

Backward Chaining Rule Induction

Douglas H. Fisher*, Mary E. Edgerton**, Zihua Chen**, Lianhong Tang***, Lewis Frey***

* Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235

** Department of Interdisciplinary Oncology, H. Lee Moffitt Cancer Center and Research Institute, SRB-3, 12902 Magnolia Drive, Tampa, FL 33612

*** Department of Biomedical Informatics, Vanderbilt University, Nashville, TN 37232

Abstract

Exploring the vast number of possible feature interactions in domains such as gene expression microarray data is an onerous task. We describe Backward-Chaining Rule Induction (BCRI) as a semi-supervised mechanism for biasing the search for IF-THEN rules that express plausible feature interactions. BCRI adds to a relatively limited tool-chest of *hypothesis generation* software and is an alternative to purely unsupervised association-rule learning. We illustrate BCRI by using it to search for gene-to-gene causal mechanisms that underlie lung cancer. Mapping hypothesized gene interactions against prior knowledge offers support and explanations for hypothesized interactions, and suggests gaps in current knowledge that induction might help fill.

BCRI is implemented as a wrapper around a base supervised-rule-learning algorithm. We summarize our prior work with an adaptation of C4.5 as the base algorithm (C45-BCRI), extending this in the current study to use Brute as the base algorithm (Brute-BCRI). In contrast to C4.5's greedy strategy, Brute extensively searches the rule space. Moreover, Brute returns many more rules (i.e., hypothesized feature interactions) than does C4.5. To remain an effective hypothesis-generation tool requires that Brute-BCRI more carefully rank and prune hypothesized interactions than does C45-BCRI. Prior knowledge serves to evaluate final Brute-BCRI rules just as it does with C45-BCRI, but prior knowledge also serves to evaluate and prune intermediate search states, thus maintaining a manageable number of rules for evaluation by a domain expert.

Keywords: rule induction, hypothesis generation, interactive induction, machine learning, data mining, prior knowledge, microarray, data analysis, molecular mechanisms, class discovery, semi-supervised learning, decision trees, Brute, non-small cell lung cancer, systems biology

Corresponding author:

Douglas H. Fisher
Department of Electrical Engineering and Computer Science
Box 1679-B, Vanderbilt University
Nashville, TN 37235
douglas.h.fisher@vanderbilt.edu

1. Introduction

With the increasing investment in gene expression microarray technology, there has been a move toward a “systems biology” approach to understanding the coupling of gene networks and signaling cascades that describe the phenotypes of living matter (e.g., [13],[35],[24]). This has led to a call for tools to (semi-)automatically explore the space of genomic interactions (e.g., [14]) in order to reduce the set of interactions to a manageable set for examination. The goal of this exploration is to focus analysts on plausible interactions, pathways, and markers, which can then be scrutinized further with hypothesis testing methods.

We describe *backward-chaining rule induction* (BCRI) for hypothesizing gene-to-gene interactions from gene expression microarray data. BCRI is a novel strategy for restricting the search through a rule-space to those rules with traceable influence on a given top-level target class. Put simply, BCRI is given a top-level classification with labeled data, and rule induction is performed to find rules that predict the specified class. Antecedent conditions found in discovered rules then become “sub-goals”, and rule induction is repeated on the data using these sub-goal conditions as classes. The process of backward-chaining on rule antecedent conditions is repeated until a termination condition is satisfied.

While our motivation for developing BCRI was to explore gene expression and related biomedical data, the method is not inherently limited to such domains. Generally speaking, BCRI is intermediate between supervised rule induction and unsupervised rule induction (e.g., association rule learning). Rather than an unconstrained exploration of the space of associations between variables, as would occur in association rule learning [18], only associations that are weakly tied to a top-level class are examined. While BCRI’s search through association space will miss many associations (with any given top-level class), our goals are that the density of “interesting” rules that BCRI discovers will be higher than the set uncovered by standard association rule learning, and that number of rules that are presented for review by a human data analyst are manageable.

BCRI can be viewed as one component in a process of *iterative exploration*. Induction from data (e.g., BCRI) can be used to find plausible interactions, which are then compared against prior knowledge. Prior knowledge can be used to (1) explain plausible interactions found through induction, (2) filter or rank these possibilities for an analyst (e.g., interactions that are already well-established in the literature might be ranked low, as might be those in which prior knowledge offers too few constraints on possible explanations), (3) implicate additional features or suggest pruning “redundant” features for subsequent induction (e.g., feature selection), (4) reveal gaps in current knowledge that induction may help fill.

In prior work (Fisher et al, 2005) our contributions were (1) the definition of the BCRI task abstraction, (2) the implementation of an initial prototype of BCRI, which we call *C45-BCRI*, (3) an illustration of BCRI in the domain of cancer prognosis, and (4) a demonstration of how BCRI generated hypotheses (e.g., gene interactions) may help fill gaps in prior knowledge. These results have proved biologically interesting and have been elaborated significantly in the medical literature [7].

In Section 2 we summarize our prior research with *C45-BCRI* on gene expression data in the lung cancer domain. Section 3 discusses the limitations of *C45-*

BCRI as a hypothesis generator, particularly (but not solely) with respect to our biomedical application. This section also motivates the use of a more search-intensive rule discovery strategy.

Section 4 introduces Brute-BCRI, which uses Brute [30] as the base rule-induction engine around which we wrap rule exploration. Brute discovers many more rules than C4.5 and the desire to constrain the search for hypotheses motivates an *interactive induction* approach in which guidance from the data analyst, prior knowledge, and data visualization are interleaved with the rule discovery process. Section 5 analyzes selected results from interactive Brute-BCRI. Section 6 closes with directions for further research.

2. Backward-Chaining Rule Induction

Consider a scenario in which an interested student is asked to find evidence from the literature that smoking causes cancer. The student finds such evidence in the form of published studies that empirically link smoking history and several forms of cancer. These studies collectively reference empirical evidence that peer-pressure promotes teenage smoking and increased stress is associated with smoking at all ages. Our interested student then researches the psychological dynamics of teenage peer pressure, finding and then “backward” researching associations between body self-image and success at resisting peer pressure. Having explored various intellectual paths rooted by teenage peer pressure, our student might return to a semi-focused literature review of stress, uncovering associations with high blood pressure, heart disease, and diminished mental agility.

BCRI is a formalized version of this intuitive, oft-performed process of finding evidence in support of initial goals, then exploring backward and outward. Specifically, the initial step of BCRI builds decision rules for predicting a user-specified class or outcome. The antecedents of rules discovered in this first step then become outcomes for which decision rule models are constructed in the second step. Antecedents of rules found in this second step, then become outcomes for decision rule learning in the third step, and so on. This section summarizes our prior work with BCRI, including (1) a formalization of the BCRI task abstraction, (2) the implementation of an initial prototype of BCRI, which we call *C45-BCRI*, (3) an illustration of BCRI in the domain of cancer prognosis, and (4) a demonstration of how BCRI generated hypotheses (e.g., gene interactions) suggested filling gaps in prior knowledge [10][7].

2.1 Applying BCRI to Gene Expression Data

We previously applied BCRI to published gene-expression and clinical data from lung cancer patients [2]. The data contains 61 instances defined over 4,996 gene attributes and eleven clinical attributes (5007 total). Classification as High versus Low risk is the as a top-level task that “kick-starts” BCRI in our application. For our analysis, patients who died at 30.1 months or less following diagnosis are high risk, and others are low risk. Details of our selection of 30.1 months as discriminating high and low risk can be found in [7].

In this application, BCRI initially applies a rule induction algorithm to discover the conditions that predict a clinical outcome -- *long* versus *short* survival periods for lung cancer patients. This produces a set of human readable rules of the form “IF

<conditions> THEN <outcome>”. This type of rule is of particular interest because it can be translated into hypothetical mechanisms for subsequent validation. For example, the rule

IF < (A-kinase anchoring protein expression is greater than 496) **AND**
(urea transporter expression is less than or equal to 397) >
THEN < Annexin V, a phospholipase inhibitor, expression levels
are greater than 750 >

can be translated into a hypothesis stated as up-regulation of A-kinase in combination with down-regulation of the urea transporter causes the phospholipase inhibitor Annexin V to be down regulated.

The rule above is one example of a rule learned by our initial implementation of BCRI. We describe this implementation next, followed by a fuller account of the results obtained with it.

2.2 C45-BCRI: An implementation of BCRI

We distinguish the general BCRI strategy from the possible implementations of this strategy. Our initial approach, reported in [10], implements BCRI as a “wrapper” around a rule-induction engine. This design, while not the most efficient from a computational point of view, makes sense in early system development because it allows us to plug in and experiment with different rule-induction algorithms.

In a wrapper-based implementation, BCRI is passed the labeled gene expression data, a set of the target classes used to label the data, and three functions: **RuleInducer**, **PriorityFn**, and **TerminateFn**. We have included pseudocode (pseudo-C code with local variable declarations excluded) in Table 1.

Data is a data set such as the Beer et al data set. *Classes* is a set of class labels that are included in and used to classify *Data*. **RuleInducer** is a function of two parameters (i.e., a supervised rule induction system) that learns if-then rules to predict a *TargetCond* from a *DataSet*. **PriorityFn** is a function that takes an if-then rule and returns a floating point priority value associated with the rule. This priority value is used to order the rule on a priority queue, and this priority queue is used to guide the exploration of rules to which backward chaining is applied. **TerminateFn** is a function that decides whether a given rule should be backward chained.

