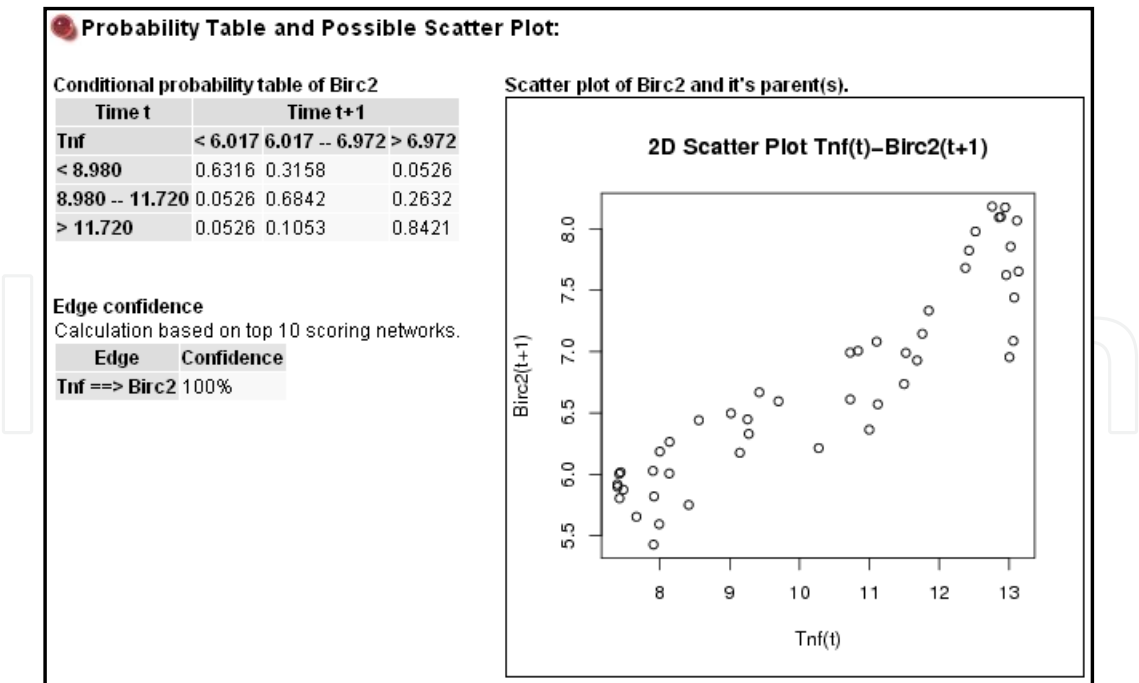


Fig. 5. The conditional probability table of Birc2 and the scatter plot for TNF and Birc2. The expression values of each gene was discretized into three bins with indicated ranges. The relation is based on the top 1 scoring network. A clear correlation between these two genes is observed. Based on miniTUBA DBN analysis, the edge Tnf -> Birc2 is identified.

