4-2011

Learning feature dependencies for noise correction in biomedical prediction

Ghim-Eng YAP
Institute for Infocomm Research, Singapore

Ah-Hwee TAN
Nanyang Technological University

Hwee Hwa PANG
Singapore Management University, hhpang@smu.edu.sg

DOI: https://doi.org/10.1137/1.9781611972818.7

Citation

YAP, Ghim-Eng; TAN, Ah-Hwee; and PANG, Hwee Hwa. Learning feature dependencies for noise correction in biomedical prediction. (2011). *11th SIAM International Conference on Data Mining 2011: Mesa, Arizona, USA, 28-30 April 2011: Proceedings*. 71-82. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/3661

Learning Feature Dependencies for Noise Correction in Biomedical Prediction

Ghim-Eng Yap
Data Mining Department,
Institute for Infocomm Research,
1 Fusionopolis Way, 21-01, Connexis,
Singapore 138632
geyap@i2r.a-star.edu.sg

Ah-Hwee Tan
School of Computer Engineering,
Nanyang Technological University,
Nanyang Avenue,
Singapore 639798
asahtan@ntu.edu.sg

Hwee-Hwa Pang
School of Information Systems,
Singapore Management University,
80 Stamford Rd,
Singapore 178902
hhpang@smu.edu.sg

Abstract

The presence of noise or errors in the stated feature values of biomedical data can lead to incorrect prediction. We introduce a Bayesian Network-based Noise Correction framework named BN-NC. After data preprocessing, a Bayesian Network (BN) is learned to capture the feature dependencies. Using the BN to predict each feature in turn, BN-NC estimates a feature's error rate as the deviation between its predicted and stated values in the training data, and allocates the appropriate uncertainty to its subsequent findings during prediction. BN-NC automatically generates a probabilistic rule to explain BN prediction on the class variable using the feature values in its Markov blanket, and this is reapplied as necessary to explain the noise correction on those features. Using three real-life benchmark biomedical data sets (on HIV-1 drug resistance prediction and leukemia subtype classification), we demonstrate that BN-NC (1) accurately detects the errors in biomedical feature values, (2) automatically corrects for the errors to maintain higher prediction accuracy over competing methods including Decision Trees, Naive Bayes and Support Vector Machines, and (3) generates probabilistic rules that concisely explain the prediction and noise correction decisions. In addition to achieving more robust biomedical prediction in the presence of feature noise, by highlighting erroneous features and explaining their corrections, BN-NC provides medical researchers with high utility insights to biomedical data not found in other methods.

1 Introduction

Medical doctors and biomedical researchers are increasingly interested in adopting machine learning tools to mine high-dimensional biomedical data sets for critical tasks including personalized medicines and population-wide disease screening. An example is the use of serum proteomic data from mass spectrometry for ovarian cancer screening, where the classifier has to be trained on hundreds of thousands of mass-to-charge (M/Z) intensities [1]. Another scenario uses DNA microarray expressions to differentiate the acute leukemia subtypes, where again many thousands of expressions are involved [2].

Where such massive volumes of genotypic features have to be monitored in a lab environment, issues such as electronic noise, chemical contamination, hybridization, and microarray spots irregularities often arise undetected. As a result, noise or unexpected errors in the recorded feature values are common [3, 4, 5]. Such errors are even more rampant if the feature values have to be manually-curated into data repositories, such as in the case of the single-nucleotide polymorphisms (SNPs) being used in genome wide association (GWA) studies [6].

To address the noise in biomedical data, the classifier has to identify the erroneous features and take into account the uncertainty in each presented value to make the right prediction. It is also necessary for the prediction to be explained to users, especially since biomedical prediction usually has serious consequences (e.g. cancer screening). Current methods for biomedical prediction such as decision tree and support vector machines cannot detect nor correct for noise, and this can lower the accuracy of the biomedical prediction significantly [7, 1].

In many cases, dependencies can be observed among the biomedical features, and this input redundancy can cancel out some of the noise. Indeed, our earlier research has shown that by effectively exploiting the captured dependencies among the biomedical features in a Bayesian network (BN) [8], we can achieve robust ovarian cancer detection using real-world serum proteomic features [1]. This is possible because whilst classifiers such as the decision tree, naive Bayes and support vector machine are only effective at capturing associations from features to the class variable, the probabilistic causal model of BN encapsulates the dependencies between features. Nevertheless, noise corrections cannot take place or are ineffective as long as the identities of the erroneous features and their probabilities of error remain undetected, and any explanation for BN prediction based on the wrongly recorded feature values would only confuse the users.

In this paper, we introduce the Bayesian Network with Noise Correction (BN-NC) framework. Capturing feature dependencies by learning a BN, BN-NC uses the BN to predict each feature in turn, and estimates its error rate using the proportion of mismatches between its predicted and stated values in the training data. This error estimate is presented to BN as the uncertainty in that feature, so that BN can accord the right amount of confidence to that feature's stated values in subsequent predictions. A probabilistic rule is generated to explain each BN prediction using the feature values in the class variable's Markov blanket. Where a feature's value during prediction differs from its stated value, BN-NC explains the correction on that feature value in a recursive manner.

Our related work on ovarian cancer detection has, to a preliminary extent, demonstrated the effectiveness of exploiting feature dependencies for accurate prediction. Here, a much more rigorous empirical analysis is presented to substantiate some of the claims in BN-NC that are important for noisy biomedical prediction. Specifically, we systematically answer the following questions:

- Suppose the true error rates of biomedical features are known, can BN-NC accurately identify all those erroneous features and estimate their error rates?

- In the presence of noise, does BN-NC outperform standard BN, decision tree, naive Bayes, and support vector machines, all of which are the popular machine learning tools for biomedical prediction?

We conduct a controlled experiment by introducing different severity of feature noise into the HIV-1 data set from the Stanford HIV Drug Resistance Database [9]. This pseudo-natural setup allows us to control the size of noise in specific features to verify whether the different levels of added noise are correctly detected by BN-NC, and furthermore, how accurate is BN-NC compared to decision tree, naive Bayes and support vector machine.

- The BN-NC explanation procedure uses features in the class variable's Markov blanket to explain BN prediction. In the learned BNs, do the features in the class variable's Markov blankets really possess known biomedical association to the class variable?

We verify that the BN learned using the HIV-1 data contains known resistance mutations as the class variable's Markov nodes by learning the BN models from many independent data partitions and summarizing the frequencies with which known mutations (e.g. TAMs) appear within the class node's learned Markov blanket.

- DNA microarray is a prevalent source of data used for biomedical prediction today. Does feature noise really exist in real-world microarray data sets, and how does the BN-NC framework perform for them?

Finally, we evaluate the prediction accuracy of BN-NC using two real benchmark DNA microarray data sets for acute leukemia subtype classifications. The empirical results demonstrate that BN-NC effectively detects naturally-occurring noisy features, and by appropriately accounting for the noise, performs comparably or better than classifiers including DT and SVM, while additionally giving valuable comprehension to BN's predictions.

