A graph-theoretical approach for pattern discovery in epidemiological research

R. A. Mushlin
A. Kershenbaum
S. T. Gallagher
T. R. Rebbeck

In this paper we describe a graph-theoretical approach for pattern discovery that is especially useful in epidemiological research when applied to case-control studies involving categorical features such as genotypes and exposures. Whereas existing approaches are limited to exploring relationships among two or three factors, or deal with thousands of genes but are unable to isolate interactions among individual genes, we focus on interactions among tens of genes. We present a pattern discovery algorithm that finds associations among multiple factors, such as genetic and environmental factors, and groups of individuals (cases and controls) in a clinical survey. To validate our approach and to demonstrate its effectiveness, we applied it to a selection of synthetic data sets that were devised to emulate the situations encountered in epidemiological studies involving common diseases with suspected associations involving multiple factors that could include inherited genotypes, somatic genotypes, demographic characteristics, or exposures. The results of this experiment show that it is possible to identify the effects of multiple factors in moderate-size surveys (involving hundreds of individuals) even when the number of factors is greater than three.

INTRODUCTION

One of the key promises of genomic medicine is the ability to predict susceptibility to complex diseases based on knowledge of inherited genotypes, somatic genetic changes, and environmental exposures. A great deal of effort has been invested in identifying the role of genes, exposures, lifestyles, and other factors in causing certain individuals to develop diseases or to exhibit poor prognoses when diagnosed. The problem is complicated by the fact that different combinations of genotypes and exposures can lead to the same disease, but may result in different levels of response to treatment or toxicity to drugs. Predicting disease risk and drug response has traditionally been the work of epidemiologists and pharmacologists. As genes have been found to play a major role in disease etiology and drug response, the fields of molecular epidemiology and

pharmacogenomics have assumed much of the burden of studying these effects in detail.

When multiple factors combine to determine a person's risk of disease or response to treatment, it is often difficult to sort out the contributions of these various factors, and to identify the combinations of these factors that are relevant to disease etiology, outcome, or drug response. Tools are needed that can efficiently search the high-dimensional feature space and discover patterns associated with a disease etiology. Standard statistical approaches have traditionally dealt only with interactions among two or three factors; new approaches are needed to deal with higher-order interactions.

In this paper we describe a pattern discovery and analysis method based on modeling the risk factors, the individuals, and the discovered patterns as graph constructs, without reference to any underlying functional (biological) model. The method itself consists of four phases:

1. Construct a graph that represents the risk factors associated with each individual.
2. Find patterns in the graph that correspond to groups of individuals with identical risk factors, and quantify the risk and significance for each pattern.
3. Construct a lattice that represents the relationships among the patterns.
4. Enumerate the interesting and significant risk factors and subpopulations.

Once the risk factor combinations and their affected populations have been identified, domain experts can compare these associations to the predictions derived from functional and etiological models, thereby strengthening or weakening the evidence for a particular model.

The complex interactions of multiple factors in disease etiology, outcome, or drug response are difficult to detect. Often the order of the interaction is high, and the main effects of each of these factors individually may be weak. A number of methods have been proposed to evaluate higher-order interactions among genes and other risk factors, including recursive partitioning,[1,2] random forests,[3] combinatorial partitioning,[4] permutation-based procedures,[5] multivariate feature selection,[6] multivariate adaptive regression splines,[7] boosting,[8] support vector machines,[9] neural networks,[10,11] Detection of Informative Combined Effects (DICE),[12] logistic regression,[13] penalized logistic regression,[14] Bayesian pathway modeling approaches,[15,16] Focused Interaction Testing Framework (FITF),[17] consensus algorithms,[18] and Classification and Regression Trees (CART). Another approach, multifactor-dimensionality reduction,[19] has been recently shown to be a special case of CART.[20] In particular, CART models have been widely applied and have the ability to detect complex interactions among multiple etiological factors. However, this method may assume an underlying model of association, may require assumptions about the identification of "purity" in the groupings identified, or may miss interactions that are not consistent with early splitting patterns. Our approach allows the detection of complex interactions among multiple etiological factors without making such assumptions.

The use of Bayesian graphical models to identify candidate genes in genome-wide association studies has recently been described.[21] Efficient algorithms for discovering association rules among features in very large databases have long been used commercially for market basket analysis,[22] but practical considerations limit the complexity of the discovered rules to a modest number of features. Our method, conversely, is aimed at a reduced set of already identified candidate gene polymorphisms. These polymorphisms may act in complex combinations to affect disease risk. Our method can handle this complexity and can shed light on the chemical pathway changes induced by combinations of polymorphisms.

The rest of the paper is organized as follows. We begin by describing our overall approach. We then give a detailed description of the implementation of our algorithm. Next we present computational evidence validating our approach. Finally, we summarize our contributions and suggest areas for future research.

OUR APPROACH

This section defines the basic concepts upon which our procedure is based. These include graph-theoretic concepts, epidemiological concepts, and set-theoretic concepts. We also describe the set-theoretic operations which form the basis for our algorithm.

Concepts and definitions

We use graphs to capture the relationship between a set of individuals and the allowed values of a specified set of features. The individuals and the feature values together make up the *nodes* of the graph. We distinguish between these two types of nodes by treating the feature values as *source* nodes (s-nodes), and the individuals as *terminal* nodes (t-nodes). We connect an s-node to a t-node with an *edge* if that feature value is exhibited by that individual. This results in a *bipartite* graph, a graph in which every node is one of two types (in our case, either an s-node or a t-node). Moreover, edges exist only between nodes of different types, never between nodes of the same type (***Figure 1***). Such a bipartite graph can be built to represent all or part of the data in the study. Hereafter, we refer to a bipartite graph as a graph, for simplicity.

A subgraph that consists of a set of nodes with edges between all pairs of nodes in the set is called a *clique*. Bipartite graphs cannot have (nontrivial) cliques because there can be no edges between any pair of nodes of the same type. There is, however, an analogous concept, called a *biclique*. A biclique is a subgraph defined by two sets of nodes where there is an edge between every node in the first set and every node in the second set. A *maximal biclique* is a biclique that is not contained in any larger biclique in the parent graph.

Figure 1 depicts a bipartite graph with t-nodes $\{p_i\}$ representing people and with s-nodes $\{f_i\}$ representing features. The node sets $\{p_4, p_6, p_7\}$, $\{f_3, f_6, f_8\}$ and all the edges between them define a maximal biclique within the larger graph. We are interested in maximal bicliques because in our application they represent the largest set of people who share a common set of features. When applied to genotype association studies, each feature f_i is one genotype (e.g., "AA") for one polymorphic locus (e.g., "GENE1"). Thus, a maximal biclique containing a set of specific genotypes for multiple loci would also contain all the individuals who share that exact combination of genotypes for those loci. For simplicity, in the remainder of this paper, we use "clique" to mean "biclique".