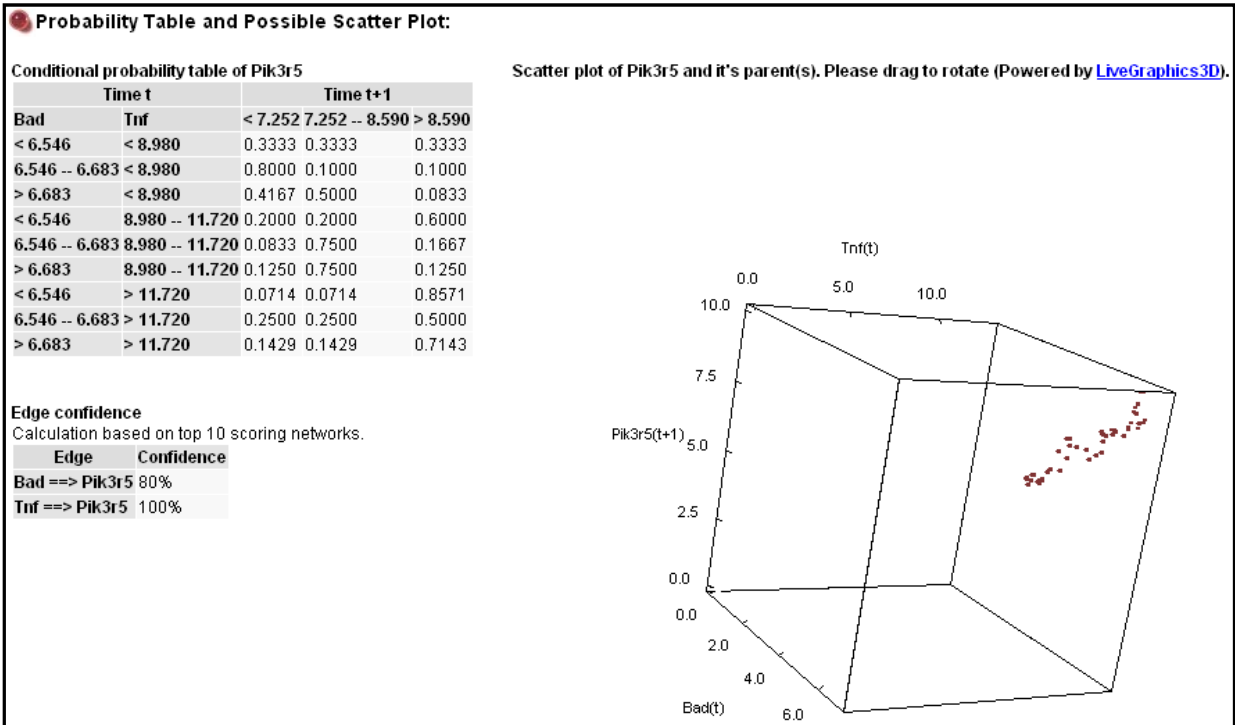


Fig. 6. The conditional probability table of Pik3r5 and the scatter plot for TNF, Bad, and Pik3r5. The expression values of each gene was discretized into three bins with indicated ranges. The relation is based on the top 1 scoring network. The plot was rotated to a position easily viewable.

3.4 Experimental verification of predicted miniTUBA results:

Our key question is the mechanism of why VTRS1 induced macrophage cell death while its virulent parent strain S1330 prevents the cell death of infected macrophages. To address this question, we have specifically analyzed the possible genes in the network that starts from “Brucea_Rough” and ends at “Macrophage_Death”. A portion of the predicted results is shown in Fig. 7. This small network demonstrates that rough *Brucella* induced high level of expressions of proinflammatory gene TNF and NF-κB pathway gene Nfkbia, both contribute to the death of infected macrophages. The caspase-1 (Casp1) appears not to play a role in the macropahge death (Fig. 7).

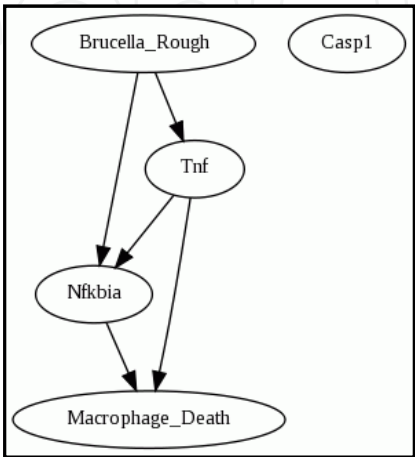


Fig. 7. Predicted network among 5 variables.

Our further experiments verified these predictions. A high level of proinflammatory response was induced by VTRS1 but not by its parent virulent strain S1330. The important roles of TNF- α and Nfkbia in VTRS1-induced macrophage death were further confirmed by individual inhibition studies (data not shown). While Casp1 plays an important role in Casp1-dependent proinflammatory pyroptosis (Bergsbaken, et al., 2009), an inhibition study using a Casp1 inhibitor (Z-WEHD-FMK) indicated that Casp1 did not play an obvious role in the VTRS1-induced macrophage cell death (data not shown). We previously found that rough attenuated vaccine strain RB51 and a *whoA* mutant RA1 induced Casp2-mediated cell death (Chen and He, 2009). In this study, miniTUBA also predicted a critical role of Casp2 in the rough attenuated *B. suis* strain VTRS1-induced macrophage cell death, which was later experimentally verified (data to be published).