The rest of this paper is organized as follows. Section 2 introduces the proposed Bayesian Network with Noise Correction (BN-NC) framework. Section 3 elaborates on the specific problems of HIV-1 drug resistance prediction based on DNA mutations and acute leukemia subtype classification using gene expressions, our preparation of the corresponding benchmark data sets, as well as the evaluation measure and baseline methods used for comparison in the experiments. Section 4 presents and discusses the results and Section 5 concludes this paper.

2 Bayesian Network with Noise Correction (BN-NC) Framework

Algorithm 1 summarizes the proposed Bayesian network with Noise Correction (BN-NC) framework, which systematically brings the noisy biomedical data set through the steps of preprocessing, model extraction, noise discovery, and diagnosis with explanation (as visualized in Figure 1). We describe the details of these steps below.

2.1 Step 1 - Select Useful Features Our data preprocessing focuses on selecting the useful features and discretizing them for BN model learning. We implement the entropy-based discretization and feature

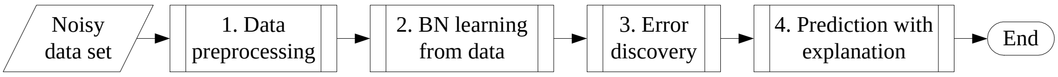


Figure 1: The Bayesian Network with Noise Correction (BN-NC) framework.

Algorithm 1 The Complete BN-NC Framework

```
Input: Training data (train), test data (test), and error threshold (t).
Output: BN learned from train, erroneous features (F) with error rates (R), inferences (I) on test, and explanations (E) for I.

  Step 1: Select Useful Features
  Let $A$ be the set of attributes in train.
  for each attribute $a \in A$ do
    if $a$ does not separate the target classes then $A = A - a$.

  Step 2: Capture Feature Dependencies
  Select the top-$k$ most-informative features (top_A).
  Learn $BN$ from train based on top_A.

  Step 3: Estimate Error Rates of Features
  Identify $F$ and estimate $R$ values using Algorithm 2.

  Step 4: Predict with the Noisy Features
  for each test case $c_i \in test$ do
    Estimate likelihoods of $F$, based on $R$ and case $c_i$.
    Enter $c_i$ into $BN$ to obtain inference $i$; $I = I + i$.
    Generate explanation $e$ for inference $i$; $E = E + e$.

  return $BN$ and the sets $F$, $R$, $I$, and $E$.
```

selection technique from Fayyad and Irani [10], which has been applied successfully, e.g. in [1], to preprocess other bio-data. The Fayyad-&-Irani technique combines the entropy-based splitting criterion of the C4.5 decision tree [11] with the minimum description length (MDL) stopping criterion. It finds the optimal cutting point for every feature so as to maximize its separation of the classes. Features without cutting points are discarded, effectively reducing the dimensions while converting the continuous features into discrete features to enable learning of discrete BN models. Other preprocessing methods can also be applied as part of this step, e.g. [12].

2.2 Step 2 - Capture Feature Dependencies We sort the discretized features in decreasing order of information gain, which measures the reduction in entropy obtained by splitting the training data on each feature, and learn a BN model based on the top-k most-informative features (the value of k is automatically selected via cross-validation using the training data). This enables BN-NC to capture the feature dependencies, something that methods such as SVM are unable to do. In this work, the CaMML BN learning program [13] is adopted. CaMML stochastically compares all the possible models to maximize a minimum message length (MML) posterior metric [13]. For each visited model, it computes a representative model and sets the representative's posterior as the sum posterior of its members. Each aggregated posterior approximates the probability that the true model lies within the MML equivalence class of a representative, so the representative with the largest MML posterior is the best model. In our experience, accurate biological causal models can usually be found based on fewer than thirty informative attributes (e.g. [1]), and CaMML learns a BN of this size in just a few minutes. Alternative methods like constraint-based BN model learning [13] can also be applied in this step.

2.3 Step 3 - Estimate Error Rates of Features

A data feature is potentially erroneous or noisy if it carries its observations with some error probability. An error rate or probability of e for a feature f implies that observations for f are wrong e*100% of the time. In practice, we do not know how many and which features are erroneous, so we need to be able to discover multiple erroneous features, and to estimate the error rate for each of them. As presented in Algorithm 2, the error discovery procedure takes in the training cases and also the corresponding learned BN. Predicting with the BN and using the proportion of misclassified training cases to estimate each feature's error rate, the procedure identifies the feature that is most likely to be erroneous, based on the intuition that the noisiest feature would be the one that is least predictable by the learned BN.

For each training case, the procedure enters all the evidence except for the feature being investigated and exploits the captured causal dependencies between that feature and the rest of the network to predict its value. After the most likely erroneous feature is identified, the procedure enters that feature's value for each training case as a *likelihood finding*, or a *soft evidence*, while it searches for the *next* most likely erroneous feature in a similar manner. The process is repeated to find the third most erroneous feature, and so forth - it terminates

Algorithm 2 **Error Discovery in BN-NC**

```
Input: Training data ($train$), $BN$ from $train$, and error threshold ($t$).
Output: Erroneous features ($F$), and their corresponding error rates ($R$).
  Step 1: Identify the top erroneous feature $f_{top}$.
  for each feature $f_i$ do
    for each record $\in$ $train$ do
      Cover-up $f_i$ and predict its value using $BN$.
    $P_{err}(f_i)$ = fraction of $train$ that $f_i$ is misclassified.
  $f_{top} = argmax(P_{err}(f_i))$, $max\_err = P_{err}(f_{top})$.
  if $max\_err < t$ then return $F$ and $R$ as empty sets.
  else $F = F + f_{top}$, $R = R + max\_err$.
  Step 2: Identify the rest of sets $F$ and $R$.
  $min\_err = P_{err}(f_{top})$.
  while $\exists$ feature $\notin F \wedge min\_err \geq t$ do
    for each feature $f_i \notin F$ do
      for each record $\in$ $train$ do
        Estimate likelihoods $L$ for feature values in $F$.
        Cover-up $f_i$; predict its value with $BN$ and $L$.
      $P_{err}(f_i)$ = fraction of $train$ with $f_i$ misclassified.
    $f_{next} = argmax(P_{err}(f_i))$, $max\_err = P_{err}(f_{next})$.
    $F = F + f_{next}$; $R = \emptyset$.
    for each feature $f_i \in F$ do
      $R = R + P_{err}(f_i)$.
    $f_{least} = argmin(P_{err}(f_i))$, $min\_err = P_{err}(f_{least})$.
  if $min\_err < t$ then $F$=$F$-$f_{least}$, $R$=$R$-$P_{err}(f_{least})$.
  return the non-empty sets $F$ and $R$.
```

only when the error rates for all the features have been estimated, or until the minimum detected error rate falls below a particular threshold. The error threshold allows experts to ignore trivial degrees of natural randomness in the features' values; in our experience, a suitable error threshold would be 0.1 or less, so that as many of the erroneous features in the data are identified as possible.