As the number of features increases, the number of people who share those features decreases. Each set of features generates a maximal clique. Maximal cliques and the relationships between them can be viewed as a *lattice*. A lattice consists of a set and a

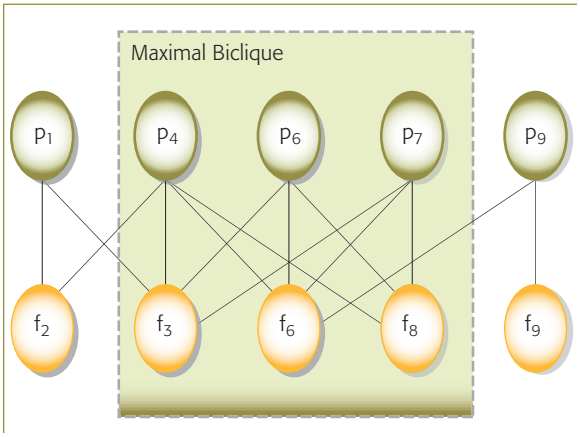

Figure 1
A bipartite graph and a maximal biclique for node types "people" (p) and "feature values" (f)

partial ordering (the "*less-than*" relationship, "<") such that for each pair of elements in the set, x and y, there are four possibilities: (1) $x < y$, (2) $y < x$, (3) x and y are equal, and (4) x and y are unrelated.

The "<" relation is transitive; that is, if x is $< y$ and y is $< z$, then x is $< z$. It is also anti-symmetric; that is, if x is strictly $< y$, then y cannot be strictly $< x$. In our case, we define a lattice on the cliques. In particular, the cliques are associated with sets of people and sets of features, and we define a notion of "<" in terms of subset relations on these sets. This will be described in detail in a later section.

In describing our method we make repeated use of constructs from both epidemiology and graph theory. *Cases* and *controls* are the two values of a binary classification variable used in an association study that define the dependent variable in the analysis. For example, cases can be those *affected* with a disease, those that have an adverse outcome in a longitudinal follow-up study, or those that have an adverse reaction to a drug in a pharmacogenetics study. Typically, a study is trying to determine if some *exposure* confers a risk of being a case. The exposure is the independent variable and can include inherited genotypes, somatic genotypes, chemical exposures, demographic characteristics, or any other risk factor of interest. In our application, we extend the notion of exposure to mean having a particular *set* of values for a specific group of features under study. Thus we shift from considering the individual features as independent risk factors to viewing a *pattern* of features as a single

Table 1 Representation of the 2×2 contingency table that forms the basis of our approach

	Cases	Controls	Row totals
Have pattern	***a***	***b***	N_{with}
Do not have pattern	***c***	***d***	$N_{without}$
Column totals	N_{cases}	$N_{controls}$	N_{total}

risk factor. This pattern may summarize information from a large number of independent variables. Here, we limit our discussion to binary independent and dependent covariates, but our approach can be extended to include polytomous variables (variables that take values from a discrete set) without loss of generality.

Risk measures

To quantify a pattern for the independent variables, we use a 2×2 table with meanings assigned to the rows and columns as shown in ***Table 1***. The values of a, b, c, and d are counts of individuals having the indicated pattern of exposure and affection (case/control) status. The value of a is referred to as the *support* for the pattern. Using this table framework, many metrics can be derived to make inferences about the relationship of the dependent and independent variables. For assessing risk in case-control studies, the *odds ratio* (OR) is commonly used:

$$OR = (a \cdot d)/(b \cdot c).$$

The odds ratio can range from 0 to ∞. For $OR > 1$, we infer that the pattern confers risk; for $OR < 1$, we infer that the pattern confers protection against affection. The null hypothesis yields $OR = 1$, which is interpreted as the pattern being unassociated with the dependent variable. To linearize and balance the risk measure around the null hypothesis, it is common to convert to a logarithmic scale. Here, we use $\log_{10}(OR)$ as the risk measure ($\log_{10}$ is convenient, but the natural logarithm is also commonly used), and other risk measures could be considered, such as positive likelihood ratio.

The probability p of obtaining a particular table a, b, c, d is given by:

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!\,b!\,c!\,d!\,(a+b+c+d)!}$$

where a, b, c and d occupy the cells in the 2×2 table indicated in Table 1. Note that, when the row and column totals (margins) are fixed, the only degree of freedom in this expression is one of the interior values such as a. When the odds ratio for a particular table having $a = a_0$, $OR(a_0)$, is > 1, the probability of obtaining a table with $OR \geq OR(a_0)$ by chance is the P-value for the table having $a = a_0$, and is given by:

$$\textit{P-value}(a_0) = \sum_{\substack{\textit{fixed margins based on } a_0 \\ a \geq a_0}} p(a)$$

where $p(a)$ is the probability for the observed table defined by $a \geq a_0$, and the sum is over all values of $a \geq a_0$ that keep the margins constant. A similar expression exists for $OR < 1$, and the sum is over $a \leq a_0$.

Consider an epidemiological case-control study represented as a case graph and a control graph. We first determine how many cases share the same values for a given set of features. This corresponds to the largest subgraph of the case graph in which all given s-nodes are fully connected to a set of t-nodes, and where neither the source nor terminal set can be enlarged without reducing the size of the other. Such a subgraph is an instance of a maximal clique. The source set of such a maximal clique is the pattern of independent variables (feature values), its terminal set is the support set, and the terminal set's cardinality is the value of a in the 2×2 table (Table 1). The cardinality of the terminal set from the maximal clique in the control graph having the same source set would determine the value of b (Table 1). Since N_{cases} and $N_{controls}$ are known and fixed, the 2×2 table for the pattern would be determined by these two counts. If the pattern of interest were known in advance, it would be a simple matter to search the case and control graphs for the desired clique. The challenge, however, is to evaluate the risk associated with *every* pattern, composed of every combination of features available in the study. This corresponds to exhaustively searching the graphs for *all* maximal cliques, and evaluating each one for risk and statistical significance. An exhaustive search is possible but may become intractable as the number of features and values per feature increases. Thus, we have implemented an algorithm that incorporates user-defined constraints to limit the complexity of the search, but is exhaustive within those bounds.

Set operations for maximal cliques

We construct a bipartite graph, $G = (S,T,E)$, where S and T are disjoint sets of nodes and E is a set of undirected edges, $e = (s,t)$, where s-node s is in S and t-node t is in T. Figure 1 illustrates such a graph for disjoint sets {p} and {f}. (The assignment of sets to S and T is arbitrary. Our method consistently assigns the feature values to S and the people to T.) Our goal is to find all maximal cliques $B = (S_B, T_B, E_B)$ of G, where S_B, T_B, and E_B are subsets of S, T, and E, respectively, and there is an edge $e = (s_B, t_B)$ for all pairs of nodes in S_B and T_B. A clique B is said to be maximal if there is no other clique $B' = (S'_B, T'_B, E'_B)$, where S_B is a (proper) subset of S'_B, or T_B is a subset of T'_B. The inner boxed portion of Figure 1 shows a maximal clique within a bipartite graph.

Our method operates on two types of candidates, which we refer to as s-cliques and t-cliques. For each s in S, we form an s-clique, $C(s) = [\{s\}, T(s)]$ where T(s) is the set of all t such that there is an edge (s, t) in E. Similarly, for each t in T we have a t-clique $C(t) = [S(t), \{t\}]$. All the candidate cliques we identify can be described as generalizations of this form. Specifically, given any set, S, of sources, we have an s-clique $C(S) = [S, T(S)]$ where T(S) is the set of t such that there exist edges (s, t) for all s in S. Similarly, we have t-cliques $C(T) = [S(T), T]$. The basic operation which is used to expand cliques is

$$C(S_1 \cup S_2) = [(S_1 \cup S_2), T(S_1) \cap T(S_2)] \text{ for s-cliques,}$$

and similarly,

$$C(T_1 \cup T_2) = [S(T_1) \cap S(T_2), (T_1 \cup T_2)] \text{ for t-cliques.}$$

We use s-cliques in discussing the algorithm further, but the same arguments can be applied to t-cliques.