These studies further demonstrate that the miniTUBA DBN analysis was able to predict important factors in a biological pathway using high throughput microarray gene expression data, which successfully guide the experimental evaluations.

3.5 Demonstration of Gibbs sampling prediction:

While the prediction of future events is not designed or important for this microarray study, we can use the same data sets to demonstrate how miniTUBA performs prediction and display predictive results (Fig. 8). In this demonstration, we showed that using the data from early time points, the results in time points 5 h and 6 h post infection could be predicted with an apparent success. More detailed verification of this method is described in the original miniTUBA publication (Xiang, et al., 2007).

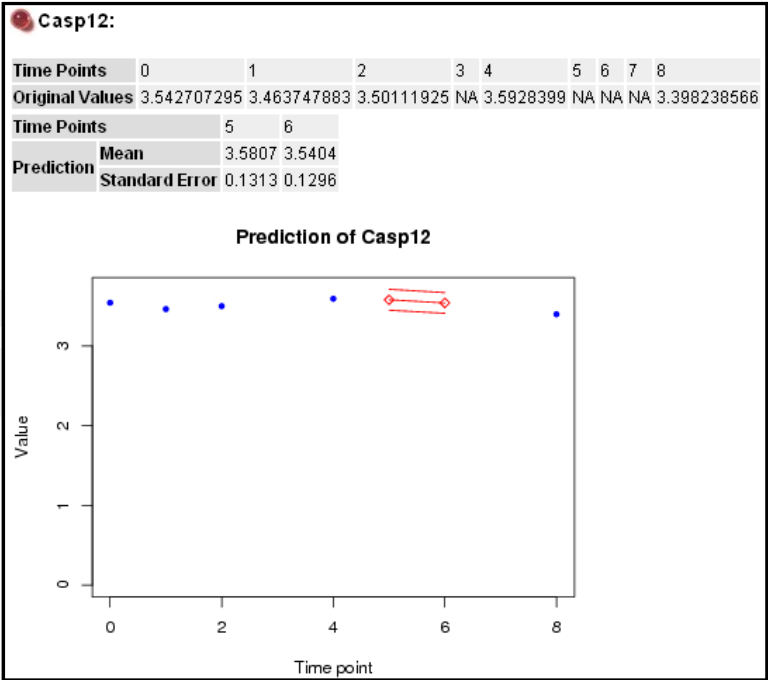


Fig. 8. Predicted results for Casp12 for the 5th and 6th time points. The round points without error ranges are the values from previously known time points and the diamond shaped points with error ranges are the predicted values and their associated standard errors. The 8th point was not used for prediction.

4. Conclusion and discussion

In this chapter, we have introduced in details the miniTUBA system, and how to apply the miniTUBA dynamic Bayesian network (DBN) approach to analyze a typical use case in the areas of host-pathogen interactions using high throughput microarray data.

The DBNs are powerful to model the stochastic evolution of a set of random variables over time. Since the biological processes and various measurement errors are stochastic in nature, DBN has been considered as a suitable technique to study biological networks and pathways. Bayesian networks (BNs) and DBNs are based on a multinomial distribution. This distribution is very flexible, and each node has a different parameterization. Therefore, it is very feasible to use DBNs to model the dynamics of biological systems and responses to parameter perturbations. However, although a few applications for both Bayesian network and DBNs to modeling gene expression data have been discussed and reported, their usefulness remains to be shown with more well-understood pathways (Xia, et al., 2004). Programmed cell death (*i.e.*, apoptosis) pathways are well studied and important for all plant and animal organisms. We first demonstrated in this report how the DBN analysis can be used to predict crucial genes for a cell death pathway, which led to correct experimental verification.

