

Learning Genetic and Gene Bayesian Networks with Hidden Variables: Bilayer Verification algorithm

Jason E. Aten

Department of Biomathematics
David Geffen School of Medicine
University of California Los Angeles
AV-617 Center for the Health Sciences
Box 951766. Los Angeles, CA 90095-1766
{jaten}@ucla.edu

Abstract—To improve the recovery of gene-gene and marker-gene (eQTL) interaction networks from microarray and genetic data, we propose a new procedure for learning Bayesian networks. This algorithm, termed Bilayer Verification, starts with a user-specified leaf node, and then searches upstream to locate portions of the biological interaction network that can be verified as unconfounded by hidden variables such as protein levels. We provide theoretical justification for this procedure, which learns Bayesian networks by recursively finding two levels of v-structures in the data. We discuss the specialization and efficiencies gained when exogenous variables (those with no parents) such as genetic markers can be included in the network.

I. INTRODUCTION

Bayesian Network learning algorithms [1] [2] [3] [4] [5] [6] when applied to microarray data sets face a significant missing data problem which greatly confounds the recovery of accurate genetic networks. Actual protein levels, the active entity of most genes, are typically unknown. These unknowns may actively confound the correct derivation of a gene-gene interaction network based on microarray data.

In an important synthesis of classical and new approaches, Schadt and coworkers[7] brought the power of classical genetic Quantitative Trait Loci (QTL) analysis to bear on gene selection from modern microarray data, producing expression-Quantitative Trait Loci (eQTL). While successful at selecting genes for further experimental investigation, this work is limited to a comparison of a handful of three node models and does not learn gene-gene interactions.

As we examined the consequences of using genetic data to help inform the recovery of larger gene-gene interaction networks from microarray data, we encountered the issue of confounding of such interactions by unknown variables, in particular protein levels. More generally, this analysis applies to hidden variables in any Bayesian networks. In our intended application, the network describes the genetic marker-gene and gene-gene influence network within a tissue.

Our analysis leads us to a new verification process by which confounding influence can, in recognizable cases, be ruled out when the observations flow from a faithful probability distribution. Such edges in the Bayesian network are verified edges, and their discovery is based on two inter-connected layers of v-structures. Although common biological feedback mechanisms would initially suggest undirected or cyclic graphs, the restriction of learned networks to directed acyclic graphs (DAGs) is useful in that DAGs suggest priorities in terms of causal order and more readily suggest intervention points that may be useful in gene or siRNA therapy.

This paper is organized as follows. In section II, we review the terminology, definitions and theorems from the literature on learning Bayesian networks that illuminate the subsequent analysis. In section III, we provide a theorem for the process of learning Bayesian networks by recursively finding two levels of v-structures in the data, a process we term Bilayer Verification, and comment on its specialization for graphs with genetic markers. In section IV, we describe the algorithm that utilizes the developed

theorem to learn Bayesian networks with verified edges. Verified edges in a Bayesian network cannot be associations due only to hidden confounders. Throughout we comment on the special handling of genotype data within this algorithm.

II. DEFINITIONS AND THEORY CONTEXT

Here we provide a skeleton of the theory and definitions needed to place the subsequent theory in context.

Definition 1: Bayesian Network. A Bayesian network is a pair (G, θ) consisting of a directed acyclic graph G in which each node is a random variable, together with a set of parameters θ that encode for each node the conditional probability distribution of that node given its parents in the DAG.

Definition 2: d-separation. In a DAG, two disjoint sets of variables X and Y are d-separated by a third set D , itself disjoint from X and Y , if and only if along every path from an X node to a Y node we find that each node Z encountered satisfies one of the following three criteria:

- Z is at a converging connection, and neither Z nor any of its descendants (along directed paths) are in D
- Z is at a diverging connection, and $Z \in D$
- Z is at a serial connection, and $Z \in D$

The definition of d-separation is a condition on all paths in the network whose legs can be taken with or against the arrows on the directed edges. While traversing each path, three types of connection patterns may be encountered when going from a node in X to a node in Y . These are illustrated in Figure 1, and are called serial, diverging, and converging connections. They are also known as chain, fork, and collider connections. If two variables are not d-separated, they are d-connected.