2.4 Step 4 - Predict with the Noisy Features

2.4.1 Likelihoods Estimation We expect the test data to be noisy too, with a noise characteristic similar to the training data. Inputting the features' values as *specific findings*, or *hard evidence*, is very likely to result in wrong predictions if those findings are incorrect. The BN allows us to specify such potentially noisy findings as *likelihoods* or *soft evidence* to reflect their uncertainties.

To illustrate, suppose a feature f has three possible values in V:$\{0, 1, 2\}$, and it is observed to have one of these values (this observation is hereafter denoted as o). The *likelihoods* of o are the conditional probabilities of o given the true value v_f of feature f, i.e., likelihoods of o are $\{\text{prob}(o|v_f = 0), \text{prob}(o|v_f = 1), \text{prob}(o|v_f = 2)\}$.

Now suppose o is $\{f = 1\}$, and f carries an error rate of e (rate e could be approximated from the training data using Algorithm 2). The likelihoods for o are then $\{P(0) * e, (1.0 - e), P(2) * e\}$, where $P(v)$ is the prior probability of value v (approximated by the proportion of the training data for which v was observed). This is because the likelihood of observing $\{f = 1\}$ when $v_f = 0$ or $v_f = 2$ is the probability that one of these other values is present in the example but we make an error, whilst the likelihood of observing $\{f = 1\}$ when $v_f = 1$ is the probability of not making an error. We generalize this likelihoods estimation below, where entering a error rate of 0.0 is akin to entering the evidence with full certainty:

Likelihood of o for the observed value $v = (1.0 - e)$,
Likelihood of o for value $v' \in \{V - v\} = P(v') * e$.

2.4.2 Explaining the BN Inferences Our BN-NC framework seamlessly integrates the recent Explaining BN Inferences (EBI) procedure [14] to automatically generate explanations for BN inferences. EBI explains the value of a target node using just the contextually influential nodes in its Markov blanket (the target node's parents, children, and spouses). This is necessary and sufficient because conditional independence implies that the Markov values form a minimal set that fully explains the inferences. To simplify its explanations, EBI exploits the context-specific independence reflected in the target node's conditional probabilities. Working back from the target node, EBI shows the derivation of each intermediate variable in terms of their own respective Markov nodes, thereby explaining how missing and noisy evidence values are corrected during inference. EBI's ability to explain the noise corrections during BN prediction is not found in any other methods, e.g., [15].

The EBI procedure consists of three key steps. First, EBI restructures the local dependencies around the target node (or the class variable) via a series of arc reversals, such that its Markov nodes become its parents while maintaining the joint distribution. The resulting conditional probability table (CPT) of the target node describes its probability distribution over all its Markov value combinations. Next, EBI simplifies the rules that it would generate, by learning from the target CPT a decision tree (DT) that captures the context-specific independence among the Markov nodes for each target value. Finally, during the inference, EBI compares the assigned Markov values against the DT for the predicted target value, and explains the prediction in terms of the nodal values along the DT path for the current context.

3 Experiment Setup

3.1 Problem Domains

3.1.1 HIV-1 Drug Resistance Prediction

HIV-1 drug resistance arises when, after the consumption of a drug by a patient, the amino acids in parts of her HIV-1 virus *mutate* to increase its resistance against that drug. Therapies that administer antiretroviral drugs for which the patient's virus is highly resistant are ineffective; not only might the virus develop new mutations making it less susceptible to similar drugs (a condition known as *cross-resistance*), the patient suffers unnecessarily from the ineffective antiretroviral drugs' adverse side-effects.

Recently, a variety of computational approaches have been used to predict a patient's drug resistance based on mutations, including linear regression [16] and support vector machines (SVM) [17]. In [17], Saigo et al. commented that no existing methods "can achieve high accuracy and good interpretability at the same time"; as such, they developed an *itemset-boosting* technique which extracts linear regression rules using mutation associations (complex features) as predicates. Itemset-boosting assumes that predicates (complex / singular features) combine linearly to predict the resistance outcome. There is no knowledge of how those features *interact* to influence the drug resistance, nor how the singular features in each mutation association are related.

To better understand the roles played by the drug-resistance mutations, other works have turned to the Bayesian network (BN) [18]. Our approaches are significantly different in a number of ways. First, the works of Deforche et al. [18] tried to understand the *evolution* of drug resistance during treatments. In contrast, we aim to discover a model of salient mutations that predicts the virus's resistance to a specific drug, or a drug class; mutations need not be the *in vivo* reactions to previous consumption of the drug, but instead can be results of site-directed mutagenesis where resistance levels are determined *in vitro* [16]. So, while the prior BN works focused on learning to interpret mutational responses by HIV-1 *after* taking a drug, we are the first to analyze BN's accuracy for the *prediction* of HIV-1 resistance.

While Deforche et al. [18, 19] focused on the pharmacological effects of drugs in the protease inhibitor (PI) family (e.g. nelfinavir), we opt to discover a probabilistic model of mutations that can predict a patient's level of resistance against the nucleotide reverse transcriptase inhibitor (NRTI) drug known as Tenofovir (TDF). Our study on TDF is motivated by the knowledge that for NRTIs, only combinations of multiple mutations can increase HIV-1's resistance to drugs, and it has been found that salient large mutation associations are more common for NRTIs than other drugs [17].

3.1.2 Acute Leukemia Subtype Classification

More than 40,000 new cases and 20,000 leukemia-related deaths were expected in the United States of America alone last year [20]. Leukemia, or cancer of the blood, has heterogeneous subtypes with diverse responses to different forms of therapies [21]. Misdiagnosis result in inappropriate therapies that can lead to excess toxicity and low survival rate [2], so recognizing the leukemia subtype is a critical step in the treatment of leukemia.

Microarray expressions are known to be noisy [3, 4, 5], and unknown errors in the expression or feature values can misguide standard classification methods to make wrong predictions. It is therefore important that our noise correction framework can reliably identify the errors among the observations and effectively correct for them during predictions. In addition to a highly robust accuracy, a desirable leukemia subtype classification method should also be able to *explain* to the user how important genes had interacted to derive the diagnosis. These modelled behaviors may then be verified clinically by experts. Unfortunately, most existing methods give "black-box" classifiers which do not provide users with sufficient comprehension on the classification outcomes.

3.2 Data Source and Data Preparation

3.2.1 HIV-1 Drug Resistance Prediction

It is hard to ascertain using available data sets that a given approach can indeed find those features that are truly noisy. This is because no one knows which of the feature values are wrong (if so, these would have been corrected). In view of this, it is necessary to demonstrate through simulated experiments that a process finds noisy features. The HIV-1 domain was chosen because mutation associations influencing resistance to each drug are relatively established, allowing us to objectively verify whether BN learning can recover these associations. In addition, by explicitly introducing different rates of errors to mutations, we can objectively compare the performance of BN-NC against competing methods within identical setups of noisy environment.