The *expansion* operation used to identify maximal cliques is depicted in ***Figure 2A***. Clique $C(S_1)$ is expanded using $C(S_2)$. The resulting clique contains the union of the source sets and the intersection of the terminal sets. In expanding s-cliques, S_2 usually contains a single element, with one important exception. Given any source set, S_1, and its associated $T(S_1)$, we can identify the set $X(S_1)$ of sources which can extend S_1 without decreasing $T(S_1)$. We call this the *extension* set of S_1. $X(S_1)$ is defined by $X(S_1) = \{s|\ T(S_1) \subset T(X(S_1))\}$. This is equivalent to saying that $(T(S_1) \cap T(X(S_1)) = T(S_1)$. Thus, $T(S_1)$ is not decreased by adding s to S_1. The operation of adding the largest $X(S_1)$ to S_1 forms a

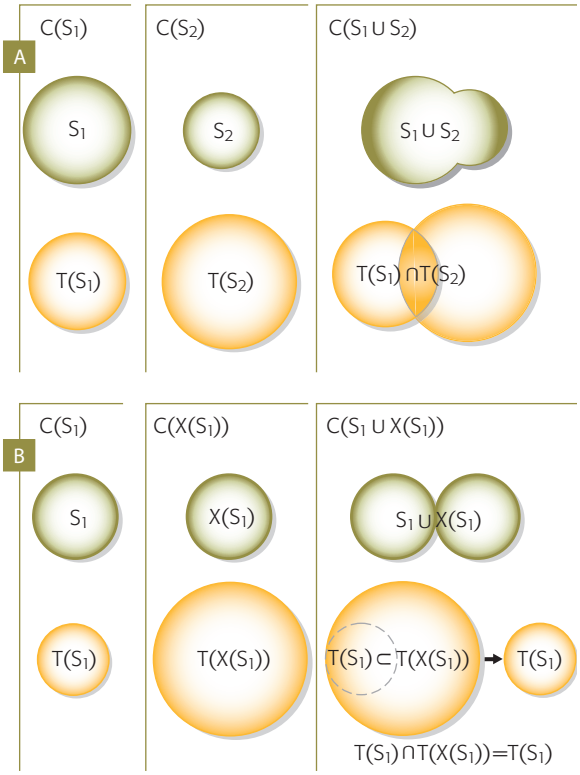

Figure 2
Clique $C(S_1)$ is (A)*expanded* by using $C(S_2)$; (B) *extended* by using $C(X(S_1))$

maximal clique. By definition, neither the extended S_1 nor $T(S_1)$ can be increased without decreasing the other. We thus define the operation of adding X(S) to S as taking the closure of S. This situation is illustrated in ***Figure 2B***. Clique $C(S_1)$ is *extended* using $C(X(S_1))$. The resulting clique contains the union of the disjoint source sets and the intersection of the completely overlapping terminal sets, which is identical to the original terminal set. This condition defines the extension $X(S_1)$. For convenience, we refer to single-element cliques together with their extensions as *singletons*. As an example of this, consider the node f_8 in Figure 1. It has the people p_4, p_6, and p_7 associated with it. This is a clique in the general sense, but it is not a maximal clique, the type of clique we wish to consider in our analysis. In particular, p_4, p_6, and p_7 are also associated with f_3 and f_6. Thus, once we choose to include f_8 in a clique, we could include f_3 and f_6 as well, without losing any people. We therefore do not consider f_8, p_4, p_6, and p_7 to define a singleton, but instead immediately add f_3 and f_6 to the clique. We have in effect acquired f_3 and f_6 "for free." The latter

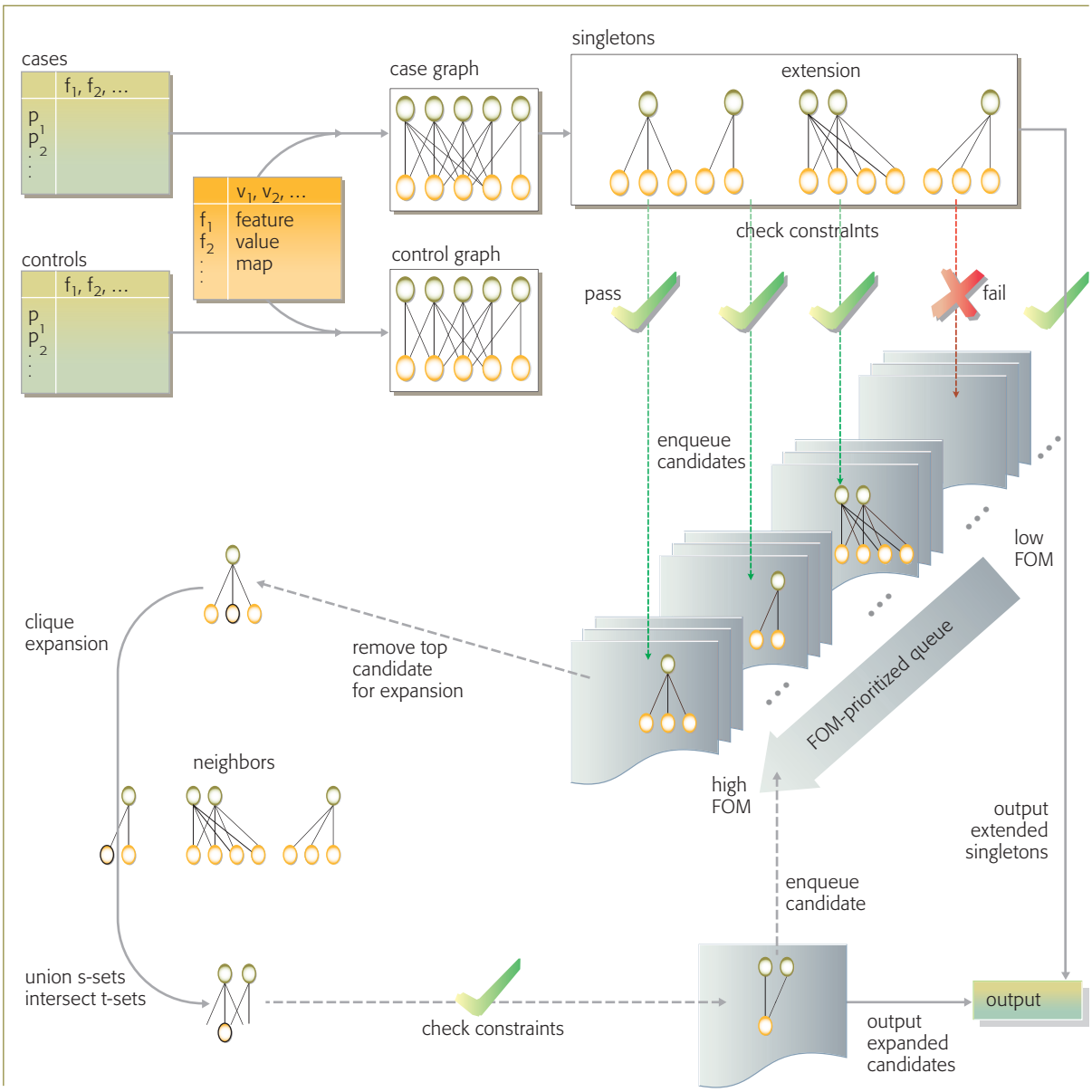

Figure 3
Program components and flow

clique is the desired maximal clique. The former clique is not.

IMPLEMENTATION

In this section, we give a schematic diagram of the algorithm and a description of each of its major components. We trace the flow of information and control from one component of the algorithm to another, describing figures of merit, constraints, how we deal with missing data and, finally, the output the algorithm produces.