Definition 3: d-separation notation and probabilistic interpretation. The d-separation criteria allows one to read complex induced conditional independence statements from a graph. It is also a common basis for the learning of Bayesian Networks. Suppose we specify three disjoint sets of nodes X , Y , and Z , where Z might be the empty set, but X and Y must have cardinality at least one. If Z d-separates X from Y , then the graph is claiming that in the probability distribution that it repre-

sents, X is conditionally independent of Y given Z . This is written symbolically as $X \perp\!\!\!\perp Y | Z$, indicating $P(X, Y | Z) = P(X | Z)P(Y | Z)$. The opposite, conditional dependence, is noted by $X \not\perp\!\!\!\perp Y | Z$. This notation is due to Dawid [8], while Pearl [6] uses this notation and further adopts 3-tuple relation notation from predicate calculus to denote conditional independence: $I(X, Z, Y)$, meaning that X is independent of Y given Z .

Definition 4: v-structure. If the two parent nodes of a collider are not directly connected by an arrow, this structure is termed an unshielded collider, or v-structure. Refer to graph c) of Figure 1.

Definition 5: faithful probability distribution. A probability distribution is faithful if each d-separation and d-dependence observed in the underlying causal graph is mirrored in the observed joint probability distribution.

A fundamental assumption in the structure learning algorithms such as the IC*[1], PC[5], and LCD [3] algorithms, is that the observed data follow from a faithful probability distribution. The interested reader is referred to [6] [5]. Other names for faithfulness [5] are stability[1] and DAG-isomorphism([6],p128). The primary difficulty being addressed by the faithfulness assumption is that the particular probability distribution being observed may have independencies that are due to an unusual coincidence of parameters rather than being a result of the structure of the edges in the graph. Peculiar parameter induced independencies will fool learning algorithms, but are expected to be uncommon.

The last essential component of the theory of Bayesian networks for interpreting the causal discovery algorithms is Verma and Pearl's [9] theorem on the equivalence of models under observation alone. Theorem 1.2.8 from ([1],p19) tells us that using observational data alone, the v-structures are what allow us to distinguish directionality in the undirected Markov Random Field or skeleton graph.

Theorem (Verma and Pearl 1990[9]) Observational Equivalence: Two DAGs are observationally equivalent if and only if they have the same skeletons and the same sets of v-structures, that is, two converging arrows whose tails are not connected by an arrow.

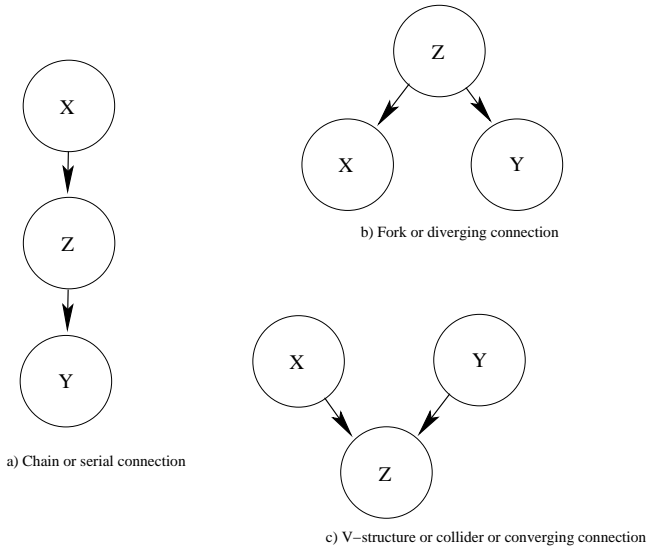


Fig. 1. The three patterns of connection in Bayesian networks

III. RECURSIVE V-STRUCTURE DERIVED BAYESIAN NETWORKS

Current algorithms for network recovery vary greatly in their computational complexity and the amount of information subsequently recovered. To contrast just two algorithms as examples, the PC algorithm[5] and the LCD algorithm[3] can handle very different sized data sets, given the time complexity of their execution. The PC algorithm will never finish on 100 variables, while the LCD will return a little structural information about many billions of variables, without necessarily tying any two micro observations together.