RuleInducer can, in principle, be any supervised rule discovery system that, given a class, will return rules that predict that class. Our current implementation adapts C4.5 (release 8, using default settings), developed by Quinlan [27] for **RuleInducer**. We point out some of the implications of this choice and other possible options for rule induction engines in Section 3.

Table 1. Pseudocode for BCRI

```

Function Wrapper-BCRI
Returns a RuleSet
With parameters DataSet Data
                TargetSet Classes
                RuleSet Function RuleInducer (DataSet, TargetCond)
                float Function PriorityFn(Rule),
                bool Function TerminateFn (Rule)) {
RuleSet ResultantRules = empty RuleSet
PQ = InitializePriorityQueue(PriorityFn);
FOR each class in Classes, Enqueue(PQ, [class → ____]);
WHILE (NOT Empty(PQ)) {
    R = Dequeue(PQ);
    ResultantRules = ResultantRules + R;
    IF (NOT TerminateFn(R) {
        FOR each a IN ANTECEDENTS(R) {
            Children = RuleInducer (Data, a);
            FOR each c IN Children Enqueue(PQ, c)
        }
    }
} /* end WHILE */
RETURN ResultantRules
} /* end BCRI */

```

We wrote a small portion of code to parse the C4.5 output and convert the decision trees into if-then rules with associated *coverage* (i.e., the number of instances in the data that satisfy the rule's antecedent) of the rule for subsequent analysis by **PriorityFn**. We use the abbreviation 'cov' to designate coverage in our presentation of the results.

PriorityFn is applied to a rule and returns a score. This score is used to store the rule on a priority queue of other scored rules. In the initial implementation, henceforth *C45-BCRI*, the coverage (COV) of the rule is used to organize the priority queue. Other possibilities include the rule's accuracy or the like. Our choice of coverage, versus accuracy or a like measure, is motivated by the observation that rule-learning systems will tend to produce accurate rules (relative to a data specific upper bound), but that these rules will vary significantly in coverage. We prefer to favor rules that cover a large proportion of data.

At each iteration of **RuleInducer**, new *TargetConditions* are defined based upon the rules. Consider the antecedent conditions for the rule given earlier:

```

IF < (A-kinase anchoring protein expression is greater than 496) AND
    (urea transporter expression is less than or equal to 397)>

```

There would be a *TargetCondition* representing that A-kinase anchoring protein expression value was greater than 496, and a separate *TargetCondition* representing that the urea transporter expression was less than or equal to 397. Each of these two *TargetConditions* would be separately submitted to **RuleInducer** to generate decision rules to predict them, and each of these resulting decision rule would be scored and placed on the priority queue.

We continue the iteration until **TerminateFn**, which indicates whether a rule should be further expanded (i.e., continued backward chaining on its antecedents), returns a value of false. In C45-BCRI we use a depth bound of three to terminate BCRI (with the first rule to predict high and low risk for survival classes as the zero-th level class).

2.3 A Sample Trace of C45-BCRI

Using High and Low Risk as the top-level classification, C45-BCRI begins with (Risk=Low) (cov 42/61) and (Risk=High) (cov 19/61) placed on the priority queue (i.e., passed as Classes to C45-BCRI). Again, 'cov' is an abbreviation for data coverage

(Risk=Low) is dequeued. Application of **RuleInducer** in the C45-BCRI implementation yields a single rule, which is placed on the queue:

[(Stage=1) → (Risk=Low) (cov 48/61) || (Risk=High) → (cov 19/61)].

(Stage=1) → (Risk=Low) is dequeued and **RuleInducer** yields a rule, which is added to the queue:

[(ELA2 > 163.3) → (Stage 1) (cov 45/61) || (Risk = High → (cov 19/61)]

The first of these rules is dequeued. A new rule is learned:

(MRPL19<= 161.4) & (EIF2S1 > 52) & (KRT15 <= 616.8) → (ELA2 >163.3) (cov: 45/61)

This rule is queued, resulting in the following priority queue:

[(MRPL19<= 161.4) & ... → (ELA2 >163.3) (cov: 45/61) || (Risk = High → (cov 19/61)]

Having the highest priority (coverage), this same rule is immediately dequeued. Each individual antecedent serves in turn as a class and rules. A simple depth bound of three is used to terminate backward chaining, and these labeled rules are terminal.

Table 3 shows the 19 rules learned from the lung cancer data by backward chaining to depth of three, beginning with an initial queue of

[(Risk = Low) → || (Risk = High) →]

Gene definitions are given in [7]. Coverage (cov) has already been defined; “acc” denotes accuracy, the percent of correct predictions (consequent) across the data matching the rule’s antecedents. As Figure 1 illustrates, the network of rules learned by BCRI (in general) is an AND/OR graph, much like the rule bases of expert systems such as Mycin [31]. We do not discuss the inference possibilities of such networks in this paper. Rather, our goal is a limited, focused exploration of the associations between variables, which is directly (initially) or indirectly (as backward chaining proceeds) tied to top-level class(es).

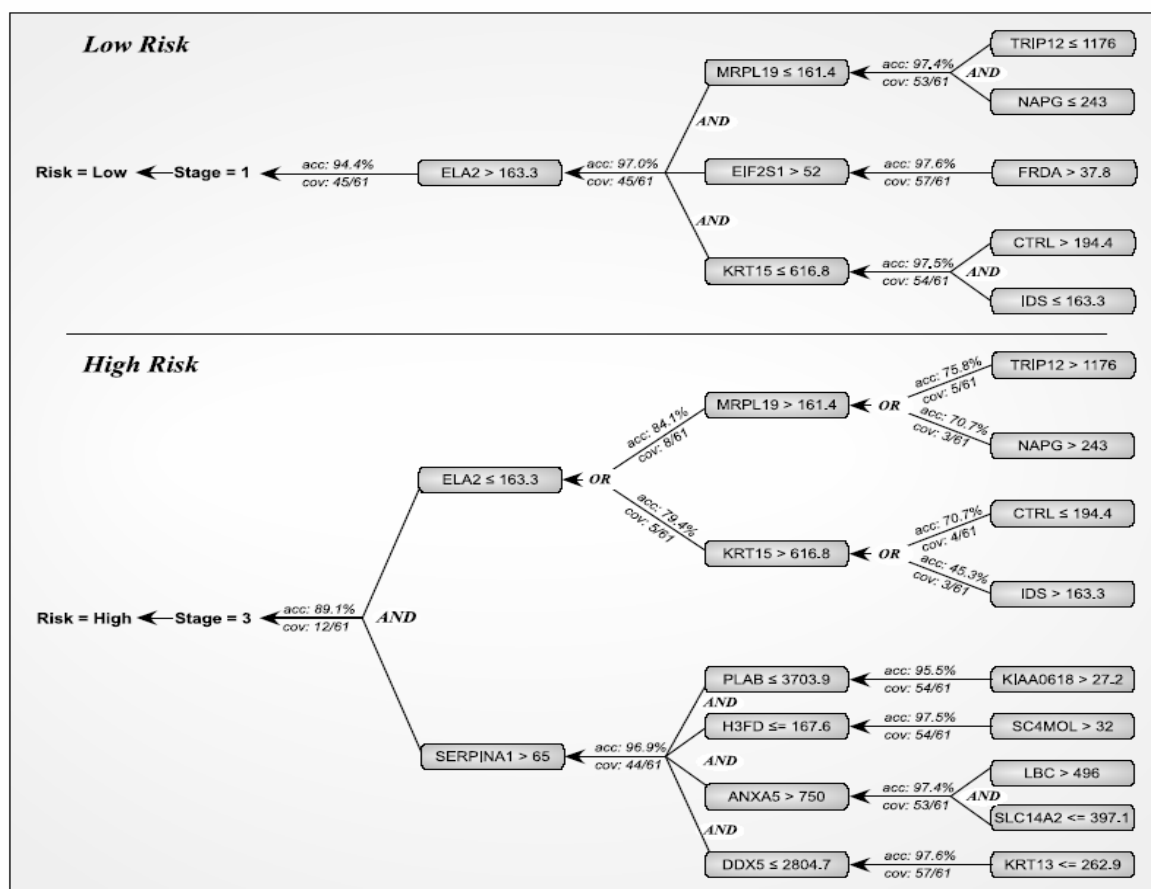


Figure 1: AND/OR rule network formed by C45-BCRI

2.4 Gene interactions learned from C45-BCRI: cancer and lung cancer relevance

Once we have generated a set of rules, we take each rule’s conjunctive antecedent as a potential interaction between genes. We also examine the relationship between the gene(s) specified in the antecedent condition and the consequent, comparing the interactions against prior knowledge. Known interactions identified from PubMed and GeneCards [28] along with chromosomal location from LocusLink [25][26][31] as established domain knowledge with which to compare the interactions hypothesized from BCRI. The OR conditions, which we will focus on in Section 3, would be considered to be independent pathways that results in the same consequent.

Our hypothesis is that BCRI will generate a set of rules that are densely populated by “interesting” rules when initiated by top-level classes of interest. We expect BCRI to generate interactions that are already well-known, interactions that are partially

supported, and novel interactions that can be used to generate hypotheses about networks relevant to lung cancer.