Two major challenges in DBN analysis for biological network modeling exist. First, continuous gene expression data has to be discretized, leading to the loss of information. The discretization simplifies the computation and stabilizes the predicated results. However, current equal quantile and interval discretization methods do not often reflect the biological realities. The customized discretization method is too time consuming and may not correlate with the unknown truth either. Therefore, alternative approaches will need to be explored to improve the discretization and minimize loss of information. How to find reliable ways to model continuous data remains to be a major challenge in the DBN and other modeling studies. Second, it is a big challenge to identify the correct time steps (*i.e.*, Markov lags) for a DBN modeling. By default, we require all variables have the same time step size. However, it might be possible to allow a mixture of different time step sizes. The time scale likely differ between variables. To identify the relevant time scale, we may allow different discretization schemes. While more finely discretized variables offer slower changes, it might be difficult to determine how many are appropriate. The generation of very large sizes of discretizations is also time consuming. One solution is to allow mixtures of time steps in the learning step. However, it is in practice very difficult because the current step depends on a range of past experiences. If the previous time steps are not multiplies of each other, a complex splining function is usually needed to dynamically interpolate the missing data. Alternatively, we can explicitly search for an optimum informative time step. A DBN search will favor small time steps because it means more data to be used. However, if the data represents only more interpolated data, it would not help. While DBN analysis can be improved in different directions, the two areas of DBN research with the largest impact are probably the discretization and correct time step setting.

Besides addressing the above challenges, dynamic Bayesian networks can further be improved through different directions: (i) those strong links (or edges) are conserved among top networks and can be detected by consensus analysis (Fig. 4). (ii) cross-species

comparison may further help to reveal the conserved core network across different species (Gholami and Fellenberg, 2010). (iii) it is possible to learn dynamic regulatory networks by incorporating multiple data types (e.g., functional classification, shared motif motifs, protein-DNA binding, protein-protein interaction) (Bernard and Hartemink, 2005). (iv) incorporation with additional quantitative measurements (Xia, et al., 2004). (v) integration of DBN gene expression data analysis with literature-based network discovery (Ozgur, et al, 2010). (vi) Dynamic Bayesian network expansion for identification of new pathway elements as shown in a similar approach with static Bayesian network (Hodges, et al, 2010). Finally, all these new directions will need to be integrated in a proper way for accurate reconstruction and prediction of biological and medical networks. Such a network analysis approach is likely applicable for study of other networks (e.g., social networks).

5. Acknowledgements

This work was sponsored by the NIH/NIAID Regional Center of Excellence for Bio-defense and Emerging Infectious Diseases Research (RCE) Program. The authors wish to acknowledge membership within and support from the Region V 'Great Lakes' RCE (NIH award 1-U54-AI-057153). This project was also supported by the NIAID R-21 grant (1R21AI057875) to Y. He.