We use the real-life HIV-1 data set from the Stanford HIV Drug Resistance Database [9] for our controlled experiments, where different degrees of errors are injected into the mutations in the isolates' DNA sequences of patients and their level of resistance to the nucleotide reverse transcriptase inhibitor (NRTI) drug known as Tenofovir (TDF) are predicted. We use the complete mutation set prepared by Rhee et al. [16]. In the data, mutations are defined as amino acid differences from subtype B consensus wild-type sequence [9]. Each mutation is represented with a position index flanked by the acid's subtype before and after mutation. For example, "M41L" refers to mutation of amino acid M into amino acid L at position 41 of the sequence.

There are three discrete levels of drug resistance ("Susceptible", "Intermediate" and "High-level"). The data set for drug TDF comprises 353 isolates (instances), of which 70% are labelled "Susceptible", 18% are "Intermediate", and the remaining 12% are "High-level". There are a total of 348 mutations (features) in this data. The errors are introduced into the data by randomly flipping the mutation values in the samples according to the intended error rate. For example, to introduce an error rate of x into mutation m, we randomly flip x*100% of m's values among the data samples.

In all the HIV-1 experiments, we performed ten independent rounds of stratified five-fold cross validation (10x5-fold cross-validation) to ensure better statistical stability [22]. In each round, data samples are randomly partitioned into five equal-sized parts, where each part contains approximately the same distribution of class labels as the overall data. We then perform five independent validations using this partitioning, each time leaving out one of the five parts for testing and using the remaining four parts to train. A total of fifty such validations are performed to get the average accuracy.

3.2.2 Acute Leukemia Subtype Classification By introducing errors into the previous HIV-1 data set, we would have effectively addressed our first three questions stated in Introduction (Section 1). In this second series of experiments, we will evaluate whether the BN-NC framework performs accurately on DNA microarray data given their natural noise characteristics, using two publicly-available acute leukemia gene expression data sets: the ALL/AML data set [2] provided by the Broad Institute Cancer Program, and the Pediatric ALL data set [23] provided by the Saint Jude Children's Research Hospital.

The ALL/AML data set classifies acute leukemia tumor samples into Acute Lymphoblastic Leukemia (ALL) and Acute Myeloid Leukemia (AML) using gene expressions [2]. The publicly-available data set comprises 38 training cases (with 27 ALL and 11 AML cases) and 34 test cases (20 ALL and 14 AML cases). For comparison with prior results, we adopt these original data partitions in our experiments. There are a total of 7,129 genes (i.e., features) in the ALL/AML data set.

The Pediatric ALL data set classifies 327 Acute Lymphoblastic Leukemia (ALL) tumor samples from children into six subtypes based on their gene expressions [23]. The training set comprises 215 examples, and another 112 examples are provided for testing. Depending on which one of the six subtypes is to be classified, an example in the data is either labelled as "belonging" or "not belonging" to that ALL subtype [23]. In our experiments, we adopt the original data partitioning in order for direct comparison with the previous results.

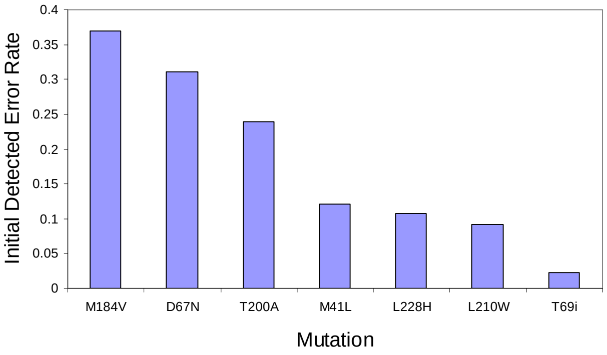


Figure 2: The initial error rates detected for TDF.

3.3 Performance Evaluation and Baselines We used prediction accuracy, or the proportion of test instances that are correctly predicted, as our performance evaluation measure. In other words,

$$accuracy = \frac{||correct||}{||test||} \tag{3.1}$$

where $||correct||$ and $||test||$ denote the number of test instances whose labels are correctly predicted and the total number of test instances, respectively.

Within identical setups of noisy environment, we compare performance of standard BN against BN aided by noise correction (BN-NC), using the J4.8 (or C4.5) decision tree (DT) [11], naive Bayes (NBayes) [24] and support vector machine (SVM) classifiers as the baseline methods (these baseline methods are available within the WEKA software [25]; we use the WEKA implementations and its default settings for all our experiments). The C4.5 DT classifier, which does not capture feature dependencies, is a suitable baseline because it has been shown to perform as well as standard BN when there is no feature noise [26]. Similarly, NBayes, which assumes independence among all data features, is suitable because it has been shown to outperform other much more complicated machine learning methods [27]. SVM is chosen as a baseline due to its generalization ability and popularity in medical classification research [28, 29]. The effectiveness of considering feature dependencies under BN-NC would be well supported if it performs better in presence of noise than basic BN, DT, NBayes and SVM.

4 Results and Discussions

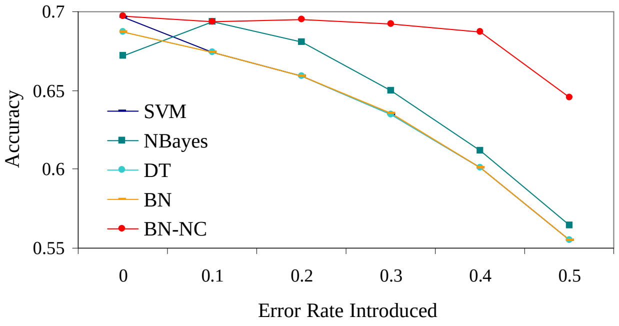


Figure 3: Test prediction accuracies for TDF (the plots for SVM, DT and BN overlap as the lowest line). BN-NC (the highest line in the figure) outperforms the other methods especially when error rate introduced is larger.

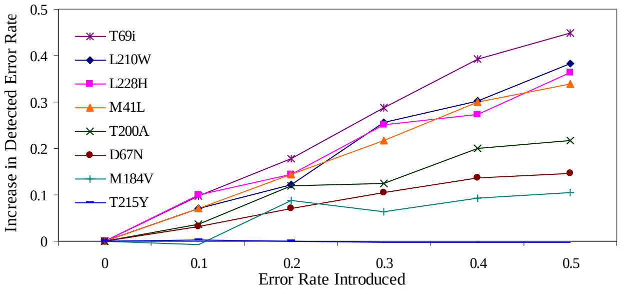


Figure 4: Increase in the BN-NC detected error rates over the initial error rates detected for TDF (the plot for mutation T215Y overlaps with the horizontal axis). The horizontal axis indicates the error rate that was introduced to every one of the mutations except T215Y.