Components and flow

A schematic diagram of the program components and flow is shown in ***Figure 3***. The program starts by building the bipartite graphs from the tables of raw data (typically, flat files). The raw case and control data, containing values for each of the features f_i for each individual p_i, are converted into bipartite graphs. An external mapping table is used to convert the raw data values into discrete, categorical feature values for use as s-nodes. Feature values not present in the raw data are mapped to a

categorical value reserved for missing data. The t-nodes are derived from the raw record identifiers.

The program then proceeds to search for maximal cliques in the case graph. The control graph is simply searched for s-nodes (feature value sets) that match those discovered in the case graph to obtain the counts for the 2×2 table. Clique discovery in the controls is thus avoided. The search is primed by finding all singleton cliques, including extensions, (maximal by construction, see above) by inspection. A copy of the singleton list is kept for use in clique expansion (see below).

Each clique is assigned a value for its figure of merit (FOM) and checked against user-specified constraints. Typical FOM choices include P-value and OR, but can include other measures derived from the 2×2 table. Typical filter constraints include minimum or maximum number of s-nodes and t-nodes, and minimum or maximum values of FOM. Cliques that meet the constraints are sent to an output file (they are acceptable), and placed in the candidate queue for expansion. The candidate queue is prioritized by FOM; that is, the clique with the best FOM is the first to be selected for expansion. A user-specified maximum queue size is imposed, based on available system memory. When the limit is exceeded, candidates at the bottom of the queue (with the worst FOM) are discarded. To improve efficiency, every candidate in the queue has an associated data structure in which the list of those singletons that have at least one t-node in common with its own t-nodes is maintained. We refer to this list as the *neighbor* set.

The program searches for new acceptable candidates by removing the candidate from the top of the queue for expansion. The selected clique is merged with each of the cliques in its neighbor set and extended if possible (see the section "Set operations for maximal cliques"). If the new candidate meets the external constraints and is not a duplicate of an existing clique, it is sent to the output file and inserted into the prioritized candidate queue on completion of the current expansion cycle. By first extending each newly formed candidate and then checking a hash table for duplicates, we avoid the quadratic process of having to compare every new clique to every existing one in order to determine maximality. The performance of the algorithm is linear in the product of the number of s-nodes, the number of t-nodes, and the number of cliques. After the queue has been augmented with all the acceptable new candidates, the new top element is removed for expansion, and the cycle is repeated until the queue is empty. The output file contains the maximal cliques that could be built from the cliques in the queue which met the constraints, along with the 2×2 table and statistics for each clique.

Figure of merit and constraints

With an infinite queue size and no constraints, the algorithm finds all acceptable patterns. For small problems this may be practical, but for problems where the number of possible patterns exceeds the queue size, some patterns will never be expanded, and it is possible that a complex pattern of interest may never be built because none of its precursors are still on the queue and available for expansion. In practice, this can be avoided by choosing a FOM that is expected to be high for immediate precursors of interesting patterns. This choice depends on the model system under study, but the algorithm does not presuppose any particular model.

The ability to externalize and tailor the queue-ordering function to the presumed shape of the landscape (shape of the FOM function in a multi-dimensional space) is actually a strength of the method that could be exploited in some situations. For our simulation we used the P-value as the basis for prioritizing the queue. Statistically, this is a "neutral" measure, in the sense that it measures the confidence in the result, not the strength of the result.

Although it is possible to leave a set of interacting features "stranded" on the pattern landscape, that would require that *all* paths from singletons to the pattern in question be discarded. For short patterns, we would have to discard only a few, even shorter, patterns. We handle these shorter patterns at the start of the process when the queue is relatively empty; therefore, we are unlikely to discard them for lack of space. For long patterns, the number of ways the pattern could be built up grows factorially; therefore, it is unlikely that we would discard *all* the paths. (For 10 genes there are a maximum of about 1 million patterns, which would require about 1 gigabyte of memory. In a real study, the number of actual patterns with any reasonable support is much

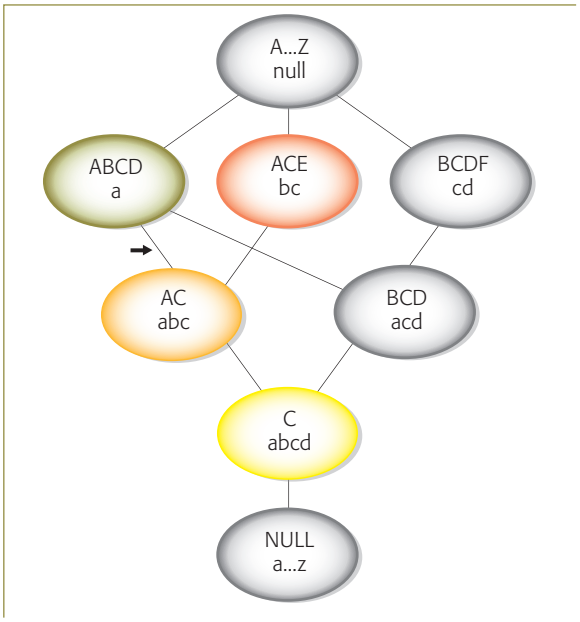

Figure 4
Example of a lattice composed of maximal bicliques

less than the maximum; thus, much less storage is required in practice.)

Unlike the FOM, which is used to prioritize cliques for expansion, constraints are used to directly filter the patterns that are put into the queue and reported as output. This prevents wasting queue space and interpretation effort on patterns that the user decides in advance would not be of interest. These constraints apply even when the queue is not full. We typically apply constraints to s-node and t-node counts, odds ratio, and P-value.

Final candidates and the lattice of cliques

When the algorithm halts, the output contains all the maximal cliques that could be built from the cliques in the queue that met the constraints. We call these the final candidates. Each final candidate is reported with its s-set (features), t-set (individuals), 2×2 table with any accompanying statistics, and FOM. The goal is to decide which patterns have the feature set that best predicts whether an individual is a case.

A few caveats are in order. First, the features that *predict* a disease or a drug response do not necessarily *cause* the disease or the response. Second, the results obtained from the sample of the population making up the data set may not be valid for the population as a whole, or for all segments of the population.

When we speak of a pattern's feature set, and the number of individuals who do and do not exhibit the pattern, we must be clear about what we are counting. Consider a data set with binary features A, B, and C. Every individual is either "1" or "0" for each of the features, for a total of $2^3 = 8$ possible unique records. But there are $3^3 = 27$ possible patterns. That is because the feature set for a pattern is the set of s-nodes that directly participates in the maximal clique, and this set implies "any value" for all features not explicitly mentioned in the set. For example, the set {A1, B1} specifically excludes A0, B0, but implicitly includes either C0 or C1. This can be written as {A1, B1, C*}, and has a t-set containing individuals with A = 1, B = 1, and C = any. Thus, each feature has three possible values: "1", "0", and "*"; hence, $3^3 = 27$ possible patterns.

The preceding example leads to an inherent ordering of patterns. Given two maximal cliques, C1(S1,T1) and C2(S2,T2), S1 $\subset$ S2 if-and-only-if T1 $\supset$ T2, and S1 $\supset$ S2 if-and-only-if T1 $\subset$ T2. This pair of properties allows us to construct a lattice of patterns from the algorithm output. Recall from the earlier section "Concepts and definitions," that a lattice is a collection of objects (cliques, in this case) and a partial ordering. The cliques themselves become nodes in the lattice graph, and edges exist between nodes corresponding to pairs of cliques for which the partial ordering relation (in this case, the subset relation) holds.