In our earlier work [10][7] 17 rules were generated (excluding the selection of Stage in the two top-most level rules) with 19 molecular species. Of the 19 molecular species selected in the rules, 12 have been associated with neoplasia in general and five specifically with lung. Using MetaCore™ [22], a pathway database tool, we find that ten of the species are connected by known pathways within a distance of 3 nodes. Of the 17 rules with gene interactions discovered in the C45-BCRI session, 12 are evaluated as plausible in terms of our knowledge base today. Of these, five of the interactions have been specified previously in the literature, although not necessarily related to lung cancer, and seven are new associations. Edgerton et al [7] gives a detailed analysis along these lines, giving both a quantitative summary and breakdown of newly hypothesized and previously known associations, as well as detailed biological explanations of why these associations are plausible and merit further investigation.

As part of our summary of prior work, we give an example of combining induced knowledge in the form of a C45-BCRI rule and existing knowledge to facilitate inference about a pathway. Fisher et al [10] and Edgerton, et al [7] give other examples where C45-BCRI rules and prior knowledge using Pathway Assist™ [21] facilitate inference about molecular pathways.

Consider the rule

(FXN > 37.8) → (EIF2S1 >52) [acc: 97.6% cov: 57/61]

which is a deepest rule of the low risk subtree in Figure 1, and where FXN is the Pathway Assist™ synonym for FRDA, the label used for the same gene in Figure 1.

Pathway Assist™ shows that FXN has an “unknown” effect on the molecular synthesis of heme, the interaction represented as a solid line with a square in Figure 2, and that heme, a small molecule depicted by the small, central oval, inhibits the gene expression of EIFS2. Relational details are listed in Table 2.

From our C45-BCRI rule, if we accept that elevated gene expression of FXN leads to elevated levels of its protein product frataxin, then we can infer that frataxin blocks the molecular synthesis of heme which results in elevated expression of EIFS2. Thus, the inductively derived rule, which might be tentatively abstracted as (FXN → EIF2S1, effect positive), together with (heme--|EIF2S1, effect negative) from prior knowledge, suggests that (FXN→ heme, effect negative, in place of unknown).

This example illustrates where induction can suggest fillers for gaps in background knowledge, and generally illustrates a gooddomain-independent hypothesis generation strategy. In this example, constraints from background knowledge and a C45-BCRI rule were sufficient to suggest, through qualitative reasoning, the value of an unknown effect. Of course, this reasoning only suggests hypotheses, and other examples are found in our data [10][7].

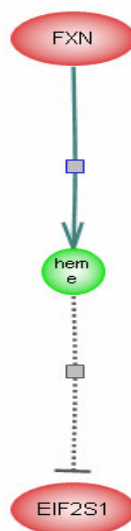


Figure 2. Pathway Assist™ diagram illustrating FXN and EIF2S1 relationships.

Table 2: Details of Figure 2 relationships given by Pathway Assist™.

<u>Type</u>	<u>Nodes</u>	<u>Effect</u>
Regulation	heme --- EIF2S1	negative
MolSynthesis	FXN ---> heme	unknown

2.5 Summary of prior work with BCRI

We have used BCRI as a means of biasing the search for gene interactions, and feature interactions generally. In particular, the contributions of prior work are (1) the definition of the BCRI task abstraction, (2) the implementation of an initial prototype of BCRI, which we call C45-BCRI, (3) an illustration of BCRI in the domain of cancer prognosis, and (4) an illustration of how (C45-)BCRI generated rules (e.g., gene interactions), coupled with prior knowledge, suggest hypotheses about the ways that genes interact that is not yet established in the literature. Importantly, these hypotheses are interesting to a medical audience [7].

The next section discusses the limitations of our C4.5-based implementation of BCRI, and motivates an alternative choice for the base rule-induction engine that more extensively searches the space of rules, and generally returns many more rules. This latter characteristic further motivates and necessitates work into using background knowledge and other heuristic strategies to filter/rank hypothesized interactions for expert commentary.

3. Dimensions for describing BCRI implementations

This section highlights several important characteristics of C45-BCRI as a tool for hypothesis generation. More generally, we discuss dimensions along which BCRI implementations can vary, pinpoint C45-BCRI in this space, and motivate a second implementation that we call Brute-BCRI.

3.1 Wrapper-based, performance-optimized, and Interactive implementations

C45-BCRI wraps rule exploration around a base rule-induction learner. The advantage of this approach is that the loose coupling between components in a wrapper model facilitates experimentation with different implementations of sub-functionalities, notably with the function parameters to Wrapper-BCRI of Table 1: *RuleInducer*, *PriorityFn*, and *TerminationFn*. The disadvantage of a wrapper-based approach is that the loose coupling can also compromise runtime performance. We are still at a stage of research where performance-optimized implementations do not make sense. In fact, shortly we will motivate even further loosening of coupling between sub-functionalities, moving to an interactive approach that folds the human analyst into the exploratory loop.

3.2 Search Control: Greedy to search-intensive rule learners

C4.5 is a greedy rule learner. The decision tree from which rules are extracted (as paths in the tree) is built through top-down induction using one-step look-ahead in selecting each divisive attribute. This general strategy for decision tree induction is so pervasive an approach [27] that we have not even bothered summarizing it. There are also other forms of greedy rule learning, notably the strategy of separate-and-conquer that is followed by systems such as CN2 (...).

There are also more search intensive approaches to decision tree induction [20] and rule learning generally. Our particular interest is with Brute [30], which generates rules that predict a given *TargetCondition*. Brute is a *data-driven* method like C4.5 – candidate rules are “generated” directly from the data. In contrast, a *model-driven* approach generates candidate rules independent of the data, but evaluates the candidates against the data. The data-driven versus model-driven distinction [6] becomes particularly relevant later when we discuss the possibility of rule generation from prior knowledge and subsequent evaluation against data.

In principle, a Brute-like system could be configured to generate and evaluate every decision rule that follows from the data, but parameter settings and hard-coded heuristics typically result in a much less than exhaustive search.

3.3 Discriminant versus characteristic rule learners

There are discriminant and characteristic rule learners [6]. A discriminant learner, or minimal description learner, results in a minimal set of rules necessary to discriminate the *TargetCondition* (or target class) from a given contrast class. Examples of the contrast class, which in practice might be the union of several contrast classes, are also provided as data. Thus the actual contrast-class data may occupy a small region of the space implied by \neg *TargetCondition*. C4.5 is a discriminant learner. The implications of this property in C45-BCRI are that relatively few OR nodes are present in a final network. C4.5 uncovers relatively few alternative ways to infer a given *TargetCondition*, because relatively few ways are necessary to discriminate data labeled *TargetCondition* from data labeled \neg *TargetCondition*.

Brute, in contrast, is a characteristic learner. The goal of a characteristic or maximal-description learner is to find all rules that are common (fully characterize) data labeled by *TargetCondition*. As such, a characteristic base learner would result in an

implementation of Wrapper-BCRI (Table 1) that introduces more OR nodes in the final rule network.

By (near) necessity, a characteristic learner tends to be search intensive – to find all or many commonalities in TargetCondition data requires looking at more patterns than might be required by a discriminant learner. We have noted that discriminant learners can be search intensive if focused on finding the best discriminating rules, but greedy solution strategies often lead to acceptable results in the case of discriminant learning. Of course, a discriminant learner such as C4.5 can be wrapped in a bootstrapping (or similarly intended) loop to build many decision trees (and rules), thus yielding a characteristic learner. Bagging of decision trees [4] does precisely this.

In Section 3.4 we motivate the importance of a characteristic rule learner for our application, and later we describe a specific implementation that uses Brute as the base learner.

3.4 Application-specific motivation for a characteristic learner

Researchers are predisposed to search for simple solutions (i.e., a single molecular profile representing a pathway to cancer) to predict outcomes for cancer patients. Selection rules are designed to achieve single solutions, such as defining cancer subtypes based on the anatomic site (e.g. lung, breast) and the cell type of origin (e.g. carcinoma arising from an epithelial cells, sarcoma arising from the stromal cells). Patients can be further divided into subtype of carcinoma. e.g. adenocarcinoma or squamous cell carcinoma, depending upon the function based specialization of the cell of origin, and into well, moderately, or poorly differentiated cancer depending upon how close the morphology resembles normal tissue. The time course of the tumor is supposedly accounted for in staging models, which are based on size and extent of spread, to predict an outcome for a patient with a subtype of cancer from a specific location. Even when these selection rules are applied to subclassify patients prior to exploring their gene expression profiles, however, we are often unable to accurately predict an outcome.

In fact, the most common tumors, the epithelial tumors, have proven very difficult to profile. Studies of gene expression have focused on attempting to assign a uniform molecular profile to a clinical phenotype-e.g. a cancer patient with a short versus a long survival time. However, this may not be a feasible approach to the study of epithelial tumors. There are likely many alternative pathways that can be hypothesized as mechanisms to describe different subsets of the patient population that have the same clinical phenotype. Thus, the implementation of an algorithm that generates “OR” nodes is a more appropriate way to study epithelial tumors in general and lung adenocarcinoma in particular. Our extension of BCRI using Brute is not only justified but necessary so that we can appropriately model lung adenocarcinoma.

In general, a characteristic base learner is desirable in any domain that exhibits causal *heterogeneity* (aka overdetermination). Heterogeneity also appears important in studies of the inheritability of certain diseases, for example – in such cases, there is not one gene that dictates the passing on of a disease, but many factors contribute.