4.1 HIV-1 Drug Resistance Prediction With no noise added, we learned BNs from the fifty training sets using the top-5 most informative features in each set. The Markov blankets of the learned BNs involved just the following seven mutations: T69i, L210W, L228H, M41L, T200A, D67N, and M184V. The BN-NC framework detected some initial noise in the seven mutations and Figure 2 shows the automatically detected initial error rates in decreasing order. The mutation values within this HIV-1 data set appear to be rather “clean”, with the average detected error rate being just 0.18. The cross-validation accuracy of BN-NC was 0.697, which was higher than the mean accuracy of 0.652 reported for drug TDF in [16]. Even though the mutations did not appear to contain a lot of noise, BN-NC was able to outperform Support Vector Machine (SVM), Naive Bayes (NBayes), J4.8 Decision Tree (DT), and BN without noise correction when no additional noise was injected.

Figure 3 presents the accuracies obtained from each method when we apply them on identical training and test splits with various degrees of noise introduced. The same error rate was added to each of the seven mutation features covered by the Markov blankets. For example, to introduce an error rate of 0.3, 30% of values for each of the seven mutations were randomly-picked from the entire data set and flipped. No error was introduced to the only other mutation, T215Y, that was learned in the BNs but not covered by any of the Markov blankets.

J4.8 DT and standard BN classifier performed very similarly during the evaluation, so much so that their lines in the chart of Figure 3 overlap. This is consistent with previous observations that DT and BN predict similarly without noise correction [26]. SVM only performed slightly worse than BN-NC with no noise added, but it’s performance quickly converged with DT and BN without noise correction once the errors were injected. NBayes performed worse than the other methods with no noise added because its assumption of complete feature independence did not match the fact that the real-world data contains inherent feature dependencies; its accuracy increased marginally with a small noise added possibly because the noise reduced the feature dependency and suited it. However, NBayes was substantially outperformed by BN-NC with more noise added to the mutations such that their erroneous values become less and less predictive of resistance. Overall, BN-NC maintained an accuracy close to 0.7 even when the error rate was raised to 0.4, while all the competing methods (SVM, NBayes, DT and the BN without noise correction) suffered a far greater drop in their accuracy.

To verify whether the proposed error discovery procedure can indeed sense the scale of the introduced noise correctly, we analyze, for various mutations, the *difference* between the detected error rate at each degree of introduced noise and the detected rate (shown in Figure 2) when no noise has been introduced. If the noise detection procedure works properly, we expect to see a steady rising trend in each mutation’s detected error rate as we add more noise. As shown in Figure 4, this is indeed the case. For each of the seven mutations in which noise was added, the detected error rate increases steadily as more noise is introduced to the mutations. The BN-NC procedure also correctly concludes that no additional noise was introduced to the mutation T215Y, which is represented by the only flat line in the figure.

The differences in the slopes for the different mutations are due to differences in the initial detected error rate, i.e., when no additional noise has been introduced and the mutation values contain only their inherent errors. This is because the additional errors are introduced into the data by randomly flipping the mutation

Table 1: Frequency of mutations in Markov blankets.

Mutation	Frequency	Is TAM	Is Q151M-complex
D67N	40	Yes	No
M41L	30	Yes	No
L210W	26	Yes	No
T69i	5	No	No
L210W & D67N	26	Yes	No

Table 2: Accuracy on the ALL/AML test set.

Method	Prediction Accuracy
Weighted Voting [2]	0.85
SVM [29]	0.88
EP [31]	0.91
ARAM [32]	0.94
SVM	0.88
NBayes	0.88
DT	0.91
BN	0.97
BN-NC	0.97

values in the samples according to the intended rate, which means that some of the inherent errors are unknowingly flipped in the process. A mutation with a smaller initial rate would have a steeper slope in Figure 4, because most of the randomly administered flips would have added on to the existing errors in the data set. Indeed, the lower the initial detected error rate for a mutation in Figure 2, the steeper the error increment that was detected for that mutation in Figure 4. The above results completely verify that the proposed noise detection procedure can indeed accurately detect not only the inherent error rates, but also the different degrees of noise that we introduced into the mutation features' values during the above controlled experiments.

Our next task is to verify if the learned BN captures known drug-resistance mutations in the class variable's Markov blanket. In drug resistance studies, a mutation selected by the virus as the first mutation to increase its resistance against a certain drug is referred to as a *major mutation*, and a mutation that is selected to increase resistance only in the presence of some other mutations is a *minor mutation*. We now verify whether well-known major and minor mutations are correctly recovered as part of the class variable's Markov blankets in the BNs; this would in turn validate the rationale underlying BN-NC's Markov blanket-based explanation procedure.

The NRTI family of antiretroviral drugs targets the reverse transcriptase (RT) DNA that is encoded by the HIV-1 virus. These inhibitors are designed to bind to the polymerase active site of the RT and block further growth of the rogue DNA. Biologically, HIV-1 excises NRTIs via two known mechanisms i) the thymidine-associated mutations (TAMs), comprising mutations D67N, M41L, L210W, T215Y/F, K70R and K219Q, and ii) the Q151M complex, comprising mutations Q151M, A62V, V75I, F77L, and F116Y [17]. We must establish that the learned Markov blankets cover these mutations.

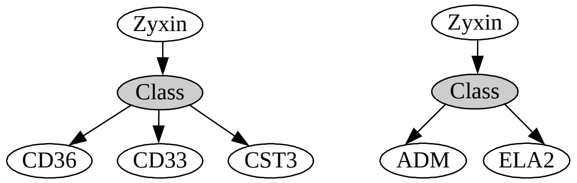
We analyzed the fifty independently learned BNs from our cross-validation experiments, and we tallied the number of times that each mutation was covered by the learned Markov blankets. Table 1 presents the findings. From this table, the major mutations are the D67N and M41L, which are both members of TAMs. Combinations of TAMs or T69i and TAMs are found in all Markov blankets, whilst no Q151M complex mutation could be found; this suggests that for TDF, TAMs are the predominant resistance pathways selected by HIV-1. Interestingly, L210W always appears in tandem with D67N, the most common mutation in the Markov blankets. Likewise, the T69i mutation (insertion of dipeptide between positions 69 and 70) always appears together with TAMs, which qualifies it as a minor mutation for HIV-1 against TDF. Indeed, biological experiments have shown that the combination of T69i and TAMs does provide strong resistance to all the NRTI drugs [30]. The findings thus successfully demonstrate that BN learning recovers features known to possess biological associations with the class variable, which enables BN-NC to generate biologically meaningful explanations based on nodes in the Markov blankets.