Sometimes not every node in the lattice is represented because the pattern list may be incomplete due to queue limitations or applied constraints, or both. In addition, the lattice may be filtered to eliminate uninteresting or statistically nonsignificant patterns. The resulting structure is a filtered lattice which is a collection of subgraphs of the full lattice. We refer to this structure simply as the lattice. A simple example of a lattice whose elements correspond to maximal cliques is shown in ***Figure 4***. Feature sets (labeled in uppercase) become more generalized moving down and more specialized moving up. Support (labeled in lowercase) becomes broader moving down and narrower moving up. Connected nodes satisfy the partial ordering requirements for a lattice. For example, we could say

that the clique [BCDF;cd] is "less than" [BCD;acd], if we use the subset relation between feature sets as the "less-than" relation (BCDF $\subset$ BCD*)

The four highlighted nodes illustrate the effect of progressively specifying features. With only feature C fixed (yellow), four individuals are found with the pattern. When feature A is incrementally added (orange), one individual (d) is lost. Further specifying feature E (red) loses one individual (a). If B and D are specified (green), however, two are lost (b,c).

Notice that some of the intermediate nodes have been filtered out, in which case edges are simply inserted to bypass the "ghost" nodes. The arrow in Figure 4 points to the location where patterns with features ABC and ACD would have been. Adding either B or D singly to AC may lead to ghost nodes for several reasons. They could have been intentionally filtered as uninteresting or nonsignificant, in which case their effect *must* have been for each of them to have removed one individual (b or c) apiece from the [AC;abc] pattern. Otherwise, either B or D removed both b and c, while the other had no effect. If adding B had removed both b and c, then the resulting pattern [ABC;a] would not have been a maximal pattern, since [ABCD;a] has the same support. If adding D had had no effect, then AC would not have been maximal, and [ACD;abc] would have replaced [AC;abc] in the lattice.

If the lattice is augmented with the s-set, t-set, and risk statistics, it can be a powerful aid in reasoning about the relationships among patterns. The algorithm expands cliques in order to search for combinations of feature values that together confer risk, but that individually, or in subsets, may not. Application of parsimony concepts is deferred until after all the cliques are discovered. At that point, one can use the lattice to trade off features for support; that is, a more general (smaller) description of the feature set covers more people, but often at the expense of conferred risk (odds ratio). The algorithm described here does not attempt to optimize such a trade-off.

Missing data, correlated features, and quantitative traits

A final consideration for the practical application of this method is how missing data are handled. Options include imputing missing data by statistical inference or omitting entire records containing any missing data, but neither are optimal solutions; whereas data imputation is effectively used in family-based studies where Mendelian or similar models can be used to predict unknown genotypes, this approach may be prone to misclassification in case-control or cohort studies of unrelated individuals. Similarly, omission of entire subjects limits the power for analysis of other variables. Statistical inference requires that the features be either uncorrelated or the correlations known, but it is those very correlations that we are trying to detect. Omitting records is wasteful of data that are often hard to collect, and even if the data were plentiful, the missing data may not be randomly distributed among records, thus introducing bias.

We propose to make use of the available data wherever possible. We exclude from all t-sets any individual that does not have *one* of the allowed values for every feature in the s-set. In other words, a missing feature value is never a match to any feature value. But a missing value for a feature not in the s-set does not in and of itself exclude an individual from t-set membership. This rule is most noticeable when counting the number of individuals that *do not* have a particular pattern of feature values. To be counted, they must have *some* value for every feature in the pattern, and at least one of the values must be different from all members of the pattern feature set. As a result of this treatment, N_{cases} and $N_{controls}$ are not constant across all patterns. When the pattern-to-pattern margin fluctuations are large, the P-values for patterns with identical OR can vary noticeably.

Single nucleotide polymorphisms (SNPs) on the same chromosome tend to be correlated to a degree more or less proportional to their proximity, a phenomenon called linkage disequilibrium (LD). LD is especially strong for SNPs in the same gene. When LD is present in the SNPs being studied, if either SNP is found to be a member of a clique, the other SNP will also tend to be a member of the same clique. If the LD is known in advance, as could be assumed for SNPs on the same gene, then one might collapse them into a single variable without significantly affecting the predictive power of the clique. However, in drawing inferences from the clique SNPs about their possible effects on pathway kinetics and disease processes, one must keep in mind that even if two SNPs are correlated, they

Table 2 The 2 × 2 contingency table for one of the data sets used in the validation runs

G1 = AA and G2 = Aa	Cases	Controls	Row totals
Match	***29***	***117***	146
Do not match	***85***	***769***	854
Column totals	114	886	1000

OR = 2.24, P = 0.000829, FOM = 3.08

both, individually or together, can still contribute to the pathology.

Quantitative variables, as might be used to study gene-environment interactions, could be included alongside SNP and other categorical variables by binning (placing in bins). Such analyses are not presented here, but the binning operation is conveniently performed during the feature-value mapping operation as shown in Figure 3.

VALIDATION

To demonstrate the utility and validity of our method, we applied it to a selection of synthetic data sets. The data sets were devised to emulate the situations encountered in epidemiology studies involving common diseases having suspected associations with multiple factors that could include inherited genotypes, somatic genotypes, demographic characteristics, or exposures. For example, the features may consist of data on SNPs, and the dependent variable may be a particular type of cancer.

Data-set construction

For this application, we undertook a simple example that considered the potential effect of nine genes as independent variables and a binary disease-dependent variable, case-control status. At each gene, we specify a SNP of interest with two alleles. We designate the common, or major allele A and the rare, or minor allele a. Because somatic cells (cells other than egg and sperm) have paired (diploid) chromosomes with one allele from each parent for each SNP, three possible allele combinations, or genotypes, arise: AA, Aa, and aa. If the frequency of A is p, then the frequency of a is $1-p$. Under ordinary circumstances, these two alleles combine in binomial proportions (i.e., Hardy-Weinberg equilibrium, or HWE, for short) to form three genotypes, such that

$$freq(AA) = p^2$$
$$freq(aa) = q^2$$
$$freq(Aa) = 2pq$$

The minor allele frequencies were supplied as parameters, and the three genotype frequencies were calculated for each gene assuming HWE, resulting in an assortment of genotype frequencies across the nine genes, assuming biallelic polymorphisms. Each individual was randomly assigned a genotype for each gene, with probability *freq(genotype)*. A genotype pattern was then specified as having an enhanced risk. Each individual was then checked for a genotype match to the specified risk pattern and labeled a case or control by using penetrance parameters $R_1 = prob(case|match)$ and $R_0 = prob(case|not\ match)$ respectively, until a given number of cases and controls were generated.