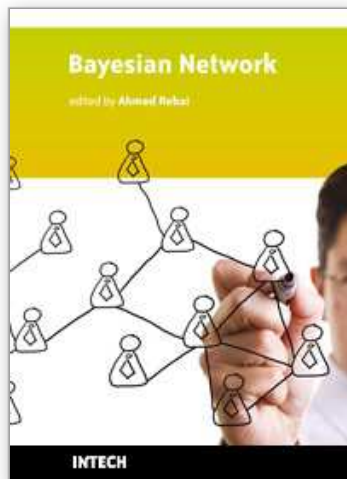
6. References

- Ashburner, M., *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, *Nat Genet*, **25**, 25-29.
- Bergsbaken, T., Fink, S.L. and Cookson, B.T. (2009) Pyroptosis: host cell death and inflammation, *Nature reviews*, **7**, 99-109.
- Bernard, A. and Hartemink, A.J. (2005) Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data, *Pac Symp Biocomput*, 459-470.
- Chen, F. and He, Y. (2009) Caspase-2 mediated apoptotic and necrotic murine macrophage cell death induced by rough *Brucella abortus*, *PLoS One*, **4**, e6830.
- Chickering, D.M. (1996) Learning Bayesian Networks is NP-Complete. In Fisher, D. and Lenz, H.J. (eds), *Learning from Data: Artificial Intelligence and Statistics V (Lecture Notes in Statistics)*. Springer-Verlag, pp. 121-130.
- Dean, T. and Kanazawa, K. (1988) *Probabilistic temporal reasoning*. Proceedings AAAI. AAAI Press / The MIT Press, St. Paul, MN, USA.
- Diniz-Filho, J.A., *et al.* (2005) Priority areas for anuran conservation using biogeographical data: a comparison of greedy, rarity, and simulated annealing algorithms to define reserve networks in cerrado, *Braz J Biol*, **65**, 251-261.
- Dojer, N., *et al.* (2006) Applying dynamic Bayesian networks to perturbed gene expression data, *BMC Bioinformatics*, **7**, 249.
- Forsythe GE, Malcolm MA and CB., M. (1977) *Computer Methods for Mathematical Computations*. Prentice Hall Upper Saddle River, New Jersey.
- Friedman, N. (2004) Inferring cellular networks using probabilistic graphical models, *Science*, **303**, 799-805.

- Friedman, N., *et al.* (2000) Using Bayesian networks to analyze expression data, *J Comput Biol*, **7**, 601-620.
- Gansner, E. and North, N.C. (2000) An open graph visualization system and its applications to software engineering, *Software-practice and experience*, **30**, 1203-1233.
- Gholami, A.M. and Fellenberg, K. (2010) Cross-species common regulatory network inference without requirement for prior gene affiliation, *Bioinformatics*, **26**, 1082-1090.
- Hartemink, A.J., *et al.* (2002) Combining location and expression data for principled discovery of genetic regulatory network models, *Pac Symp Biocomput*, 437-449.
- He, Y., *et al.* (2006) *Brucella melitensis* triggers time-dependent modulation of apoptosis and down-regulation of mitochondrion-associated gene expression in mouse macrophages, *Infect Immun*, **74**, 5035-5046.
- Hodges, A.P., *et al.* (2010) Bayesian network expansion identifies new ROS and biofilm regulators. *PLoS ONE*. **5**(3):e9513.
- Husmeier, D. (2003) Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks, *Bioinformatics*, **19**, 2271-2282.
- Imoto, S., *et al.* (2006) Computational strategy for discovering druggable gene networks from genome-wide RNA expression profiles, *Pac Symp Biocomput*, 559-571.
- Irizarry, R.A., *et al.* (2003) Exploration, normalization, and summaries of high density oligonucleotide array probe level data, *Biostatistics*, **4**, 249-264.
- Ispolatov, I. and Maslov, S. (2008) Detection of the dominant direction of information flow and feedback links in densely interconnected regulatory networks, *BMC Bioinformatics*, **9**, 424.
- Kim, S.Y., Imoto, S. and Miyano, S. (2003) Inferring gene networks from time series microarray data using dynamic Bayesian networks, *Brief Bioinform*, **4**, 228-235.
- Korb, K.B. and Nicholson, A.E. (2004) *Bayesian artificial intelligence*. Chapman & Hall/CRC Press, London, UK.
- Li, J., Wang, Z.J. and McKeown, M.J. (2007) A framework for group analysis of fMRI data using dynamic Bayesian networks, *Conf Proc IEEE Eng Med Biol Soc*, **2007**, 5992-5995.
- Neapolitan, R.E. (2003) *Learning Bayesian Networks*. Prentice Hall.
- Ong, I.M., Glasner, J.D. and Page, D. (2002) Modelling regulatory pathways in *E. coli* from time series expression profiles, *Bioinformatics*, **18 Suppl 1**, S241-248.
- Ozgur, A., *et al.* (2010) Literature-based discovery of IFN- γ and vaccine-mediated gene interaction networks. *Journal of Biomedicine and Biotechnology*. **Volume 2010**, Article ID 426479, 13 pages.
- Pappas, G., *et al.* (2006) The new global map of human brucellosis, *Lancet Infect Dis*, **6**, 91-99.
- Pe'er, D., *et al.* (2001) Inferring subnetworks from perturbed expression profiles, *Bioinformatics*, **17 Suppl 1**, S215-224.
- Peelen, L., *et al.* (2010) Using hierarchical dynamic Bayesian networks to investigate dynamics of organ failure in patients in the Intensive Care Unit, *J Biomed Inform*, **43**, 273-286.
- Rangel, C., *et al.* (2004) Modeling T-cell activation using gene expression profiling and state-space models, *Bioinformatics*, **20**, 1361-1372.