Our analysis in the following theorem leads to an algorithm, termed Bilayer verification, whose complexity falls in the middle. Our algorithm will capture more of the structure of the network than is learned by LCD, albeit less than that learned by PC, and yet will operated efficiently on large data mining problems such as those posed by microarray data sources.

The following analysis will allow us to both verify direct links and detect hidden confounding influences in the learned graph. Its name derives from the fact that by examining two layers of learned network at once, we can verify the edges and check for confounding at the earlier learned edge.

Theorem 1: Bilayer Verification correctness.

Let five observed random variables X_1, X_2, X_3, X_4, X_5 be drawn from a faithful probability distribution, so that independence between variables is entirely due to the underlying influence structure and not a rare cancelation of parameter values and observed frequencies. Suppose that the following pattern of dependence, characteristic of two interlocked v-structures (collider patterns), is observed.

First v-structure:

$$X_1 \not\perp\!\!\!\perp X_2 \quad (1)$$

$$X_1 \not\perp\!\!\!\perp X_3 \quad (2)$$

$$X_2 \perp\!\!\!\perp X_3 \quad (3)$$

$$X_2 \not\perp\!\!\!\perp X_3 \mid X_1 \quad (4)$$

Second v-structure:

$$X_2 \not\perp\!\!\!\perp X_4 \quad (5)$$

$$X_2 \not\perp\!\!\!\perp X_5 \quad (6)$$

$$X_4 \perp\!\!\!\perp X_5 \quad (7)$$

$$X_4 \not\perp\!\!\!\perp X_5 \mid X_2 \quad (8)$$

In the absence of confounding, such dependencies would be evidence for figure 2 graph C to be the underlying causal structure.[5][1] However even in the presence of confounding, the edge ** from X_2 to X_1 can be either verified, ambiguous (possibly confounded), or dismissed as surely confounded according to the following rules for edge deletion and verification:

- 1) if $(X_1 \not\perp\!\!\!\perp X_4 \mid X_2)$ and $(X_4 \perp\!\!\!\perp X_1)$, then delete the ** edge $X_2 \rightarrow X_1$ as it was only due to confounding.
- 2) if $(X_1 \perp\!\!\!\perp X_4 \mid X_2)$ and $(X_4 \not\perp\!\!\!\perp X_1)$, then mark ** edge $X_2 \rightarrow X_1$ as verified (not confounded).

Proof: Since $X_2 \not\perp\!\!\!\perp X_1$, there exists a marginally d-connected path between X_2 and X_1 . Similarly, $X_3 \not\perp\!\!\!\perp X_1$, shows that there exists a marginally d-connected path between X_3 and X_1 . Because $X_2 \perp\!\!\!\perp X_3$ and $X_2 \not\perp\!\!\!\perp X_3 \mid X_1$ are together characteristic only of a collider at X_1 , we know that all paths from X_2 to X_3 that account for the conditional dependence $X_2 \not\perp\!\!\!\perp X_3 \mid X_1$ must transit X_1 via two distinct directional edges both heading into X_1 from either side. Hence any final edge from X_2 to X_1