As an illustration, consider a set of genes in which the minor allele frequencies are all 10 percent. HWE then gives $freq(AA) = 0.9*0.9 = 0.81$, $freq(Aa) = 2*0.9*0.1 = 0.18$, and $freq(aa) = 0.1*0.1 = 0.01$ for each gene. Using these frequencies, we assign genotypes for all genes to a population such that the genotype frequencies in the population approximate HWE. For example, each person is assigned G1 = AA with probability 81 percent, G1 = Aa with probability 18 percent, and G1 = aa with probability 1 percent. In this example, a population of 1000 people would thus have about 810 people with genotype AA, 180 people with genotype Aa, and 10 people with genotype aa, for each gene. We then specify a (multi-)gene pattern as conferring an increased risk of disease. For example, people with G1 = AA *and* G2 = Aa could have a 20 percent risk, that is, $p(case|match) = 0.2$, whereas everyone else, including people that match only one of the two genotypes in the pattern, has a 10 percent risk, that is, $p(case|not\ match) = 0.1$. Note that there is no increased risk for a partial match. We assign people to the cases with 20 percent probability if they have G1 = AA *and* G2 = Aa, and with only 10 percent probability otherwise. In this example, $0.81*0.18 = 0.1458$, or about 146 of the 1000 people would match, and of those, about 20 percent, or 29 people, would be cases, and 80 percent, or 117 people, would be controls. Of the 854 people who did not match, about 10 percent, or 85 people, would be cases, and 90 percent, or 769 people, would be controls. The 2 × 2 table for such a pattern is shown in *Table 2*.

Results and analysis

We used a variety of parameter values to construct a series of situations for application of our pattern discovery method. Populations were generated that had pattern complexity, allele or genotype frequencies, and risk (i.e., odds ratio) effects that are typical of those seen in molecular epidemiology studies. Among the properties explored were the effects of the complexity of the doped pattern (intentionally risk-enhanced pattern) and its overall frequency in the population (N_{with} / N_{total} in Table 1) on the ability of the algorithm to reliably detect the pattern. All populations were generated with $N_{cases} = 500$ and $N_{controls} = 1000$. The controls used were a random sample of the much larger number of controls generated. Pattern complexity was varied to include from one to five genes, and estimated pattern frequency, computed as the product of the individual genotype frequencies, ranged from 0.1 percent to 81 percent. The risks R_0 and R_1 were kept constant at 10 percent and 20 percent, respectively, resulting in an average odds ratio of 2.3. We used statistical significance, expressed as $-\text{Log}_{10}$(P-value), as a FOM for prioritizing the queue and ranking the results. The only constraint was that the minimum fraction of cases with the pattern of interest (support) was set to 5 percent for all the runs. For some of the rarer patterns, additional runs were made using 1 percent support.

Table 3 summarizes the results of some representative runs. The results of each run show the observed overall frequency, odds ratio, FOM = $-\text{Log}_{10}$(P-value), and rank of the discovered pattern out of the total patterns reported. A "–" indicates "pattern not discovered." At 5 percent support, the method reliably picks up patterns down to 5 percent frequency, and detects even rarer patterns. At 1 percent support, there are roughly 10 times more patterns reported, and reliable detection drops off at around 3 percent frequency. A larger population would be needed for reliable results at the 1 percent level. The combined run mixes two populations and resolves their components.

Figure 5 summarizes the coverage, detection, and #1 ranking for the runs in Table 3. Runs marked with a triangle detected the test pattern. Runs marked with asterisks ranked the test pattern #1. Of the 23 runs with 5 percent support, the test pattern was detected in 18, ranked #1 in 14, and in the top 10 in 17. Of the five patterns not detected at 5 percent support, four were rerun with 1 percent support, and three were detected, but not with high ranks due to insufficient sample size for such rare patterns. Nevertheless, for frequencies in the range typically encountered in clinical studies of common diseases, the pattern of interest is clearly visible.

The combination run demonstrates the utility of the method for resolving multiple etiologies in heterogeneous populations. The populations of runs 9 and 10, with two and three genes respectively, were mixed by combining the original two case files and two control files into a single pair with 1000 cases and 2000 controls. The algorithm was run with 5 percent minimum support. The two components were found at ranks #6 and #8. Note that the odds ratio and FOM are degraded from the separate component values because in the mixed population, the individuals from population #9 that happen to have pattern 10 do not have the enhanced risk of the individuals from population 10 with the same pattern, and *vice versa*. As a result, the odds ratios of both components are reduced to values consistent with a risk of 15 percent for each pattern in the combined population, compared to 20 percent in the separate populations. (We can average $R_0 = 10\%$ and $R_1 = 20\%$ because the proportions of both cases and controls in the mixture are equal.) To test this prediction we generated the two original populations 10 times each, with $R_1 = 15\%$ instead of 20% (data not shown). The observed odds ratio for the components in the mix was within 1 standard deviation of the mean odds ratio for the separate components with $R_1 = 15\%$. Given this worst-case mix scenario, the observed ranks (#6 and #8) are an indication of the strength of the overall method.

The lattice structure described previously is used to inspect the neighborhoods of the two components of the mixed pattern. *Figure 6* shows the relationships between the designated patterns 9 and 10 having ranks #8 and #6, respectively, and other patterns that ranked better. Rank #7 is a 3-gene specialization of the 2-gene component with rank #8. Ranks #4 and #2 are 4-gene specializations of the 3-gene component with rank #6, and ranks #3 and #1 overlap with two of the 3-gene component's features. Overall, all but one (#5) of the top 8 patterns differ from the designated patterns by either a single addition or a single substitution. The protective pattern with rank #5 has *no* features in common with any of the other patterns. This is not surprising, because one might

Table 3 Results of representative validation runs

Run	N genes	Component Genotype Freqs. (%)	Estimated Freq. (%)	Observed Freq. (%)	Odds Ratio	FOM	Rank	Total
A. Single-doped patterns run at minimum support of 5 percent								
1	1	81	81.0	82.7	2.5	7.8	1	1839
2	2	81, 73	59.3	62.6	2.4	13.4	1	1953
3	1	49	49.0	51.5	2.7	18.1	1	1236
4	3	81, 73, 66	39.0	42.6	2.2	11.5	1	1837
5	4	81, 73, 66, 59	22.9	29.2	2.0	8.8	1	1945
6	1	25	25.0	25.3	2.7	15.0	1	1034
7	2	49, 46	22.5	24.3	2.7	14.5	1	1260
8	5	81, 73, 66, 59, 53	12.0	14.5	2.3	7.4	1	1833
9	2	37, 35	13.0	13.9	2.0	5.2	1	1188
10	3	49, 46, 43	9.7	11.3	2.5	7.5	1	1211
11	2	25, 25	6.3	8.5	2.1	4.1	1	983
12	1	5.4	5.4	7.2	2.0	3.2	1	1753
13	3	37, 35, 32	4.2	5.6	3.2	6.4	1	1131
14	1	3.6	3.6	4.1	2.7	4.0	1	1736
15	1	7.7	7.7	9.3	2.4	6.1	2	1776
16	1	49, 37, 27	5.0	6.5	1.9	2.7	4	1134
17	4	52, 46, 40, 35	3.3	4.0	2.2	2.7	9	1683
18	4	49, 46, 43, 40	3.9	4.3	2.2	2.9	15	1144
19	1	2.1	2.1	—	—	—	—	1845
20	3	25, 25, 25	1.6	—	—	—	—	1017
21	5	49, 46, 43, 40, 37	1.4	—	—	—	—	1158
22	1	1	1.0	—	—	—	—	1796
23	4	25, 25, 25, 25	0.4	—	—	—	—	1003
B. Single-doped patterns run at minimum support of 1 percent								
24	1	25	25.0	25.3	2.7	15.0	1	11734
25	1	3.6	3.6	4.1	2.7	4.0	1	10601
26	2	25, 25	6.3	8.5	2.1	4.1	2	11685
27	1	49, 37, 27	5.0	6.5	1.9	2.7	38	12138
28	1	2.1	2.1	1.9	2.5	1.9	216	10199
29	3	25, 25, 25	1.6	2.2	1.9	1.3	1059	11651
30	1	1	1.0	1.2	2.0	1.0	2322	10005
31	4	25, 25, 25, 25	0.4	—	—	—	—	11633
32	5	25, 25, 25, 25, 25	0.1	—	—	—	—	11614
C. Double-doped patterns run at minimum support of 5 percent								
33a	2	37, 35	13.0	14.2	1.5	3.4	8	1145
33b	3	49, 46, 43	9.7	10.4	1.6	3.7	6	1145