3.5 OR-overlap ranging from exclusive OR to full overlap

Different rules with the same consequent condition represent OR nodes in a rule network resulting from BCRI. In some cases these may represent exclusive OR nodes representing mutually exclusive conditions. In other cases alternative rules may represent an inclusive OR, with varying degrees of overlap in the data that matches each rule's antecedent. Given that the antecedents of each rule represent distinct populations of data, a question that motivates subsequent analysis is the extent to which these populations overlap.

If the antecedents of two rules are logically inconsistent, then the overlap is empty of necessity. If the antecedents are not a contradiction, then we can look at the extent of overlap in the training data. If there is little or no overlap then this "empirical inconsistency" suggests that there may be biological causes that are worth investigating – we are dealing with what appear to be two alternative, and in this case mutually-exclusive, pathways to the same outcome. This can also motivate further data collection to validate the observation. In contrast, if the antecedents of two rules cover the same subclass of data then their conjunction may represent a single pathway.

In general, when analyzing OR nodes we are interested in the extent of *logical* (in)consistency and data or *empirical* (in)consistency. In particular, the extent of data overlap, with empty overlap representing mutual exclusion (for biological reasons?), full overlap representing a single pathway, and overlap between these extremes suggesting distinct pathways/factors that mutually contribute to a *TargetCondition*.

4. Brute-BCRI

This section describes Brute-BCRI, a second realization of BCRI that uses Brute as the base rule induction system. We open with a description of the Brute system, which is less well known than C4.5. We then motivate the need for an interactive induction approach that interleaves input from a human analyst on decisions to backward chain, where human guidance is informed by prior knowledge and results of other forms of data analysis.

4.1 Brute

Given data and a *TargetCondition*, Brute conducts an extensive search for rules that predict *TargetCondition* with greater than a specified accuracy, and with a conjunctive antecedent that has greater than specified coverage. The search is limited by other parameters as well, such as setting the maximum number of conjuncts per antecedent. Brute will return all such rules that it finds under the constraints imposed by the parameters. Brute's search is systematic and non-redundant. Brute's search strategy is very similar to the search for association rules with the consequent fixed as *TargetCondition*.

4.2 Brute-BCRI as Interactive Induction

A lesson from machine learning of macro-operators is that if a system adds to the set of operators that can be applied during search, there must be corresponding improvement to the search control knowledge that selects from among the operators [32]. Without this latter property, problem solving behavior does not improve, but rather degrades.

The analogy with search control and macro-operator learning is apt in our case, because an analyst would be overwhelmed by the number of rules returned in a Brute search. For example, backward chaining on a condition in our domain using C45-BCRI might result in a couple of rules, but when using Brute-BCRI would lead to tens or hundreds of rules when coverage and accuracy thresholds were both set high (e.g., coverage greater than 70%). In the case of some conditions, C45-BCRI returned rules at about 10% coverage (see Figure 1). With a coverage threshold set to 10%, Brute-BCRI might uncover thousands of rules.

Our second realization of BCRI uses Brute as the base rule induction algorithm. Brute-BCRI, as we call it, is a collection of automated components, but the interaction of these components is not automated or otherwise spanned by a single software shell. The application of Brute-BCRI is highly interactive.

Interactive induction methods [3] interleave a human analyst and induction software in a collective induction process [8][9][1][5]. The intent of interactive induction is that human background knowledge compensates for too-little data – a human can bias the search in reasonable directions when data is insufficient to enable the induction software to make reliable search decisions autonomously.

In addition to prior knowledge, other data analytic tools such as clustering and data visualization can inform an expert's decisions on backward chaining in Brute-BCRI.

4.3 Rule Evaluation and Selection in Brute-BCRI

In C45-BCRI (Table 1) rules are evaluated and queued based on coverage. The general strategy of queuing rules based on an intra-rule measure like coverage (or accuracy) are not adequate to manage larger numbers of rules. Rather, there are several factors that control decisions to backward chain and decisions to pause and reflect on a hypothesized interaction.

1) *intra-rule measures* such as coverage and accuracy continue to play a role. A user can view Brute output by experimenting with various settings of accuracy and coverage

- pseudo-randomly,
- in response to prior knowledge (e.g., guided by settings of a rule discovered by C45-BCRI), or
- systematically, as in iterative deepening search, by gradually decreasing the coverage limit for example, in search of a manageable and non-trivial number of highly accurate rules

2) *inter-rule measures* such as the number of rules in which a gene condition participates as an antecedent. The more rules that a gene participates in, the greater its

apparent association to other genes. More generally, inter-rule overlap in antecedent conditions can inform decisions to backward chain.

3) *inter-rule population overlap* – in Section 3.4 we noted that looking at the overlap in data covered by rule antecedents can inform decisions to backward chain. For example, if two populations overlap little, but the antecedent conditions of the respective rules overlap significantly, then reflecting on the biological significance of the antecedent differences might be desirable.

4) *prior knowledge of feature interactions and links to outcome* – in our applications, conditions that involve genes with previously established links to cancer can be desirable subgoals for backward chaining. Rules involving such genes in antecedents or as outcome can implicate other genes that might be in a pathway for cancer.

5. Results with Brute-BCRI

Again, Brute can produce tens, hundreds, or thousands of rules for a given *TargetCondition* depending on search control settings. In this walkthrough of Brute-BCRI it was important to limit the search. We decided to focus on rules to predict high-risk only. There are hundreds of rules at 70% coverage with an accuracy greater than 80%. To reduce these to a manageable number, we limited our attention to rules with 100% accuracy at a minimum of 70% coverage. There are 17 rules satisfying this condition.

5.1 Focusing on genes at Level 1

Figure 3 shows the shared structure on the antecedents of the 17 level-1 rules. To limit search, we first select for those attributes that are present as antecedent conditions in more than two rules, i.e. high inter-rule coverage. Two gene expression attributes, SORT1 and TMSB4X, fulfill this criterion. Our method is to look for pathways associated with TMSB4X and the attributes that associate with it in rules. This step is repeated for SORT1.

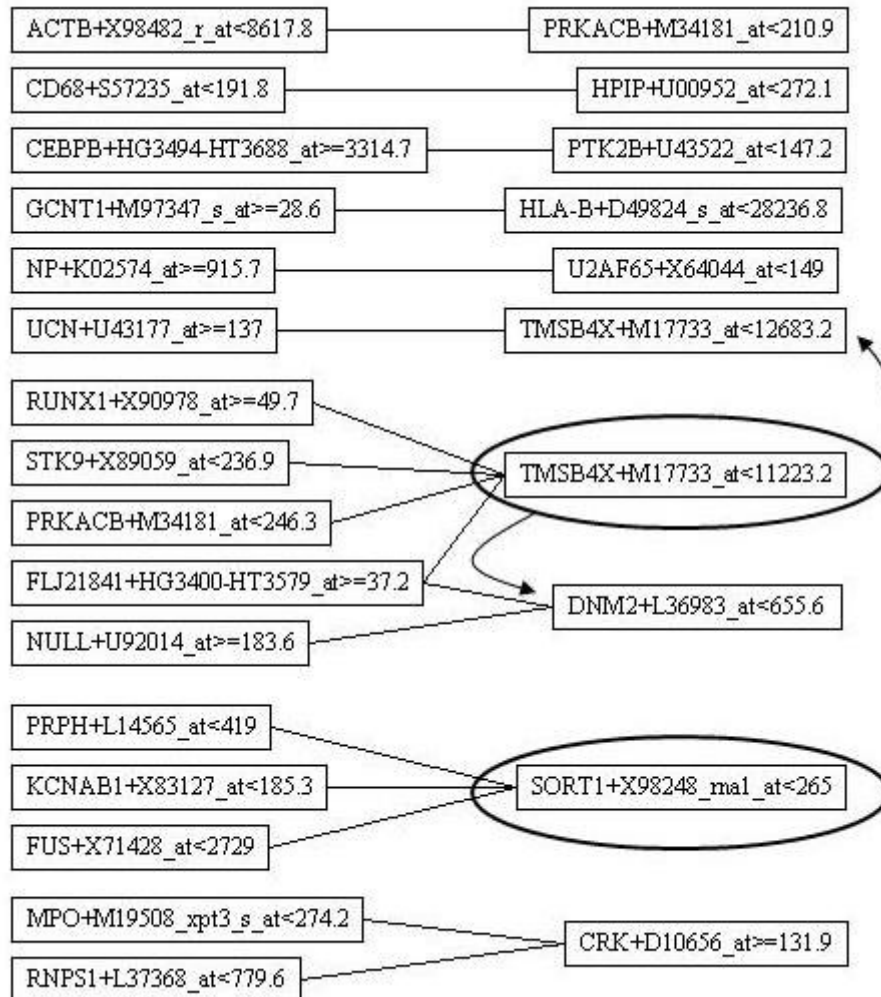


Figure 3. Antecedent connectivity in level 1 rules.

In addition to the SORT1 and TMSB4X being most prevalent in Brute's top rules for predicting high risk, there was prior knowledge that suggested the selection of at least TMSB4X for further investigation. Though it is not shown in Figure 1, the condition $TMSB4X < 10145.5$ had been found in 3 rules predicting high risk by C45-BCRI.