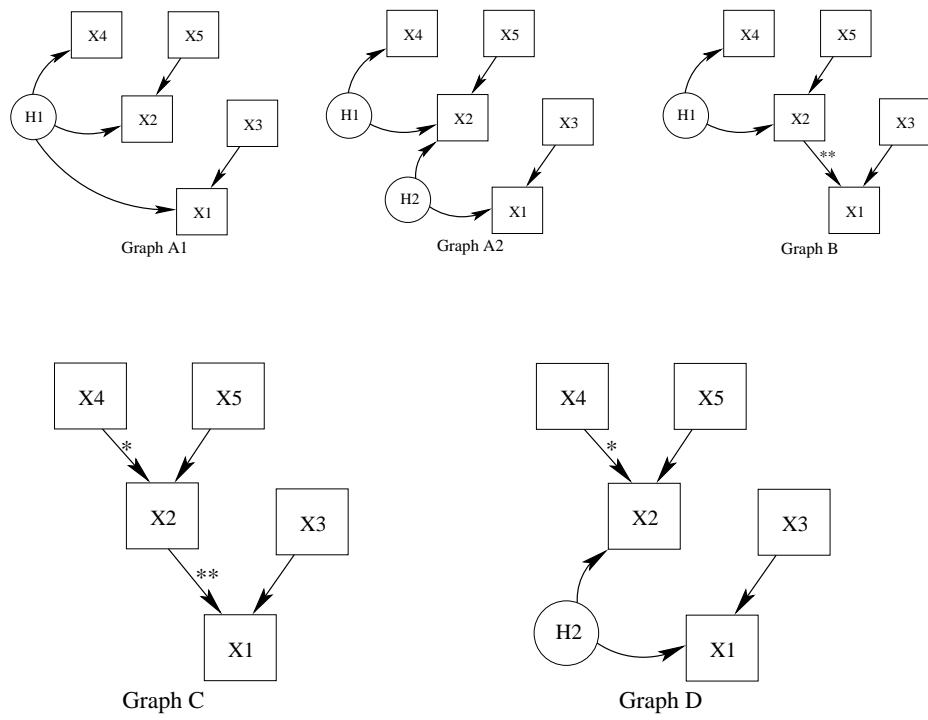


Fig. 2. Illustrating bilayer verification, graphs A1,A2,B,C,D

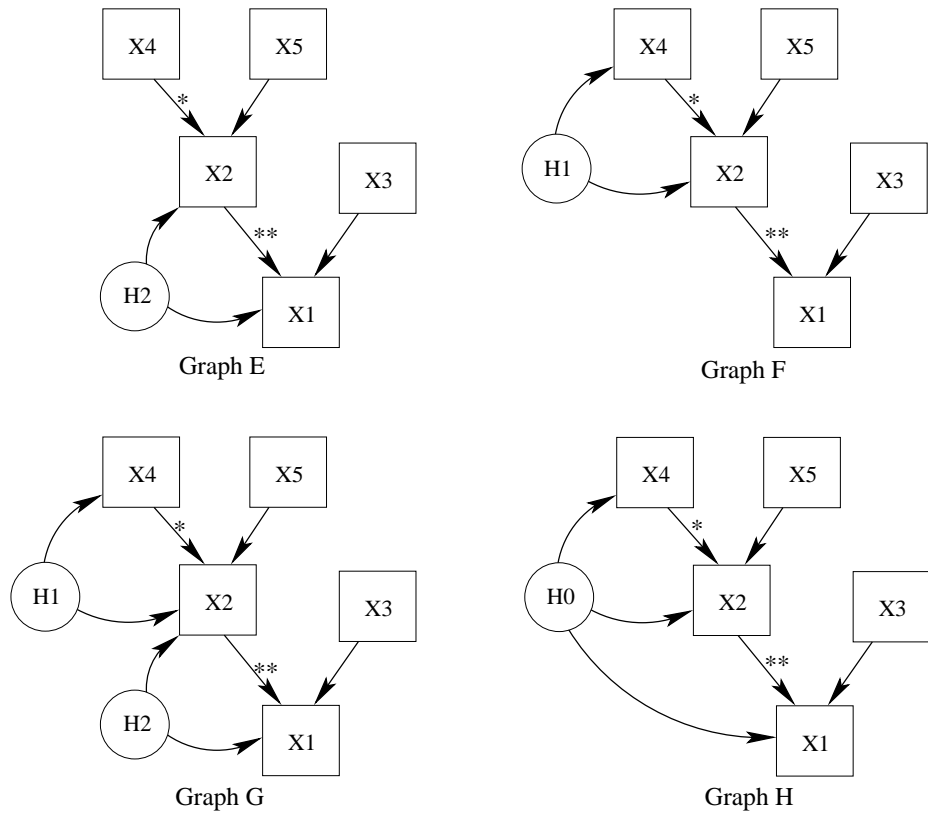


Fig. 3. Illustrating bilayer verification, graphs E,F,G,H

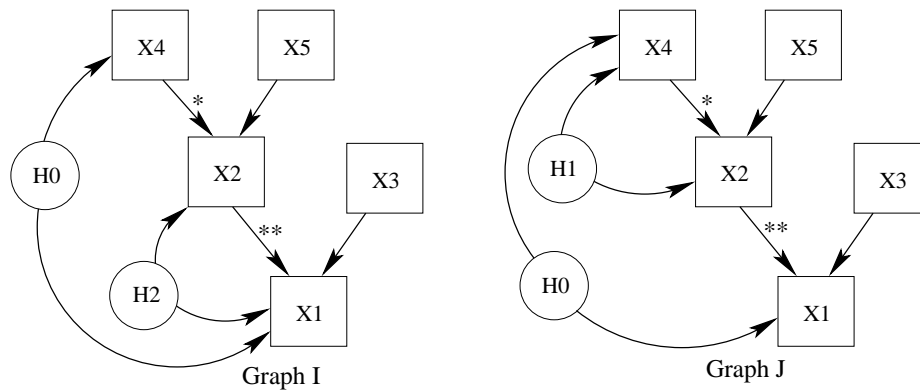


Fig. 4. Illustrating bilayer verification, graphs I,J

such as ** in the figures must be oriented with arrowhead into $X1$.

Having used $X3$ and the properties of d-separation at a v-structure to establish the directionality of the final edge ** in any paths from $X2$ to $X1$, it now suffices to analyze just the left side of the v-structure in terms of confounding. The right side paths (from $X3$ to $X1$) may have additional confounders, but they will not alter the conclusion about the directionality of the ** edge. An identical analysis holds for the second layer of v-structure utilized in equations (5)-(8), and hence confounding involving $X5$ and $X2$ is also safely ignored.

Now consider the following table (table I) in conjunction with Figures 2 and 3 and 4, which exhaust the possibilities for confounding of the 3 variables $X1$, $X2$, and $X4$. Circles represent latent, unobserved confounders. Squares represent observed variables. The table is marked yes if the independence statement holds, and — if it does not.

Graphs B, C, and F share the $X1 \perp\!\!\!\perp X4 | X2$ and $X1 \not\perp\!\!\!\perp X4$ property, and in each of these three graphs the ** edge $X2 \rightarrow X1$ is direct and unconfounded. Hence if this pattern of independence is observed we can confirm or mark the ** edge as verified (not confounded). Graphs A1 and D share the $X1 \not\perp\!\!\!\perp X4 | X2$ and $X1 \perp\!\!\!\perp X4$ pattern, and in both cases there is no true direct cause from $X2 \rightarrow X1$, instead just confounding exists. Graphs I and J share the same properties in the table as the other indeterminate cases A2, E, G, and H. Moreover these properties hold for all permutations of the presence and absence of the * and ** edges of graphs I and J; this conclusion follows from a

graph	$X1 \perp\!\!\!\perp X4 X2$	$X1 \perp\!\!\!\perp X4$
A1	—	yes
A2	—	—
B	yes	—