4.2 Acute Leukemia Subtypes Classification

4.2.1 The ALL/AML Data Set We use the same train-test split (with 38 training and 34 test examples) as previous works [2, 29, 31, 32] to facilitate comparison. Using the entropy-based feature selection and discretization method, 866 genes in the training set are partitioned into two to four intervals each, with no cutting points for the others, i.e., just 866 / 7,129 = 12.1% of the genes are discriminatory among the two subtypes.

Table 2 shows that BN-NC framework generates just a single misclassification on the 34 test samples (accuracy 0.97) using 24 genes, which outperforms all previous reported results on this data set by weighted voting [2], support vector machines (SVM) [29], emerging patterns (EP) [31], and adaptive resonance associative maps (ARAM) [32]. SVM, Naive Bayes classifier (NBayes) and J4.8 decision tree (DT) made between three to four misclassifications on the same test examples using the identical set of genes as BN and BN-NC.

The standard BN without our noise correction per-



(a) Using 24 genes (b) Using 18 genes

Figure 5: Markov blanket in learned BN for ALL/AML.

Table 3: Info. gains and error ests. for genes in Fig 5(a).

Gene-ID	Full Description	Info. Gain	Error Estimate
X95735	Zyxin	0.8680	0.0000
M27891	CST3 Cystatin C	0.7043	0.0000
M23197	CD33 antigen	0.5917	0.0526
M98399	CD36 antigen	0.5917	0.0789

formed very well on the test set. This is possible if the important genes contain little or no error, in which case any noise correction would not improve the result. Figure 5(a) shows the Markov blanket of the learned BN, consisting of genes *Zyxin*, *CD36 antigen*, *CD33 antigen* and *CST3 Cystatin C*. The corresponding information gains and error estimates discovered by our BN-NC framework are listed in Table 3. The genes in the class variable's Markov blanket in Figure 5(a) overlap with those that Tan and Pan [32] identified. Specifically, genes *Zyxin* and *CST3 Cystatin C* are consistently found to be highly discriminatory of leukemia tumors. With reference to Table 3, these two particular genes possess high information gain values of 0.868 and 0.704, respectively. Using our error discovery procedure, no error is detected for either of these genes, and very little error for the other two, which explains the standard BN's ability to discriminate tumor subtypes even with no noise correction. In this case, the insights from BN-NC in Figure 5(a) and Table 3 show us the underlying gene interactions and how their inherent noise characteristics affect the BN's accuracy. More importantly, the above results show that BN-NC does not overcompensate or indiscriminately assign error estimates to "clean" features but accurately reports the noise in each feature.

The genes within the Markov blanket of the learned BN in Figure 5(a) are known to have biological significance to acute leukemia, which once again validates that the BN-NC strategy of explaining BN classifications using the genes in Markov blanket is biologically sound. Specifically, Zyxin is a binding partner of transcription factor ZNF384, which is recurrently involved in ALL translocations [33]. Moreover, Zyxin is located in chromosome 7, which contains genes related to AML [34].

Table 4: BN-NC explanations for a ALL/AML test case.

(a) Without noise correction	(b) After noise correction
Class is AML (p = 0.517), as	*Class is ALL (p=0.639), as*
Zyxin $\leq$ *994 (p = 1.0),*	*Zyxin* $\leq$ *994 (p = 1.0),*
ADM > 185 (p = 1.0), and	*ADM > 185 (p = 0.8), and*
ELA2 > 197.5 (p = 1.0).	*ELA2 > 197.5 (p = 0.8).*

CST3 Cystatin C is an endogenous protein inhibitor of cathepsins, related to the etiology or the cause of ALL and AML [34]. In addition, CD33 antigen in chromosome 19q13.3 has been used for targeted antibody therapy to destroy AML cells [35], and high CD36 antigen expressions are associated with low AML survival [36].

Using 18 genes, BN learning displaces *CST3 Cystatin C* from the target's Markov blanket (Figure 5(b)), resulting in five misclassifications on the test set (accuracy 0.85). BN-NC discovers an error rate of 0.0789 for both ADM Adrenomedullin and ELA2 Elastatse 2 neutrophil, which are the two genes that are now captured in the Markov blanket apart from Zyxin, and BN-NC corrected two of the misclassifications automatically.

The genes ADM Adrenomedullin and ELA2 Elastatse 2 neutrophil are also biologically associated with leukemia. Adrenomedullin production is correlated with differentiation in human leukemia cell lines and peripheral blood monocytes [37], while Elastatse 2 neutrophil is a myeloid-restricted protein highly expressed in AML cells and is a potential protein vaccine against AML [38].

Table 4 shows the explanations generated by BN-NC for a test case, before and after the noise correction. The explanations provide us with insights to the BN inferences that would otherwise not be visible. First, the same BN classifier has actually reversed its diagnosis after our error discovery and likelihoods estimation procedure enables it to properly account for the uncertainty in the Markov values. Second, this change in tack is actually due to the BN's lower beliefs in the values of genes ADM Adrenomedullin and ELA2 Elastatse 2 neutrophil. Finally, BN-NC predicts ALL with a higher confidence (0.639) than it had predicted AML (0.517), which matches the fact that this is an ALL sample.

4.2.2 The Pediatric ALL Data Set There are 215 training and 112 test cases in the Pediatric ALL data set. Yeoh et al. [23] used support vector machines (SVM) as the classifier for this problem and correlation-based feature selection (CFS) for selecting discriminatory genes. They reported that the number of genes that gave the best test prediction accuracy varied from one to twenty among the six subtypes. For example, a single gene was sufficient to differentiate samples belonging to

Table 5: Test prediction accuracy on the Pediatric ALL test sets. The best results for each subtype are bolded.

Subtype	SVM [23]	SVM	NB	DT	BN	BN-NC
T-ALL	**1**	**1**	**1**	**1**	**1**	**1**
E2A-PBX1	**1**	**1**	**1**	**1**	0.99	**1**
TEL-AML1	0.99	0.98	**1**	0.96	0.99	**1**
BCR-ABL	0.97	**0.99**	0.98	0.98	0.98	**0.99**
MLL	0.98	**1**	**1**	0.97	**1**	**1**
Hype.>50	**0.96**	0.93	0.95	0.93	**0.96**	**0.96**

subtypes T-ALL and E2A-PBX1, but seven to as many as twenty genes were required to accurately predict the other four subtypes. For BN, by cross-validating using the training examples (results not shown), we find that using twelve to fifteen most-informative genes is optimal for all the six subtypes, and we use the identical sets of genes to compare the standard BN, BN-NC, J4.8 DT, SVM and the NBayes methods in all of our experiments.