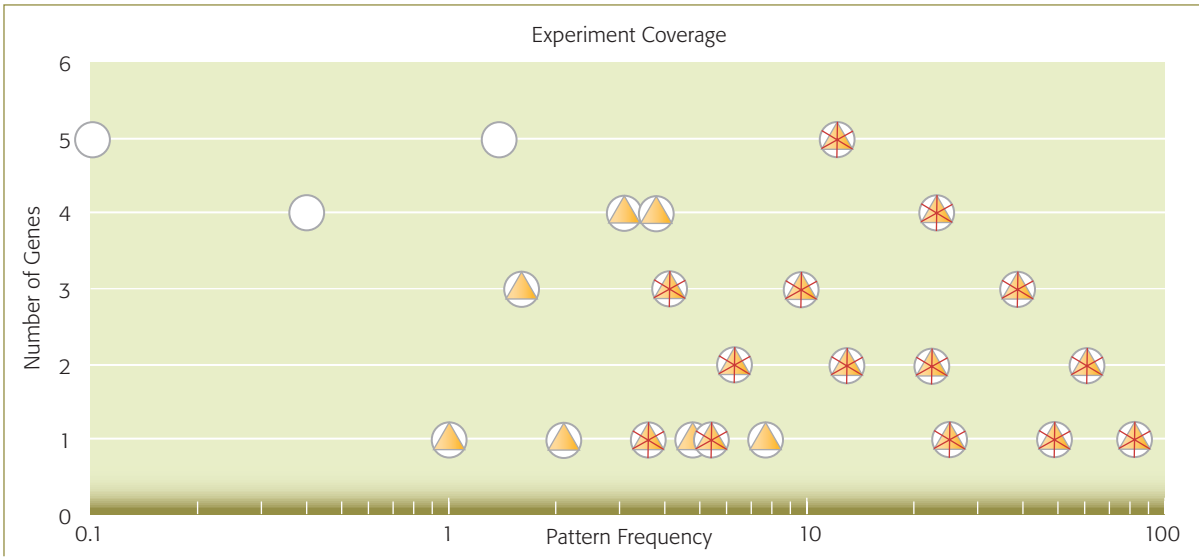

Figure 5
Coverage (◯), detection (△), and #1 ranking (✳) for the runs in Table 3

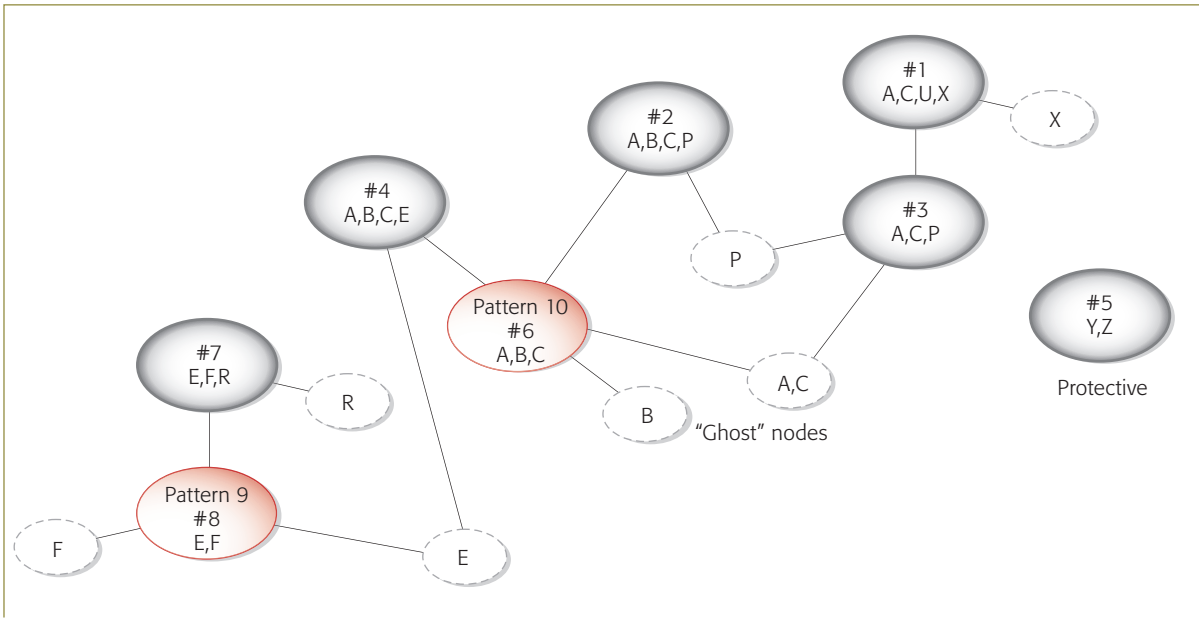

Figure 6
Sublattice surrounding the cliques of the two-component validation run

expect protective patterns and risky patterns to have little in common.

The rankings shown in Table 3 are based on P-value, and thus reflect an estimate of the type-1 error associated with statistics derived from the 2×2 table. For real data, the 2×2 table itself would have estimated errors for all four cells based on the genotyping error rate and the missing data rate. Quantitative estimates of these errors are beyond the scope of this paper. One could, however, explore the distribution of FOM, and in particular, the distribution of best FOM, for an ensemble of data sets with randomized affection status. This was not done for

our synthetic examples because we could compare the list of cliques directly with the known true patterns.

SUMMARY AND FUTURE DIRECTIONS

We have described a graph-theoretical approach to searching for patterns in categorical data and demonstrated its ability to reliably detect and identify patterns that confer risk in situations typical of molecular epidemiological case-control studies. In particular, our method performs well on multivariate patterns that often present a challenge for traditional methods. The technique can be used to sort out multiple patterns conferring independent risks, indicating its potential for resolving multiple etiologies. We also showed how the use of the lattice concept can be helpful in understanding the relationships among discovered patterns.

An effort to expand the scope of our approach and enhance the features and performance of the implementation is under way. Until now we have focused on the effects one clique (or family of cliques) at a time. The individual elements of a clique are implicitly connected with the logical AND operator. When considering more than one clique from separate families, the combination is aptly described using the logical OR operator between the cliques. Hence, a full description of a combination of cliques would involve a mixed Boolean expression. We are currently working on an extension to our method that automatically detects such combinations and simplifies their Boolean descriptions.

Another area for future work involves more extensive analysis of the overlaps in source sets and terminal sets within the lattice. The strict definition of clique can be relaxed, allowing sets with less than complete connectivity to participate in patterns, perhaps more closely approximating the situations encountered in real studies. We are exploring additional enhancements to the algorithm, including more complex FOM and constraints. We are currently engaged in analyzing data from a real epidemiological study involving genetic and environmental risk factors for breast and endometrial cancer.