C	yes	—
D	—	yes
E	—	—
F	yes	—
G	—	—
H	—	—
I	—	—
J	—	—

TABLE I

ANALYSIS OF THE DEPICTED GRAPHS WITH RESPECT TO TWO SPECIFIC INDEPENDENCE PROPERTIES

simple enumeration (not shown) and checking of the four possibilities for the * and ** edge states (present/absent). ■

More broadly, note that $X1 \not\perp\!\!\!\perp X4$ reveals that there is some marginally d-connected path between $X1$ and $X4$, and $X1 \perp\!\!\!\perp X4 | X2$ reveals that $X2$ intercepts (is present on) all such paths. More specifically, on all such paths $X2$ is not the location of a collider, but rather is a chain or fork connection. A fork guarantees that the path departs $X2$ and arrives directionally at $X1$, and any chain must be oriented to flow from $X2$ towards $X1$ by the directionality argument for edge ** that began the proof. Hence we know that a directed path from $X2$ to $X1$ exists and can denote it by the ** edge.

Additionally, in the case when the grandparent ($X4$ above) is a genetic marker and hence is reasonably modeled as exogenous (does not have any input arrows from observed or confounding unobserved

variables), then we note that only (referring again to figures 2 - 4 and table I) graphs C , D , and E apply, and yet the exact same conclusion is reached. Indeed since this case is a subset of the above analysis, the conclusions of the theorem continue to hold for any parentless grandparent variables. However we gain from the additional knowledge of no arrows directed into $X1$. Graph C , D , and E now correspond in an invertible manner to unique patterns of independence. If we encounter the pattern associated with graph C or E we can immediately conclude that edge $**$ is verifiably present when $X1 \not\perp\!\!\!\perp X4$, even though there is also confounding present in graph E . This follows from examining graphs C , D , and E , which exhaust the possibilities when learning second layer relationships in which $X4$ has no inputs. Notice that once we conclude that $X1 \not\perp\!\!\!\perp X4$, we can immediately rule out graph D . In the remaining graphs (C and E) we are assured of the presence of the $**$ edge.