As shown in Table 5, BN-NC outperforms the Support Vector Machine (SVM) in [23] on the same train-test split for the three subtypes TEL-AML1, BCR-ABL, and MLL, and it outperforms SVM in our experiments for subtypes TEL-AML1 and Hype.>50. BN-NC outperforms Naive Bayes (NBayes, denoted in the table as "NB") for subtypes BCR-ABL and Hype.>50, and is more accurate than J4.8 Decision Tree (DT) in four subtypes. Compared to standard BN (denoted in the table as "BN"), BN-NC performed superiorly in classifying subtypes E2A-PBX1, TEL-AML1 and BCR-ABL. The results show that, by effectively exploiting the feature dependencies in the training data, BN-NC was indeed able to predict better on the same test set compared to SVM, NBayes, DT and standard BN, all of which do not discover nor account for feature noise within the data. Overall, our BN-NC outperforms all the other methods for the six Pediatric ALL subtype classification tasks.

Next, we look to TEL-AML1 for an example of how feature interactions and noise corrections are directly interpretable under the BN-NC framework. With reference to Table 5, BN-NC achieved perfect test prediction for TEL-AML1. The BN learned from the training set is shown in Figure 6, where the Markov blanket for the target node *Class* is demarcated. This *Markov blanket* for the *Class* variable shields it from the other nodes and completely predicts its behavior. Our proposed BN-NC framework exploits this fact to auto-generate explanations for the BN predictions and any noise correction.

Tables 6(a) and 6(b) show the BN-NC explanations for the BN prediction on a test example of TEL-AML1, before and after noise correction. Similar to what we have seen from the ALL/AML experiments earlier, noise correction enables the BN to consider the uncertainties in the gene measurements, to correctly classify this particular test example as not_TEL (with a much higher confidence than before). The ability to reliably estimate the error rate of each gene and translate it into likelihoods estimations for proper consideration by the BN classifier is very important. In this case, as shown by the explanation in Table 6(b), more than half of the Markov nodes in Figure 6 (for example, gene $a36239$ and gene $a1077$) exhibit certain degree of uncertainty that would have been overlooked by the BN (Table 6(a)) if not for our noise correction. BN-NC auto-generates the explanation in Table 6(c) as insight to the corrections in values of genes $a37780$ and $a1299$ during the inference.

Table 6: BN-NC explanations for TEL-AML1 test case.

(a) Without noise correction	(b) After noise correction
Class is TEL (p = 0.773), as	*Class is not_TEL (p = 0.919), as*
a38652 = 0 (p = 1.0),	*a38652 = 0 (p = 1.0),*
a36239 = 2 (p = 1.0),	*a36239 = 2 (p = 0.6),*
a1077 = 1 (p = 1.0),	*a1077 = 1 (p = 0.8),*
a38203 = 0 (p = 1.0),	*a38203 = 0 (p = 1.0),*
a35614 = 0 (p = 1.0),	*a35614 = 0 (p = 1.0),*
a32224 = 0 (p = 1.0),	*a32224 = 0 (p = 1.0),*
a37780 = 1 (p = 1.0),	*a37780 = 0 (p = 0.7),*
a38578 = 1 (p = 1.0),	*a38578 = 1 (p = 0.7),*
a41442 = 0 (p = 1.0),	*a41442 = 0 (p = 1.0),*
a36985 = 1 (p = 1.0), and	*a36985 = 1 (p = 0.6), and*
a1299 = 2 (p = 1.0).	*a1299 = 0 (p = 0.8).*

(c) Gene value correction by BN-NC

a37780 is corrected from (a37780 = 1) to (a37780 = 0), as
Given a38203 = 0 and a35665 = 0,
(a37780 = 0) has p = 0.7,
while (a37780 = 1) only has p = 0.3.
a1299 is corrected from (a1299 = 2) to (a1299 = 0), as
Given a35614 = 0,
(a1299 = 0) has p = 0.8,
while (a1299 = 2) only has p = 0.1.

To summarize the experimental results, our BN-NC framework performs more robustly on noisy data compared to BN without noise correction and other competing methods including Decision Tree (DT), Naive Bayes (NBayes) and Support Vector Machine (SVM). Moreover, in contrast to methods that do not yield predictive models that are directly interpretable in the form of graphs or rules (e.g. SVM [29] and lazy classification methods like k-NN [39]), BN-NC automatically generates probabilistic rules to explain each BN inference and any noise correction. Together, the resulting graphical BN model and the auto-generated explanations help to make the BN classifications more comprehensible and offer us new insights to the feature interaction processes.

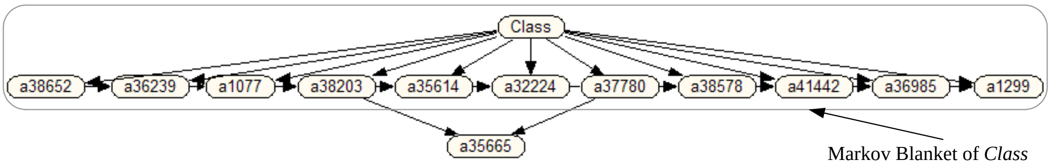


Figure 6: BN learned from training set for TEL-AML1.

5 Conclusion

By effectively capturing and exploiting feature dependencies from noisy biomedical data, we have introduced the BN-NC framework, a coherent knowledge discovery process for noisy biomedical data classification, and demonstrated its efficacy using a number of real-world data sets. The capability to automatically identify noisy features and thereafter correct their values in real-time during the prediction, as well as to explain both the classification outcomes and the noise corrections underlying those decisions, is unique to our proposed framework.

As a key contribution in this paper, we have sought and found answers to important concerns regarding the overall efficacy of the proposed BN-NC framework when applied to biomedical prediction. We have validated empirically using a HIV-1 drug resistance prediction data set that i) the noise discovery procedure correctly identifies noisy features, ii) the BN-NC framework consistently outperforms state-of-the-art classifiers including decision tree, naive Bayes and support vector machine in overcoming noisy feature values, and iii) the features within the target variable's Markov blanket have known target associations that facilitate auto-generation of biologically meaningful explanations by BN-NC. In addition, we have validated using two real-world leukemia subtype classification data sets that BN-NC indeed performs at the frontier of existing methods on noisy DNA microarray data, which represents a prevalent source of biomedical data used for medical classification today.

The proposed BN-NC framework is readily applicable to noisy biomedical classification tasks, and it also extends to other domains suffering from noisy features. Moving ahead, we shall continue to research and develop effective solutions to overcome problems posed by data noise. Issues to be investigated include the performance of alternative feature selection techniques such as the Fisher criterion adopted in [32], and the learning performance of existing metric or constraint-based algorithms for Bayesian network learning from data [13]. We hope that our good results and the biologically-sound causal models presented in this work would alleviate some of the concerns of researchers regarding the ability to learn correct BN structures, and this would encourage more of us to apply BN learning from data to other problems.

References

[1] G.-E. Yap, A.-H. Tan, and H.-H. Pang, "Learning causal models for noisy biological data mining: An application to ovarian cancer detection," in *Proceedings of AAAI-07*, Vancouver, BC, July 2007, pp. 354–359.

[2] T. R. Golub, D. K. Slonim, and et al., "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–537, October 1999.