CITED REFERENCES

1. L. Breiman, R. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Boca Raton, Chapman & Hall (1984).
2. A. S. Foulkes, M. Reilly, L. Zhou, M. Wolfe, and D. J. Rader, "Mixed Modeling to Characterize Genotype Phenotype Associations," *Statistics in Medicine* **24**, No. 5, 775–789 (2005).
3. *Probabilistic Methods in Discrete Mathematics: Proceedings of the Fourth International Petrozavodsk Conference, Petrosavodsk, Russia, June 3–7, 1996*, V. F. Kolchin, V. YA Kozlov, Y. L. Pavlov, and Y. V. Prokhorov, Editors, V. S. P. International Science (1997).
4. M. Nelson, S. Kardia, R. Ferrell, and C. Sing, "A Combinatorial Partitioning Method to Identify Multilocus Genotypic Partitions That Predict Quantitative Trait Variation," *Genome Research* **11**, 458–470 (2001).
5. J. Hoh, A. Wille, R. Zee, S. Cheng, R. Reynolds, K. Lindpaintner, and J. Ott, "Selecting SNPs in Two-Stage Analysis of Disease Association Data: A Model-Free Approach," *Annals of Human Genetics* **64**, 413–417 (2000).
6. J. Lepre, J. J. Rice, Y. Tu, and G. Stolovitzky, "An Efficient Algorithm for Pattern Discovery and Multivariate Feature Selection in Gene Expression Data," *Bioinformatics* **20**, No. 7, 1033–1044 (2004).
7. J. H. Friedman, "Multivariate Adaptive Regression Splines," *Annals of Statistics* **19**, 1–66 (1991).
8. R. Schapire, "The Strength of Weak Learnability," *Machine Learning* **5**, No. 2, 197–227 (1990).
9. V. Vapnik, *The Nature of Statistical Learning Theory*, New York, Springer (2000).
10. J. H. Friedman and J. Tukey, "A Projection Pursuit Algorithm for Exploratory Data Analysis," *IEEE Transactions On Computers* **C-23**, No. 9, 881–889 (1974).
11. J. H. Friedman and W. Stuetzle, "Projection Pursuit Regression," *Journal of the American Statistical Association* **76**, 817–823 (1981).
12. N. Tahri-Daizadeh, D. Tregouet, V. Nicaud, N. Manuel, F. Cambien, and L. Tiret, "Automated Detection of Informative Combined Effects in Genetic Association Studies of Complex Traits," *Genome Research* **13**, No. 8, 1952–1960 (2003).
13. J. Huang, A. Lin, B. Narasimhan, T. Quertermous, C. A. Hsiung, L. T. Ho, J. S. Grove, M. Olivier, K. Ranade, N. J. Risch, and R. A. Olshen, "Tree-Structured Supervised Learning and the Genetics of Hypertension," *Proceedings of the National Academy of Sciences of the United States of America* **101**, No. 29, 10529–10534 (2004).
14. J. Zhu and T. Hastie, "Classification of Gene Microarrays by Penalized Logistic Regression," *Biostatistics* **5**, No. 3, 427–443 (2004).
15. D. V. Conti, V. Cortessis, J. Molitor, and D. C. Thomas, "Bayesian Modeling of Complex Metabolic Pathways," *Human Heredity* **56**, Nos. 1–3, 83–93 (2003).
16. V. Cortessis and D. C. Thomas, "Toxicokinetic Genetics: An Approach to Gene-Environment and Gene-Gene Interactions in Complex Metabolic Pathways," *International Agency for Research on Cancer Scientific Publication* **157**, 127–150 (2004).
17. J. Millstein, D. Conti, F. Gilliland, and W. Gauderman, "A Testing Framework for Identifying Susceptibility Genes in the Presence of Epistasis," *American Journal of Human Genetics* **78**, No. 1, 15–27 (2006).
18. G. Alexe, S. Alexe, Y. Crama, S. Foldes, P. Hammer, and B. Simeone, "Consensus Algorithms for the Generation of All Maximal Bicliques," *Discrete Applied Mathematics* **145**, No. 1, 11–21 (2004).

19. M. D. Ritchie, L. W. Hahn, N. Roodi, R. Bailey, W. D. Dupont, F. F. Parl, and J. H. Moore, "Multifactor Dimensionality Reduction Reveals High-Order Interactions among Estrogen-Metabolism Genes in Sporadic Breast Cancer," *American Journal of Human Genetics* **69**, No. 1, 138–147 (2001).
20. L. Bastone, M. Reilly, D. L. Rader, and A. S. Foulkes, "MDR and PRP: A Comparison of Methods for High-Order Genotype-Phenotype Associations," *Human Heredity* **58**, No. 2, 82–92 (2004).
21. C. Verzilli, N. Stallard, and J. Whittaker, "Bayesian Graphical Models for Genomewide Association Studies, *American Journal of Human Genetics* **79**, No. 1, 100–12 (2006).
22. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proceedings of the 20th International Conference on Very Large Databases*, Santiago de Chile, Chile, September 12–15, 1994, pp. 487–499.

Accepted for publication August 22, 2006.
Published online December 31, 2006.

Richard A. Mushlin
IBM Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (mushlin@us.ibm.com).
Dr. Mushlin is a research staff member in the Functional Genomics and Systems Biology group of the Computational Biology Center. He received a Ph.D. degree in chemical physics from Brandeis University in 1979, and joined IBM in 1982. Dr. Mushlin's long and diverse career at IBM includes developing an experimental MRI system, managing a software development group, deploying a World's Fair kiosk system, designing clinical information system applications, and most recently, developing tools and methods for clinical genomic data analysis.

Aaron Kershenbaum
IBM Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (aaronk@us.ibm.com).
Dr. Kershenbaum is a research staff member in the Systems department at the Thomas J. Watson Research Center. He received B.S.E.E and M.S.E.E. degrees simultaneously in 1970 from the Polytechnic Institute of Brooklyn and a Ph.D.E.E. degree in 1976 from Polytechnic University. He joined Network Analysis Corporation in 1969, where he helped design some of the earliest computer networks, including the ARPANET and NASDAQ. In 1978, he joined the faculty at Polytechnic University, where he helped develop a curriculum in computer communications and served as Director of the Network Design Laboratory. He joined IBM in 1990, where he has done research in applying network theory to problems in communications network design and analysis, computational biology, computational linguistics, and interactions among concepts in ontologies. He received an IBM Outstanding Innovation Award for his work in computer network design. Dr. Kershenbaum is a member of the ACM and a Fellow of the IEEE.

Stephen T. Gallagher
University of Pennsylvania, Blockley Hall, 423 Guardian Drive, Philadelphia, Pennsylvania 19104 (sgallagh@cceb.med.upenn.edu).
Mr. Gallagher is a Programmer Analyst in the Center for Clinical Epidemiology and Biostatistics at the University of Pennsylvania School of Medicine. His work in LIMS development and genetics databases supports the Laboratory for Molecular Epidemiology. His primary research interests lie in efficient data management of large-scale study data and novel methods of pattern detection.

Timothy R. Rebbeck, Ph.D.
University of Pennsylvania, 904 Blockley Hall, 423 Guardian Drive, Philadelphia, Pennsylvania 19104 (trebbeck@cceb.med.upenn.edu).
Dr. Rebbeck is Professor of Epidemiology, co-leader of the Abramson Cancer Center's Cancer Epidemiology and Risk Reduction program, Director of the Center for Genetics and Complex Traits, Director of the Center for Population Health and Health Disparities, and Director of the Laboratory for Molecular Epidemiology at the University of Pennsylvania School of Medicine. Dr. Rebbeck's research focuses on the genetic and molecular epidemiology of cancer. He has directed several molecular epidemiologic studies whose goals were to identify and characterize genes that are candidates for involvement in cancer etiology, and to describe the relationship of allelic variation of these genes with biochemical or physiological traits, cancer occurrences, and cancer outcomes. ■