This reasoning results in the following corollary to the Bilayer verification.

Corollary 1: Parentless grandparent verifies parent-child relationship. If a variable M is known to have no parents (as with a genetic marker), then observing $M \not\perp\!\!\!\perp X2$ at a stage in learning recursive v-structures at which we've learned $X2 \rightarrow X1$ means that we can immediately mark $X2 \rightarrow X1$ as a verified edge, because the $**$ edge must be present even if the $X2 \rightarrow X1$ relationship is also confounded (as in graph E).

IV. BILAYER VERIFICATION ALGORITHM

Our analysis leads naturally to the formulation of a Bayesian Network learning algorithm that efficiently recovers a user-specified portion of the network even in the presence of confounding factors while exploiting the knowledge gained when genetic markers are present in the network.

Prior work and inspiration must be credited to Pearl's definitions of Potential and Genuine Cause and his IC* algorithm ([1]), Cooper's analysis [3] of the "+++" pattern, and the FTC/FBD algorithms [4]. Our approach differs from these in that it is oriented towards growing a network both with genetic-marker root nodes and furthermore by growing "from the leaf up." That is, our network is learned focusing on a given user-specified leaf or sink node,

and we actively ignore variables that appear to be unrelated to this outcome variable Z . We thus gain efficiency and a wide exploration of the data space at the price of potentially missing some causal factors.

Algorithm: Bilayer Verification algorithm

- 1) Choose a sink node Z .
- 2) Compute all pairwise marginal dependencies.
- 3) Considering the set U of neighbors of Z , pare down the set U by eliminating indirect relationships as follows. For distinct nodes $u_i \in U$ and $u_j \in U$, if $u_i \perp\!\!\!\perp Z | u_j$, then u_i is at best an indirect cause of Z , and so eliminate u_i from U for the purposes of v-structure detection. If u_i survives this test and u_i is a genetic marker (or a known exogenous node) then add the marker immediately as a parent of Z .
- 4) Now detect upstream causes by applying the unshielded collider (v-structure) test. For each distinct pair (X, Y) , where both X and Y are drawn from the chiseled down set U , test for both $X \not\perp\!\!\!\perp Y | Z$ and for $X \perp\!\!\!\perp Y$ marginally. If both tests hold true, then add both X and Y as parents of Z . If either X or Y has not yet been considered as a sink, add it to a queue Q of sinks to check.
- 5) Verification: once there are two or more layers in the network, verification becomes possible. If $X \rightarrow Z^1 \rightarrow Z^0$, then test for two conditional independencies: if $X \perp\!\!\!\perp Z^0 | Z^1$ and $X \not\perp\!\!\!\perp Z^0$ then mark $Z^1 \rightarrow Z^0$ as confirmed. If $X \not\perp\!\!\!\perp Z^0 | Z^1$ and $X \perp\!\!\!\perp Z^0$ then delete edge $Z^1 \rightarrow Z^0$ as it was due to confounding. (N.B. Otherwise the edge is ambiguous; it could be real, confounded, or have both a genuine direct and confounding factor.)
- 6) Root connections: if exogenous (root) variables such as genetic markers M or predicted genotypes at specified loci are in the graph, for each Y learned as a parent of Z , check for $Y \not\perp\!\!\!\perp M$ (QTL existence) and $M \perp\!\!\!\perp Z | Y$. If both these conditions hold, then add M as a parent of Y .
- 7) Repeat step 4 until all distinct pairs (X, Y) have been checked for v-structure.
- 8) Recursively select a new sink node Z from the queue Q and start again at step 3.

The root connections test in step 6 examines whether $M \rightarrow Y \rightarrow Z$. This conditional independence test can be accomplished for the gene expression microarray data as follows, given that Y and Z are continuous gene expression values, while M , being a genotype in our context, takes on either two (in the case of SNP genetic marker data) or three (for microsatellite genetic marker data) discrete values. We simply regress Z on Y and compute the residuals $\{E(Z_i|Y_i)\}$ for $i = 1..n$ after Y 's influence on Z has been taken into account.