Finally, the data visualization of Figure 4 also suggests an interesting reason to investigate TMSB4X and SORT1 – this clustering by coverage, shows that the coverage of the intra-SORT1-rule similarity is high (i.e., rules placed adjacent in the right margin), that the intra-TMSB4X-rule similarity is high, and that similarity between these two sets is relatively low (i.e., the SORT1 rules are towards the top and the TMSB4X rules are at the bottom) – the dashed white rectangles highlight the differences in coverage.

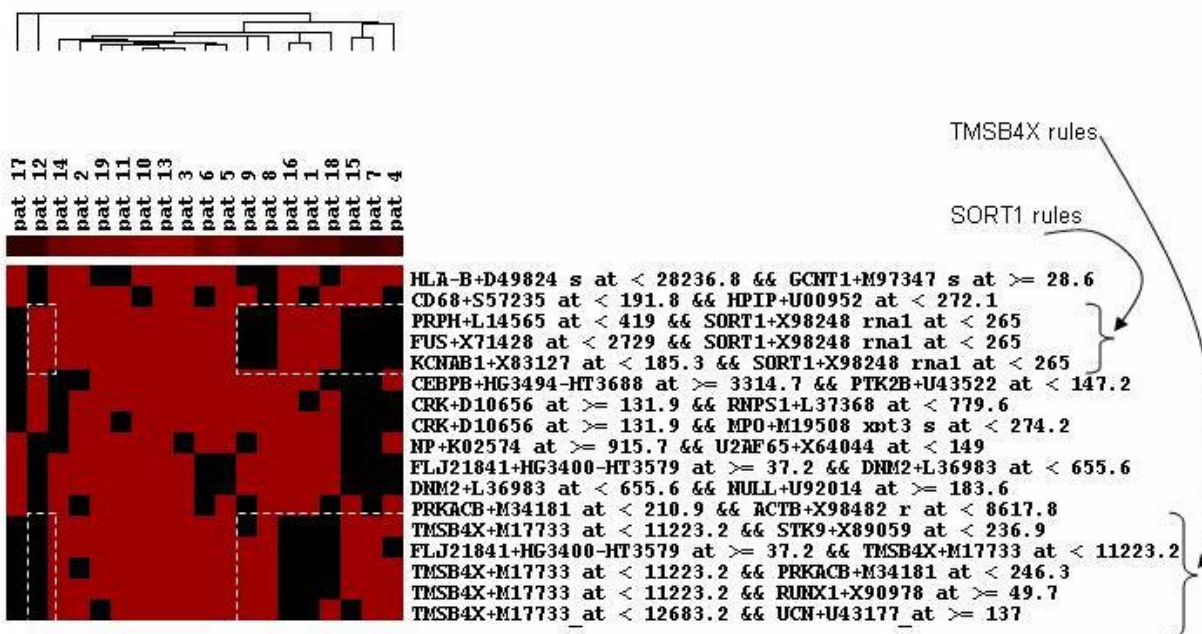


Figure 4. Clustering of rules based on data (patients) covered.

We found the association of FLJ21841 both with TMSB4X and with DNM2 interesting (see Figure 3), and included this with our TMSB4X pathways. Examination of Figure 4 shows that patient coverage for these DNM2 rules does not consistently overlap with the patient coverage for the TMSB4X rules, and we will discuss this later in the context of our level 2 analysis.

Collections of gene attributes that were included in rules of interest were compared against known gene and gene product interactions. We used the Metacore™ database of signaling and metabolic pathways [22], also reference above in the review of our C45BCRI implementation. The following results describe the results of these comparisons.

5.2 Level 1 analysis for TMSB4X

As described above, there are 7 genes initially associated with TMSB4X. The set of gene attributes includes UCN, PRKACB, FLJ21841, RUNX1, STK9, DNM2, and one labeled as NULL. The Null gene is excluded because of insufficient sequence information to identify it. Other genes, such as STK9, are excluded by Metacore™ because of insufficient information about its interactions. TMSB4X and UCN are not shown because they do not have sufficiently close interactions with the pathways connecting the remaining four genes: PRKACB (shown as PKA-cat), FLJ2141 (shown as Nestin), RUNX1, and DNM2 (shown as Dynamin 2).

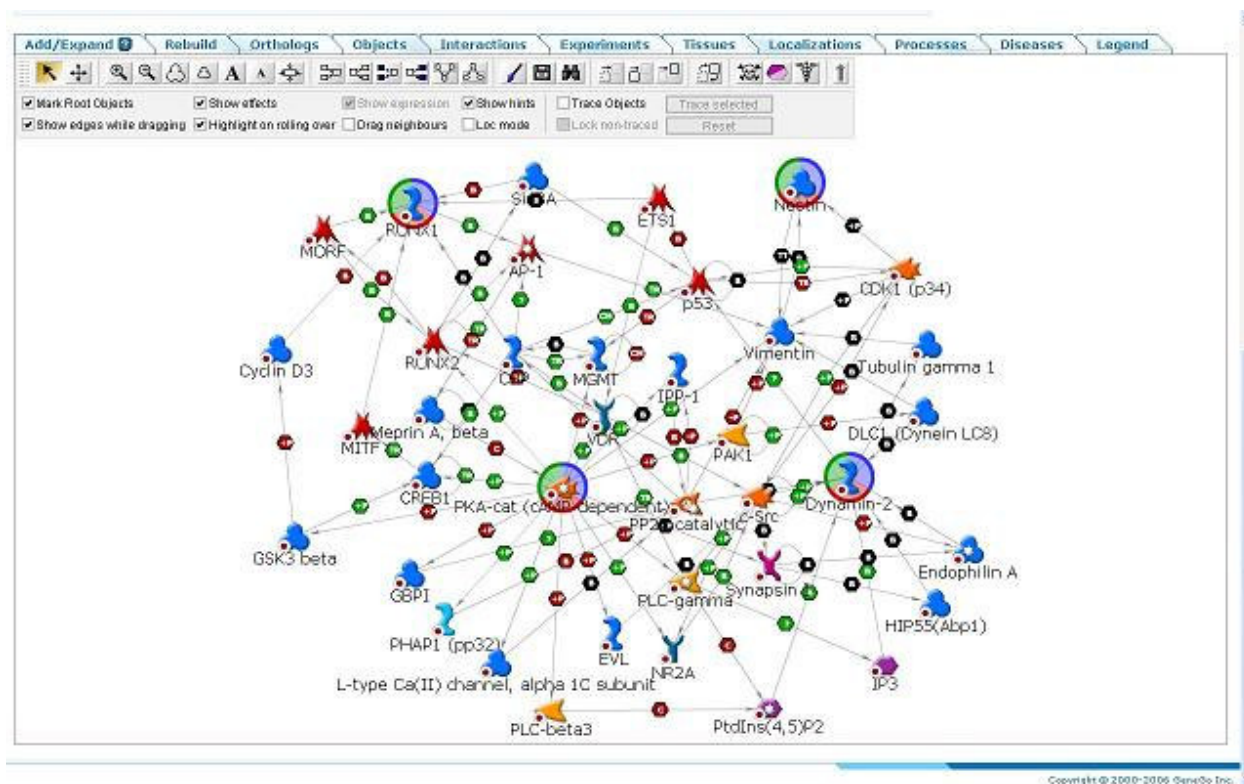


Figure 5. Metacore™ Pathway visualization of Level 1 TMSB4X Genes

We note with interest that TMSB4X does not connect with any of these gene expression attributes within our maximum distance of 3 used to draw the pathways. This contrasts with the results generated from our C45-BCRI implementation. (Unpublished C45 BCRI results for TMSB4X identified genes that were more closely associated with TMSB4x, such as alpha-1-antitrypsin.) We can hypothesize that TMSB4X is acting under an independent stimulus, and that these other pathways are ones that, when associated with TMSB4X, generate a more aggressive phenotype of lung adenocarcinoma.

Indeed, in this network, we see that two of the antecedents associated with TMSB4X are themselves central points in signaling networks that can lead to cancer. These are RUNX1 and PRKACB (or PKA-cat). We note that DNM-2 is also connected to p53, a pro-apoptotic gene that is often mutated in lung adenocarcinoma. The repeated implication of p53 using both C4.5 and Brute is important as a validator of the ability of BCRI with either of these rule induction engines to generate plausible hypotheses. Even more interesting though is the discovery here that c-SRC, a well known oncogene that is associated with aggressive disease phenotypes in ovarian and other epithelial tumors, may be associated with lung cancer. c-SRC has only been recently considered as having a possible role in lung cancer [36][15]. This discovery underlines the value of using a characteristic as opposed to a discriminant rule induction engine.

5.3 Level 1 analysis for SORT1

In Figure 3 we noted that SORT1 below a certain threshold associates with below threshold levels of PRPH (Peripherin), FUS, and KCNAB1 in three separate rules.

Pathway analysis of the gene attributes in these rules shows that SORT1 and PRPH (Peripherin) are connected by the Nerve Growth Factor Receptor (NGFR) and a Tyrosine Kinase receptor (TrkA). Sortilin binds to Nerve Growth Factor Receptor and enhances its activity, while TrkA that acts to modulate the activity of NGFR. The interaction of TrkA and peripherin is unknown.