[3] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," in *Proceedings of RECOMB*, 2000, pp. 127–135.

[4] J. Han, "How can data mining help bio-data analysis?" in *Proceedings of BIOKDD (with KDD Conf.)*, 2002, pp. 1–2.

[5] R. Yamaguchi and T. Higuchi, "State-space approach with the maximum likelihood principle to identify the system generating time-course gene expression data of yeast," *IJDMB*, vol. 1, no. 1, pp. 77–87, 2006.

[6] J. T. L. Mah, D. C. C. Poo, and S. Cai, "UASMAs (Universal Automated SNP Mapping Algorithms)," in *Proceedings of 36th International Conf. on Very Large Data Bases*, 2010, pp. 1406–1413.

[7] P. Bajcsy, J. Han, L. Liu, and J. Yang, *Data Mining in Bioinformatics.* Springer Berlin, 2005, pp. 9–39.

[8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* M. Kaufmann, 1988.

[9] S. Y. Rhee, J. G. Matthew, R. Kantor, B. J. Betts, J. Ravela, and R. W. Shafer, "Human immunodeficiency virus reverse transcriptase and protease sequence database," *Nucleic Acids Research (NAR)*, vol. 31, no. 1, pp. 298–303, 2003.

[10] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of IJCAI*, 1993, pp. 1022–1029.

[11] J. Quinlan, *C4.5: Programs for machine learning.* San Francisco, CA, USA: Morgan Kaufmann, 1993.

[12] S. Nikolajewa, R. Pudimat, M. Hiller, M. Platzer, and R. Backofen, "BioBayesNet: A web server for feature extraction and Bayesian network modeling of biological sequence data," *NAR*, vol. 35, no. Web Server issue, pp. W688–W693, 2007.

[13] K. B. Korb and A. E. Nicholson, *Bayesian Artificial Intelligence.* CRC Press, 2004.

[14] G.-E. Yap, A.-H. Tan, and H.-H. Pang, "Explaining inferences in Bayesian networks," *Journal of Applied Intelligence*, vol. 29, no. 3, pp. 263–278, 2008.

[15] J. R. Koiter, "Visualizing inference in Bayesian networks," Ph.D. dissertation, Delft Univ. of Tech., 2006.

[16] S.-Y. Rhee, J. Taylor, G. Wadhera, A. Ben-Hur, D. L. Brutlag, and R. W. Shafer, "Genotypic predictors of human immunodeficiency virus type 1 drug resistance," *PNAS*, vol. 103, no. 46, pp. 17 355–17 360, 2006.

[17] H. Saigo, T. Uno, and K. Tsuda, "Mining complex genotypic features for predicting HIV-1 drug resistance," *Bioinformatics*, vol. 23, no. 18, pp. 2455–2462, August 2007.

[18] K. Deforche, R. Camacho, K. V. Laethem, P. Lemey, A. Rambaut, Y. Moreau, and A.-M. Vandamme, "Estimation of in vivo fitness landscape experienced by HIV-1 under drug selective pressure useful for prediction of drug resistance evolution during treatment," *Bioinformatics*, vol. 24, no. 1, pp. 34–41, 2008.

[19] K. Deforche, R. Camacho, and et al., "Bayesian network analysis of resistance pathways against HIV-1 protease inhibitors," *Infection, Genetics & Evolution*, vol. 7, no. 3, pp. 382–390, 2007.

[20] American Cancer Society, *Cancer Facts & Figures 2009*. Atlanta: American Cancer Society, 2009.

[21] C. H. Pui and W. E. Evans, "Acute lymphoblastic leukemia," *NEJM*, vol. 339, pp. 605–615, 1998.

[22] D. Delen, G. Walker, and A. Kadam, "Predicting breast cancer survivability: A comparison of three data mining methods," *Artificial Intelligence in Medicine*, vol. 34, no. 2, pp. 113–127, 2005.

[23] E.-J. Yeoh, M. E. Ross, and et al., "Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling," *Cancer Cell*, vol. 1, pp. 133–143, March 2002.

[24] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of UAI*. San Mateo: M. Kaufmann, 1995, pp. 338–345.

[25] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[26] G.-E. Yap, A.-H. Tan, and H.-H. Pang, "Discovering and exploiting causal dependencies for robust mobile context-aware recommenders," *IEEE TKDE*, vol. 19, no. 7, pp. 977–992, July 2007.

[27] C. Elkan, "Magical thinking in data mining: Lessons from CoIL challenge 2000," in *Proceedings of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. San Francisco, California: ACM, 2001, pp. 426–431.

[28] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, Jr., and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 1, pp. 262–267, 2000.

[29] T. S. Furey, N. Cristianiini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, pp. 906–914, 2000.

[30] V. Vivet-Boudou, J. Didierjean, C. Isel, and R. Marquet, "Nucleoside and nucleotide inhibitors of HIV-1 replication," *Cellular and Molecular Life Sciences*, vol. 63, pp. 163–186, 2006.

[31] J. Li and L. Wong, "Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns," *Bioinformatics*, vol. 18, no. 5, pp. 725–734, 2002.

[32] A.-H. Tan and H. Pan, "Predictive neural networks for gene expression data analysis," *Neural Networks*, vol. 18, pp. 297–306, 2005.

[33] H. Janssen and P. Marynen, "Interaction partners for human ZNF384/CIZ/NMP4 - Zyxin as a mediator for p130CAS signaling?" *Experimental Cell Research*, vol. 312, no. 7, pp. 1194–1204, 2006.

[34] C. Yoo, I.-B. Lee, and P. A. Vanrolleghem, "Interpreting patterns and analysis of acute leukemia gene expression data by multivariate fuzzy statistical analysis," *Computers and Chemical Engineering*, vol. 29, no. 6, pp. 1345–1356, 2005.

[35] J. G. Thomas, J. M. Olson, S. J. Tapscott, and L. P. Zhao, "An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles," *Genome Research*, vol. 11, pp. 1227–1236, 2001.

[36] G. Perea, A. Domingo, and et al., "Adverse prognostic impact of CD36 and CD2 expression in adult de novo acute myeloid leukemia patients," *Leukemia Research*, vol. 29, no. 10, pp. 1109–1116, 2005.

[37] A. Kubo, N. Minamino, Y. Isumi, K. Kangawa, K. Dohi, and H. Matsuo, "Adrenomedullin production is correlated with differentiation in human leukemia cell lines and peripheral blood monocytes," *FEBS Letters*, vol. 426, no. 2, pp. 233–237, 1998.

[38] Q. Le, J. Melenhorst, N. Hensel, R. Eniafe, and A. Barrett, "348: Human neutrophil elastase stimulating CD4+ and CD8+ T cells is a potential protein vaccine for leukemia patients with diverse HLA types," *BBMT*, vol. 14, no. 2, pp. 127–128, 2008.

[39] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE TIT*, vol. 13, pp. 21–27, 1967.