V. SPECIAL HANDLING FOR GENOTYPES

Genotypes in an F2 mouse inter-cross commonly constructed for QTL analysis[10] are discrete valued variables which do not have parents in the graph, as their status is fixed from conception. Hence they are readily presumed to be upstream causal factors relative to all gene expression traits in the network. When the marker is at the beginning of a chain of marker M followed by two traits X , and Y , as in $M \rightarrow X \rightarrow Y$, then we check for correlation with an additive, dominant, and recessive model between the marker M and the conditioned trait $Y|X$. Since both X and Y are continuous, conditioning is easily accomplished by computing the residuals $Y - E(Y|X)$ after a linear regression of Y on X . Since in the graph we are checking $M \rightarrow X \rightarrow Y$, if we remove the influence of the direct parent X on the variation of Y , then we should see no further relationship between M and the residuals $Y|X$.

VI. CONCLUSION AND FUTURE WORK

Our analysis of confounding and our tuning of Bayesian Network learning theory for genetic data aims to inform the analysis of microarray data and the better construction of gene-gene interaction networks. Our next steps are to examine the performance characteristics of this theory when applied to the data of Schadt[7] and similar F2 mouse inter-crosses.

The author would like to thank and acknowledge support from grant HG02536-04 from the National Human Genome Research Institute (NHGRI) of the National Institutes of Health; and grant

DGE9987641 from the IGERT Bioinformatics program of the National Science Foundation.

REFERENCES

- [1] J. Pearl, *Causality: Models, Reasoning, and Inference*. Cambridge, UK: Cambridge University Press, 2000.
- [2] R. E. Neapolitan, *Learning Bayesian Networks*. Upper Saddle River, NJ 07458: Pearson Prentice Hall, 2004.
- [3] G. F. Cooper, "A Simple Constraint-Based Algorithm for Efficiently Mining Observational Databases for Causal Relationships," *Data Mining and Knowledge Discovery*, vol. 1, pp. 203–224, 1997.
- [4] P. R. Cohen, D. E. Gregory, L. A. Ballesteros, and R. S. Amant, "Two Algorithms for Inducing Structural Equation Models from Data," *In Preliminary Papers of the Fifth International Workshop on Artificial Intelligence and Statistics 1995*, pp. 129–139, 1995.
- [5] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, 2nd ed. Cambridge, Massachusetts: The MIT Press, 2000.
- [6] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, revised 2nd ed. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1988.
- [7] E. E. Schadt, J. Lamb, X. Yang, J. Zhu, S. Edwards, D. GuhaThakurta, S. K. Sieberts, S. Monks, M. Reitman, C. Zhang, P. Y. Lum, A. Leonardson, R. Thieringer, J. M. Metzger, L. Yang, J. Castle, H. Zhu, S. F. Kash, T. A. Drake, A. Sachs, and A. J. Lusis, "An integrative genomics approach to infer causal associations between gene expression and disease," *Nature Genetics*, vol. 37, no. 7, pp. 710–717, July 2005.
- [8] A. P. Dawid, "Conditional independence in statistical theory," *Journal of the Royal Statistical Society, Ser. B*, vol. 41, pp. 1–31, 1979.
- [9] T. Verma and J. Pearl, "Equivalence and synthesis of causal models." *In Proc. of the 4th Workshop on Uncertainty in Artificial Intelligence, July 1990.*, pp. 220–227, 1990, reprinted in P. Bonissone, M. Henrion, L.N. Kanal, and J.F. Lemmer (Eds.), *Uncertainty in Artificial Intelligence*, vol 6, pp255-68. Amsterdam: Elsevier.
- [10] E. E. Schadt, S. A. Monks, T. A. Drake, A. J. Lusis, N. Che, V. Colinayo, T. G. Ruff, S. B. Milligan, J. R. Lamb, G. Cavet, P. S. Linsley, M. Mao, R. B. Stoughton, and S. H. Friend, "Genetics of gene expression surveyed in maize, mouse and man," *Nature*, vol. 422, no. 6929, pp. 297–302, March 20 2003.