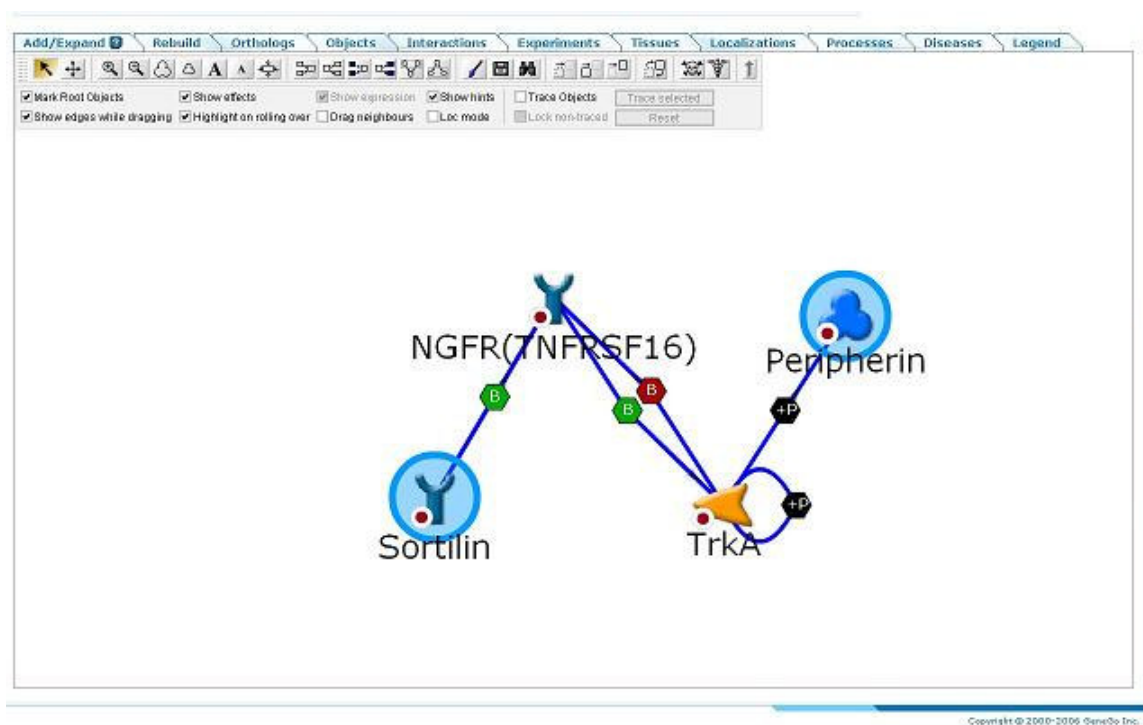


Figure 6. Metacore™ Pathway visualization of Level 1 SORT1 Genes

Although this pathway has been studied in the context of its importance in neuronal growth and differentiation, these factors have been more recently implicated in the context of cancers in non-neuronal cell types. For example, TrkA has been detected as highly expressed in lung carcinomas. In addition, the nerve growth factor that binds to the receptor modulated by TrkA appears to be highly expressed in several carcinomas, leading to the hypothesis that an interaction between the nerve growth factor and TrkA is directly involved in causing the proliferation of certain types of tumors. [16][29]

5.4 Level 2 Analysis for TMSB4X and Sort1

At level 2 we generate rules with a minimum 70% coverage and 100% accuracy that predict the TMSB4X and Sort 1 threshold conditions. There are 429 such rules (collectively involved 336 unique genes) generated that predict the TMSB4X condition, and 96 such rules (collectively involving 102 unique genes) that predict the Sort1 condition. These rules are too numerous to ascertain pathway distributions and hypothesize a manageable number of interactions or pathways. As we did at level 1 we analyzed these rules for high inter-rule coverage, i.e. the frequency with which any gene is used in the rules. We selected the four most frequently used genes to predict Sort 1 and the four most frequently used genes to predict TMSB4X.

5. 5 Level 2 Analysis for TSMB4X

ARHGDIB, STX4A, P5 and HRB. are the four most frequently used genes in rules that predict TMSB4X1. These were use as input to Metacore along with TMSB4X to generate Figure 7. TMSB4X and HRB (shown as RIP) are not sufficiently connected to appear in the pathway, emphasizing again that we seem to be discovering pathways operating under a different stimulus from TSMB4X, but acting in concert with it. P5 is also not shown because there are no documented interactions for it in the database.

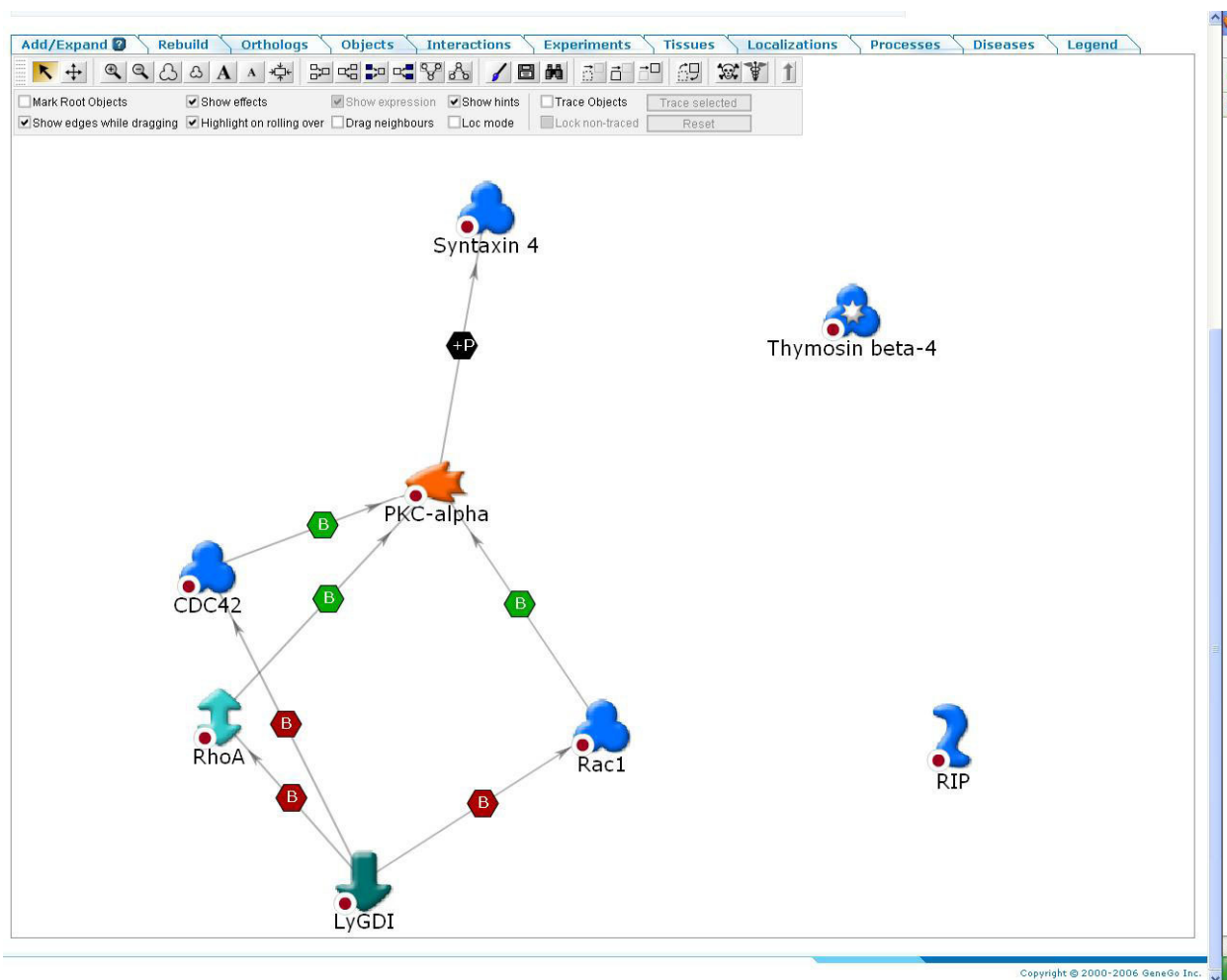


Figure 7. Metacore™ Pathway visualization of Level 2 TSMB4X Genes

RhoA, cdc42, and rac-1 are known to be involved in signaling mechanisms that contribute to cancer aggressiveness, and have recently been highlighted as candidates for molecularly targeted therapies in cancer [12]. Protein kinase-C has also been highlighted as a candidate target in treating lung cancer, and an association with lung adenocarcinoma has been specifically noted. [17]

5. 6 Level 2 Analysis for SORT!

RPS3, HU-K4, RNF5 and LAMP2. are the four most frequently used genes in rules that predict SORT1. These were used as input to Metacore along with SORT1 to generate Figure 8. LAMP2 and RPS3 do not appear because there are no documented interactions. HU-K4 is a member of the family of phospholipases (PLD), specifically PLD3. There are no documented interactions for PLD3. We included those for PLD1 and PLD2 as potentially representative.

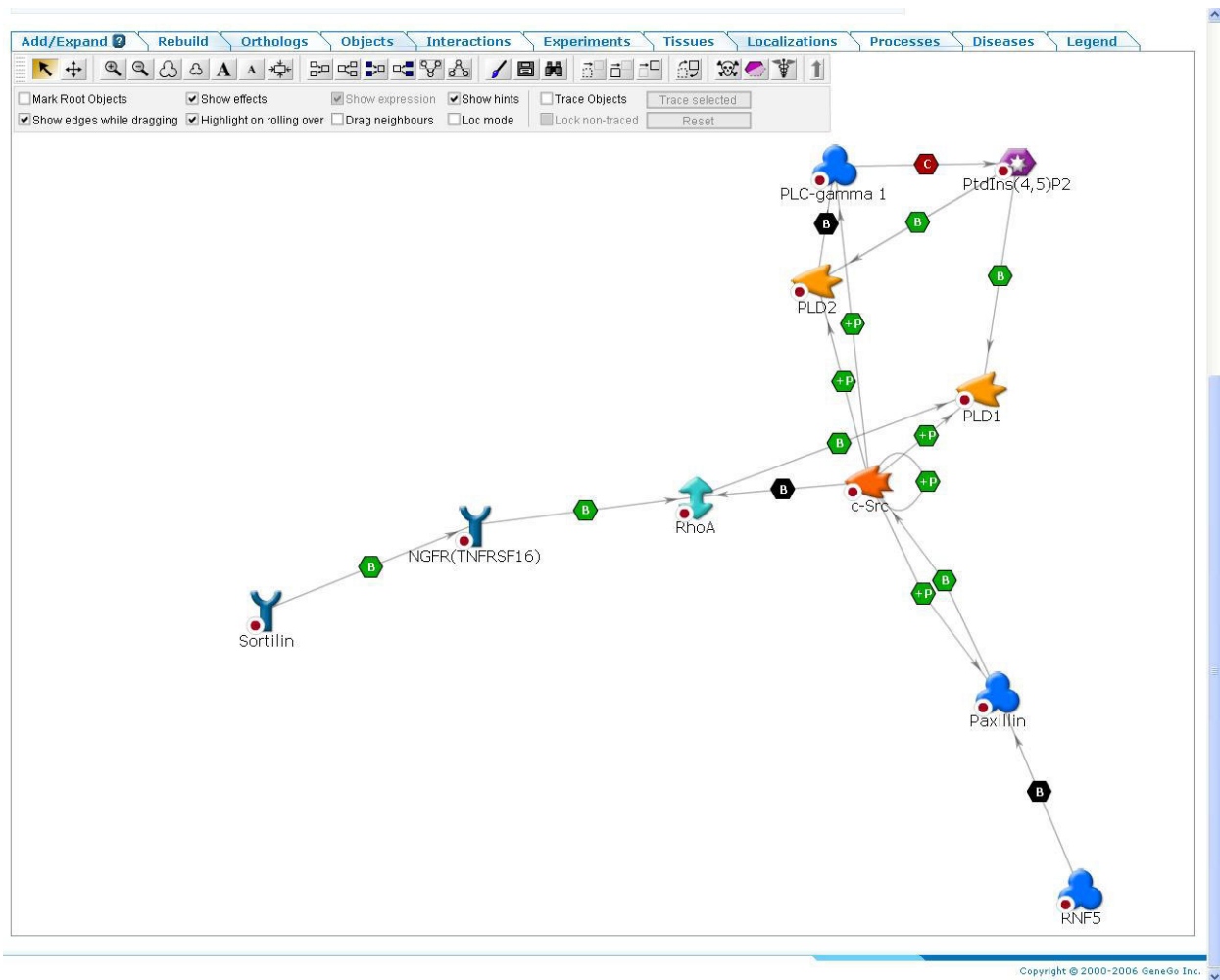


Figure 8. Metacore™ Pathway visualization of Level 2 TSMB4X Genes

Most interesting to note here is the re-appearance of c-src, a gene we discussed earlier in the context of TMSB4X. c-SRC was found to be connected to DNM-2, which we included in the level1 networks for TMSB4X because of a common association with FLJ21841 (Nestin) in the rule set in Figure 3. However, we noted at that time that patients covered by the DNM2 rules (see Figure 4) did not always overlap with patients covered by TMSB4X. Therefore, our results indicate that c-src is not connected with TMSB4X, but appears under independent stimulus and is associated with SORT1 and DNM2 rules. Therefore, when we discuss patients that are covered by TMSB4X rules and that are not covered by DNM2 or by SORT1 rules, we should exclude c-src from the hypothesized networks.

Notably RhoA appears again here. It is a signaling molecule that can be used by a number of different networks. In our level2 TMSB4X analysis, RhoA appeared as a signaling molecule in the PKC network. Here, it appears under the stimulus of the c-src network. Thus, it makes an interesting drug target that can be used to treat patients that otherwise appear mutually exclusive with respect to rule coverage.

5. 7 General Discussion of Results

Our discussion of the Brute-BCRI findings has focused on linkages between genes and cancer (for both level 1 and level 2 rules). In contrast, our initial motivation for BCRI was on hypothesizing gene interactions generally. We have not dismissed this original motivation, but it has taken a backseat to the more specific (not contradictory) goal of hypothesizing gene interactions in the context of cancer in the current study.

Our motivation in moving to Brute, a characteristic rule learner, was to introduce more OR nodes in the AND/OR network generated by BCRI. OR nodes are necessary to capture the heterogeneity that appears to underlie epithelial tumors – various molecular pathways can contribute to cancer risk. As expected, Brute seems to find pathways that were obscured when using C4.5's discriminant analysis. However, we need to exclude these Brute-discovered pathways from a similar analysis of low risk patients before we would strongly hypothesize that the Level 1 rules are particularly indicative of high risk. Our current appeal to biological explanation based on prior knowledge supports the links between genes implicated in Level 1 rules and high risk – subsequent comparisons to low risk rules would serve as validation.

Generally, a comparison between rules predicting contrast classes (TargetConditions) can be automated so as to identify those rules that are associated with one class over another. Of course, this comparison is already present in a discriminant learner such as C4.5. In addition, in a broad sense a discriminant learner like C4.5 is discriminating – it is limited to finding a few rules, and the goal is to find the best rules. For example, the rules that we examined from Brute did not include certain genes that were implicated by C4.5 (e.g., ELA2, Serpina1) we examined. This discussion suggests that it may be desirable to combine the features of characteristic and discriminative learners, using the former for breadth and the latter for “depth” by using contrast conditions to sharpen the focus of search.

6. Concluding Remarks

Knowledge discovery from data includes hypothesis generation and hypothesis testing. There is a paucity of formal, (semi-) automated methods for hypothesis generation about gene interactions. Gene expression microarray data, where hindered by the sparse number of samples, may lend itself better to hypothesis generation than prediction model building.

Regardless of the context, the goal of BCRI is to limit hypothesis generation to a manageable set of relationships for expert scrutiny. Experts can then assess the plausibility of rules (uncovering mechanisms) and the utility of rules (discovering clinical applications). We have investigated backward chaining rule induction to postulate governing gene networks in the context of lung cancer survival. BCRI can be applied in other domains as well, by initiating the process with different top level goals/classes.

Thus far, our contributions are (1) the definition of the BCRI task abstraction, (2) the implementation of two BCRI prototypes, (3) illustrations of hypothesis generation and data exploration with BCRI in the domain of cancer prognosis. An evaluation of the rules discovered suggests that conditioning the space of associations that is searched on some meaningful, overriding task/classification may better yield a rule set that is densely-populated with mechanistically plausible rules. Our wrapper prototypes of BCRI

are not optimal from a time-cost standpoint. Future work will look at other rule discovery systems as the core of BCRI, as well as more tightly couple the wrapper and core method to improve efficiency.

We can also use this mapping to bias induction [9][23] and supplement or revise, as needed, existing knowledge from induced knowledge [19] in a “systems biology” approach. To this effect, we have recently completed work using an iterative approach that uses the networks learned from BCRI combined with existing knowledge to discover modifiers to known signaling networks that qualify their relevance to outcome in lung cancer patients [11]. This approach could be especially effective in understanding the effectiveness of molecularly targeted therapies for lung cancer.

We will evaluate BCRI using other criteria. We realize that BCRI’s search through rule space will miss many associations. However, our goal is not to discover all the plausible rules that govern gene interactions, but to reduce them to a manageable number that is enriched for relevance to survival. Thus, we expect to have a high density of relevant rules using BCRI. One relevant comparison is against unsupervised association rule learners [18] in terms of the number of rules learned, and the density of “interesting” rules, though this latter criterion may be difficult to formalize in a comparative study.

We are also interested in comparing prediction accuracy of the rule network learned by BCRI against standard rule induction engines. To exploit the inference procedures of rule-based expert systems will require that we modify BCRI to produce rules that incorporate uncertainty (e.g., variance in antecedent thresholds, variance around accuracy point estimates of rules). Recent work by Waitman et al [33] provides guidance on how this can be done. Waitman et al [34] also show how similarity between rules can be computed, and rules can be visualized in terms of this similarity metric using multidimensional scaling. The identification of clusters in this visualization may provide additional information to filter rules for expert scrutiny.

In summary, the inference possibilities of a rule network constructed through backward-chaining rule induction are intriguing. We emphasize that it is not our goal in this paper to evaluate the accuracy of BCRI-induced rule networks for predicting clinical outcome. Rather, our primary goal is a limited, focused exploration of the associations between variables. Hypothesis generation in high density data with an effectively infinite number of combinations to examine requires an automated, computer tool for searching the plethora of possible interactions, and presenting selected possibilities (e.g., selected by heuristics involving accuracy and coverage, and heuristics relating to top-level outcomes of interest) to an expert analyst for comment and scrutiny.

Finally, the primary direction of future research is develop theories about productive hypothesis generation in the context of an interactive exploratory system, and use these theories to further inform semi-automation of theory development. Our exploration of hypothesis generation in cancer domains will help formalize the interactions between the various tools that are used in the current interactive Brute-BCRI approach. These tools include data and rule clustering and visualization. Each of these tools informs analyst decisions on backward chaining choices, but the current strategies are ad hoc. We want to formalize the strategies that are helpful in inductive exploration and consultation so as to inform a semi-automated approach to data exploration, hypothesis generation, and consulting prior knowledge.

Acknowledgements

We thank the reviewers for helpful and corrective comments. We also thank Mark Ross for his help in building and maintaining hardware on which BCRI was implemented. The research efforts of D.F., M.E., L.T., and Z.C. are supported in part by a National Institute of Health (NLM) grant (1R01LM008000) to M.E. and by funds from the Office of Research at Vanderbilt University Medical Center. L.F. is supported by a fellowship from the National Library of Medicine (5T15LM007450).

References

- [1] R. Bareiss, B. W. Porter and K. S. Murray, Supporting start-to-finish development of knowledge bases, *Machine Learning* 4 (1989), 259-283.
- [2] D. Beer, S. Kardia, C. Huang, et al, Gene-expression profiles predict survival of patients with lung adenocarcinoma, *Nature Medicine* 8 (2002), 816-824.
- [3] W. L. Buntine and D. A. Stirling, Interactive Induction, in: *MI-12: Machine Intelligence 12, Machine Analysis and Synthesis of Knowledge*, J. Hayes, D. Michie and E. Tyugu, eds, Oxford University Press, Oxford, UK, 1990, pp. 121-138.
- [4] L. Breiman, Bagging predictors, *Machine learning*, vol 26 (1996), 123-140.
- [5] P. Clark and S. Matwin, Using qualitative models to guide inductive learning, in: *Proc. Tenth Int. Machine Learning Conference (ML-93)*, P. Utgoff, ed, Kaufmann, CA, 1993 pages 49-56.
- [6] T. G. Dietterich and R. S. Michalski, A comparative review of selected methods for learning from examples, in: *Mach. Learn.*, New York, 1983, pp. 41-81.
- [7] M. Edgerton, D. Fisher, L. Tang, L. Frey, and Z. Chen, Data mining for gene networks relevant to poor prognosis in lung cancer via backward-chaining rule induction: An implementation in lung cancer research, *Cancer Informatics* (in press).
- [8] B. Evans and D. Fisher, Overcoming process delays with decision tree induction, *IEEE Expert* 9 (1994), 60-66.
- [9] B. Evans and D. Fisher, Decision tree induction to minimize process delays, in: *Handbook of Data Mining and Knowledge Discovery*, Oxford University Press, Oxford, UK, 2002, pp 874-881.
- [10] D. Fisher, M. Edgerton, L. Tang, L. Frey and Z. Chen, Searching for meaningful feature interactions with backward-chaining rule induction, in: *Proceedings of the Sixth Biennial Conference on Intelligent Data Analysis*, Madrid, LNCS 3646, Berlin Heidelberg: Springer-Verlag, 2000, pp. 86-96.
- [11] L. Frey, M. Edgerton, D. Fisher, L. Tang and Z. Chen, Using prior knowledge and rule induction methods to discover molecular markers of prognosis in lung cancer, *American Medical Informatics Association Symposium*, 2005, Washington DC.
- [12] G. Fritz and B. Kaina, Rho GTPases: promising cellular targets for novel anticancer drugs, *Curr Cancer Drug Targets* 6(1) (2006 Feb), 1-14.
- [13] A. Guffanti, Modeling molecular networks: a systems biology approach to gene function, *Genome Biol* 3 (2002), reports4031.
- [14] C. Huels, S. Muellner, H. Meyer, et al, The impact of protein biochips and microarrays on the drug development process, *Drug Discov Today* 7(18 Suppl) (2002), S119-24.

- [15] F. M. Johnson, B. Saigal, M. Talpaz and N. J. Donato, Dasatinib (BMS-354825) tyrosine kinase inhibitor suppresses invasion and induces cell cycle arrest and apoptosis of head and neck squamous cell carcinoma and non-small cell lung cancer cells, *Clin Cancer Res.* 11(19 Pt 1) (2005 Oct 1), 6924-32.
- [16] H. Koizumi, M. Morita, S. Mikami, E. Shibayama and T. Uchikoshi, Immunohistochemical analysis of TrkA neurotrophin receptor expression in human non-neuronal carcinomas, *Pathol Int.* 48(2) (1998 Feb), 93-101.
- [17] M. Lahn, C. Su, S. Li, M. Chedid, K. R. Hanna, J. R. Graff, G. E. Sandusky, D. Ma, C. Niyikiza, K. L. Sundell, W. J. John, T. J. Giordano, D. G. Beer, B. M. Paterson, E. W. Su and T. F. Bumol, Expression levels of protein kinase C-alpha in non-small-cell lung cancer, *Clin Lung Cancer* 6(3) (2004 Nov), 184-9.
- [18] H. Mannila, Association rules, in: *Handbook of Data Mining and Knowledge Discovery*, W. Kloggen and J. Zytkow, eds., Oxford University Press, Oxford, UK, 2002, pp. 344-348.
- [19] R. Mooney, Induction over the unexplained: Using overly-general theories to aid concept learning, *Machine Learning* 10 (1993), 79-110.
- [20] P. Murphy and M. Pazzani, Exploring the decision forest: an empirical investigation of Occam's Razor in decision tree induction, *Journal of Artificial Intelligence Research*, vol. 1 (1994), 257-275.
- [21] A. Nikitin, S. Egorov, N. Daraselia and I. Mazo, Pathway studio – the analysis and navigation of molecular networks, *Bioinformatics* 19 (2003), 2155-2157.
- [22] Y. Nikolsky, S. Ekins, T. Nikolskaya and A. Bugrim, A novel method for generation of signature networks as biomarkers from complex high throughput data, *Toxicol Lett.* 158 (2005), 20-29.
- [23] J. Ortega and D. Fisher, Flexibly exploiting prior knowledge in empirical learning, in: *Proceedings of the International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Francisco, 1995, pp. 1041-1047.
- [24] N. Provart and P. McCourt, Systems approaches to understanding cell signaling and gene regulation, *Curr Opin Plant Biol* 7 (2004), 605-9.
- [25] D. Pruitt and D. R. Maglott, RefSeq and LocusLink: NCBI gene-centered resources, *Nucleic Acids Res* 29(1) (2001), 137-140. URL: <http://www.ncbi.nlm.nih.gov/projects/LocusLink/> .
- [26] D. Pruitt, K. S. Katz, H. Sicotte et al, Introducing RefSeq and LocusLink: curated human genome resources at the NCBI, *Trends Genet* 16(1) (2000), 44-47, URL: <http://www.ncbi.nlm.nih.gov/projects/LocusLink/> .
- [27] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, 1993, URL: <http://www.rulequest.com/> .
- [28] M. Rebhan, V. Chalifa-Caspi, J. Prilusky, et al, GeneCards: encyclopedia for genes, proteins and diseases, Weizmann Institute of Science, Bioinformatics Unit and Genome Center (Rehovot, Israel), 1997, URL: <http://bioinformatics.weizmann.ac.il/cards> .
- [29] A. Ricci, S. Greco, S. Mariotta, L. Felici, E. Bronzetti, A. Cavazzana, G. Cardillo, F. Amenta, A. Bisetti and G. Barbolini, Neurotrophins and neurotrophin receptors in human lung cancer, *Am J Respir Cell Mol Biol.* 25(4) (2001 Oct), 439-46.
- [30] P. Riddle, R. Segal and O. Etzioni, Representation Design and Brute-force induction in the Boeing Manufacturing Domain, *Applied Artificial Intelligence* 8 (1994), 25-147.

- [31] E. Shortliffe, R. Davis, S. Axline, et al, Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system, *Comput. Biomed. Res* 8 (1975), 303-320.
- [32] J. Shrager, T Hogg and B. A. Huberman, A graph-dynamic model of the power law of practice and the problem-solving fan effect, *Science* 242 (1988), 414-416.
- [33] L. R. Waitman, D. Fisher and P. King, Bootstrapping rule induction, in: *Proceedings of the IEEE International Conference on Data Mining*, IEEE Computer Society Publications Office, Los Alamitos, CA, 2003, pp. 677-680.
- [34] L. R. Waitman, D. Fisher and P. King, Bootstrapping rule induction to achieve and increase rule stability, *Journal of Intelligent Information Systems*, in press.
- [35] A. D. Weston and L. Hood, Systems biology, proteomics, and the future of health care: toward predictive, preventative, and personalized medicine, *J Proteome Res* 3 (2004), 179-96.
- [36] R. Zheng, S. Yano, Y. Matsumori, E. Nakataki, H. Muguruma, M. Yoshizumi and S. Sone, SRC tyrosine kinase inhibitor, m475271, suppresses subcutaneous growth and production of lung metastasis via inhibition of proliferation, invasion, and vascularization of human lung adenocarcinoma cells, *Clin Exp Metastasis* 22(3) (2005), 195-204.