- Rau, A., *et al.* (2010) An empirical Bayesian method for estimating biological networks from temporal microarray data, *Stat Appl Genet Mol Biol*, **9**, Article 9.
- Remick, D.G. (2003) Cytokine therapeutics for the treatment of sepsis: why has nothing worked?, *Curr Pharm Des*, **9**, 75-82.
- Schurig, G.G., Sriranganathan, N. and Corbel, M.J. (2002) Brucellosis vaccines: past, present and future, *Vet Microbiol*, **90**, 479-496.
- Smith, V.A., *et al.* (2006) Computational inference of neural information flow networks, *PLoS computational biology*, **2**, e161.
- Smyth, G.K. (2005) limma: Linear Models for Microarray Data In, *Statistics for Biology and Health*. Springer, New York, pp. 397-420.
- Watt, E.W. and Bui, A.A. (2008) Evaluation of a dynamic bayesian belief network to predict osteoarthritic knee pain using data from the osteoarthritis initiative, *AMIA Annu Symp Proc*, 788-792.
- Winter, A.J., *et al.* (1996) Protection of BALB/c mice against homologous and heterologous species of *Brucella* by rough strain vaccines derived from *Brucella melitensis* and *Brucella suis* biovar 4, *Am J Vet Res*, **57**, 677-683.
- Xia, Y., *et al.* (2004) Analyzing cellular biochemistry in terms of molecular networks, *Annu Rev Biochem*, **73**, 1051-1087.
- Xiang, Z., *et al.* (2007) miniTUBA: medical inference by network integration of temporal data using Bayesian analysis, *Bioinformatics*, **23**, 2423-2432.
- Yu, J., *et al.* (2004) Advances to Bayesian network inference for generating causal networks from observational biological data, *Bioinformatics*, **20**, 3594-3603.
- Zou, M. and Conzen, S.D. (2005) A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data, *Bioinformatics*, **21**, 71-79.

IntechOpen



Bayesian Network

Edited by Ahmed Rebai

ISBN 978-953-307-124-4

Hard cover, 432 pages

Publisher Sciyo

Published online 18, August, 2010

Published in print edition August, 2010

Bayesian networks are a very general and powerful tool that can be used for a large number of problems involving uncertainty: reasoning, learning, planning and perception. They provide a language that supports efficient algorithms for the automatic construction of expert systems in several different contexts. The range of applications of Bayesian networks currently extends over almost all fields including engineering, biology and medicine, information and communication technologies and finance. This book is a collection of original contributions to the methodology and applications of Bayesian networks. It contains recent developments in the field and illustrates, on a sample of applications, the power of Bayesian networks in dealing the modeling of complex systems. Readers that are not familiar with this tool, but have some technical background, will find in this book all necessary theoretical and practical information on how to use and implement Bayesian networks in their own work. There is no doubt that this book constitutes a valuable resource for engineers, researchers, students and all those who are interested in discovering and experiencing the potential of this major tool of the century.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yongqun He and Zuoshuang Xiang (2010). MiniTUBA: a Web-Based Dynamic Bayesian Network Analysis System, Bayesian Network, Ahmed Rebai (Ed.), ISBN: 978-953-307-124-4, InTech, Available from: <http://www.intechopen.com/books/bayesian-network/miniTUBA-a-web-based-dynamic-bayesian-network-analysis-system->

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen