# Using Iterative Ridge Regression to Explore Associations Between Conditioned Variables

**Nimrod Bar-Yaakov[1], Zehava Grossman[2,3], Nathan Intrator[1]**

[1] School of Computer Science, Tel-Aviv University, Tel Aviv 6998, Israel

[2] Central Virology Laboratory, Public Health Laboratories, Israel Ministry Of Health, Sheba Medical Center, Ramat-Gan 52621, Israel

[3]School of Public Health, Sackler Faculty of Medicine, Tel-Aviv University, Tel Aviv 6998, Israel

**Contact information:**

*Nimrod Bar-Yaakov, tel.: +972-54-6863225, fax: +972-3-6409357, e-mail: nimrodby@post.tau.ac.il

Prof. Zehava Grossman, tel.: +972-3-5302458, fax: +972-3-5302457, e-mail: zehava.grossman@gmail.com

Prof. Nathan Intrator, tel.: +972-3-6407598, fax: +972-3-6409357, e-mail: nin@post.tau.ac.il

## 1) Abstract

We address a specific case of joint probability mapping, where the information presented is the probabilistic associations of random variables under a certain condition variable (conditioned associations).

Bayesian and Dependency networks graphically map the joint probabilities of random variables, though both networks may identify associations that are independent of the condition (background associations). Since the background associations have the same topological features as conditioned associations, it is difficult to discriminate between conditioned and non-conditioned associations, which results in a major increase in the search space.

We introduce a modification of the dependency network method, which produces a directed graph, containing only condition-related associations. The graph nodes represent the random variables and the graph edges represent the associations that arise under the condition variable. This method is based on ridge-regression, where one can utilize a numerically robust and computationally efficient algorithm implementation.

We illustrate the method's efficiency in the context of a medically relevant process, the emergence of drug resistant variants of HIV in drug-treated, HIV-infected people. Our mapping was used to discover associations between variants that are conditioned by the initiation of a particular drug treatment regimen. We have demonstrated that our method can recover known associations of such treatment with selected resistance mutations as well as documented associations between different mutations. Moreover, our method revealed novel associations that are statistically significant and biologically plausible.

**Key words:**

Joint Probability Mapping, Ridge Regression, Dependency Networks, Bayesian Networks, HIV, Resistance mutations.

## 2) Introduction

The search methods of associations, dependencies, and correlations between random variables have been thoroughly studied during the last five decades, and in some cases, there is a need to recover a specific type of associations: those that result from the existence of a joint event. For example, dependencies between variables can change with the appearance of another variable; different pattern of associations are created with the appearance of this variable. This variable will herein be referred to as **Condition**, and the associations generated by this condition – **Conditioned associations**.

Mathematically speaking, we want to find a subset $S$ out of a set of random variables $\{x_1, x_2, \dots x_n\}$ where $p(S/c)$ significantly differs from $p(S)$.

Exploring the conditioned associations can recover hidden mechanisms or processes that operate only when the condition exists. Such mechanisms can be quantitative and qualitative modifications of elements of the immune system in response to pathogens, emergency services recruitment upon a catastrophic event, or DNA mutations in an infecting virus arising under the selection pressure of a novel treatment.

Pearl (2009) specifies the benefits of using a graph representation of joint probability mapping, including clarity, convenience and economical representation. Graphical notation and terminology will be used further in this paper.

### a) Problem Example

Assume there are three random binary variables: Car rental shortage ($\boldsymbol{x_1}$), Airport closure ($\boldsymbol{x_2}$), Western wind ($\boldsymbol{x_3}$) , and a Condition – Icelandic volcano eruption ($\boldsymbol{c}$) .

Exploring the correlations between $\boldsymbol{x_1}$, $\boldsymbol{x_2}$ and $\boldsymbol{x_3}$ will show that $\boldsymbol{x_1}$correlates with $\boldsymbol{x_2}$while none of them correlates with $\boldsymbol{x_3}$. Thus, normally there is no apparent correlation between western wind and airport closure or car rental shortage, or $p(\boldsymbol{x_1, x_2/x_3}) = \boldsymbol{p(x_1, x_2)}$. Yet, when an Icelandic volcano erupts, a western wind can carry the ash cloud over the airport, causing it to shut down, in which case

$p(\boldsymbol{x_2}, \boldsymbol{x_3}/\boldsymbol{c}) \gg \boldsymbol{p}(\boldsymbol{x_2}, \boldsymbol{x_3})$. In a graph representation, retrieving the $\boldsymbol{x_3} \rightarrow \boldsymbol{x_2}$ edge will suggest that a new association was created under the Condition $\boldsymbol{c}$ (see figure 1).

**[suggested place for figure 1]**

**b) Using Bayesian Networks to infer about conditioned patterns**

Bayesian Networks (BNT) represent the conditional dependencies between random variables, using a Directed A-cyclic Graph (DAG) (Cooper and Herskovits 1992), and are commonly used in similar problems to ours, mapping the associations between genes (Friedman, et al. 2000) and the associations between HIV DNA mutations (Deforche, et al. 2006).

The condition variable is usually referred to as a variable with a predefined value. Predefining the value of a variable is commonly termed as intervention or perturbation of the observed data.

Pearl (2009) uses the term Atomic Intervention to describe an external manipulation of a network variable value. He suggests (J. Pearl 1993) adding to the condition variable a parent variable that activates the induced state.

Pe'er et al (2001) explored the causal relationships between genes' expression level. In their paper, they note that external perturbations such as medical treatment, that have no direct influence over other variables (in this case gene expression level) but indirectly effect the values of many other variables, should be regarded as "indicator variables", and added to the Network with the constraint that they cannot have other variables as network parents.

Note that the above examples require manipulating or placing constraints over the discovered network structure.

The result of inducing the condition variable to our data will be a joint probability map, containing a node representing the condition. In order to deduce the conditioned associations, one can traverse through the nodes connected to the condition node, or review the whole connected component containing the condition node.

**[suggested place for figure 2]**

Figure 2 displays the joint probability mapping of our example, represented by a Bayesian Network. It is clear that the new edge, $\boldsymbol{x_3 \rightarrow x_2}$ which appears in the resulting network, suggests that this unique association has been discovered.

However, this case demonstrates the ambiguity encountered when trying to discover conditioned associations using BNT. The $\boldsymbol{x_3 \rightarrow x_2}$ association can result from existing correlation between $(\boldsymbol{x_3})$ and $(\boldsymbol{x_2})$ in the general population and is not specific to the conditioned case, in which **c=true**. When this graph is the only source of information, there is no way to distinguish between ***Background associations*** (in the general case) and ***Conditioned associations*** (in the conditioned case).

This matter could be resolved by exploring the differences between the BNT without the condition variable and the BNT containing the condition variable. Yet in many cases, the proportion of the samples with positive value of the condition variable may be large enough for the conditioned associations to appear in the BNT without the condition variable. In other cases, when the proportion of the samples with positive condition value is small enough, the conditioned associations will not be discovered in the BNT with the condition variable. In fact, the former ambiguity was clearly visible in the simulation described later in this paper (see paragraph 5e).

Deforche et al. (2006) overcomes this issue by preexamining the candidate variables for the BNT, and selecting variables that show significant correlation with the condition. This way the chance that background associations will arise is lowered; conversely, conditioned associations with variables that are not significantly correlated with the condition may be lost. In the above example, since western wind $(\boldsymbol{x_3})$ is not correlated with the volcanic eruption $(\boldsymbol{c})$, $(\boldsymbol{x_3})$ will not be used in our BNT, therefore the $\boldsymbol{x_3 \rightarrow x_2}$ association will be lost.

Another drawback might be caused by the stochastic nature of searching the BNT model and its Directed A-Cyclic Graph (DAG) structure. Reconstructing BNT structure from given observational data requires searching through all possible models, looking for the model that maximizes the log-likelihood of the data. Since this problem is NP-Hard (Chickering, Heckerman and Meek 2004), cases with more than a few variables require heuristic search methods (Friedman, et al. 2000). Since prior causal knowledge is missing in many cases, our model search can result in structures that satisfy the observed statistical dependencies, but cannot accurately recover the underlying "real" structure or suggest an equivalent model that may hide the conditioned associations.

**[suggested place for figure 3]**

Consider the case displayed at figure 3a, where an additional variable is added to our network – Ash cloud ($\boldsymbol{x_4}$): this DAG can have *observationally equivalent* (J. Pearl 2009) DAGs (3b) where the statistical dependencies between the variables (herein, *d-separation* (J. Pearl 2009)) are still maintained, i.e., ($\boldsymbol{c}$) and ($\boldsymbol{x_2}$) are independent given ($\boldsymbol{x_4}$). In the case displayed in (3b), the causal relationship is obviously wrong.

This example shows the ambiguity faced when trying to draw conclusions concerning the conditioned association, while exploring the graph structure of the BNT. In cases where the network structure is unknown, a single criterion cannot be applied in order to locate the conditioned associations, since we can alternate between two different graph structures that represent a single conditioned association.

This ambiguity effects the search procedure applied in order to locate the novel conditioned associations. To ensure that the conditioned association edges are covered by our search space, the edge directions should be ignored and the edges of the connected component containing the condition should be added to our search space. However, our search space will be greatly obscured by background association added to our result. Applying the methods described above, such as adding an activating variable (J. Pearl 1993) to the condition variables, or removing the edges from the parents of the

condition variable, while increasing the inference capabilities of the network, still does not eliminate the ambiguity regarding other variables.

### c) Dependency Networks

Heckerman et al. (2000) suggest a method that deals with some of the problems that arise, when interpreting a resultant BNT, especially the tight topology constraints, such as Directed A-cyclic Graph, and the dependencies implied from the d-separation criteria.

This method iteratively regresses, using probabilistic decision trees. Each variable uses the rest of the model variables, and uses the prediction capabilities of the regressor variables as criteria for adding an edge between the regressor variables and the predicted variable. This method retains the "Markov Blanket" property of the BNT – given the parents' values for a network variable; it renders this variable independent of all other variables.

Dependency Networks provide another advantage over other modeling methods – since the computation is done locally over the variable's immediate neighborhood, the computational effort is polynomial over the number of variables, and as such, more efficient.

Meinshausen et al. (2006) suggest a method for reconstructing the variables' graphical neighborhood using an estimation of Tibshirani's (1996) Lasso method, applying penalty over the regression coefficients norm, and applying a zero value to coefficients with negligible prediction values. While regressing a variable using all the other variables as predictors. Non-zero coefficients will add an edge between the predictor and the predicted variables, that is, enabling a robust and computational efficient method for reconstructing a dependency network, especially in high dimensional cases.

Friedman et al. (2008) later enhance this model, suggesting an efficient exact solution to the lasso peer regression problem, using the Banerjee et al. (2008) blockwise coordinate descent approach. This method, herein graphical lasso, is demonstrated on cell signaling data from 11 different proteins.

Dependency networks efficiently solve the above problems, providing a straightforward interpretation and usage of a robust numerical mechanism. Yet the ambiguity between the background and conditioned associations still needs to be solved, since the condition variable is still part of the resulting network.

**d) Summary of current state of the art and the proposed method.**

The above sections display the inherent limitations of locating conditioned associations using BNT modeling, mainly the DAG structure that constrains the network structure and observational equivalency that may produce different graphical models for similar variable dependencies. Both limitations can cause the conditioned associations to appear far, or not on the path of the condition variable in the result graph, and so, largely increase the number of associations that need to be tested as candidates for conditioned associations.

Dependency Networks and their recent adaptations solve this problem by addressing the local dependency neighborhood of the variable. Therefore, at least one of the conditioned association vertices should be a neighbor of the condition variable.

Yet, both Bayesian and Dependency networks may identify associations that are independent of the condition (background associations). Since the background associations have the same topological features as conditioned associations, it is hard to discriminate between conditioned and non-conditioned associations, which results in a major increase in the search space.

We introduce a modification of the dependency network method, which obtains a graph containing only conditioned related associations.

We suggest a tool, based on an efficient and numerically robust implementation of ridge regression, which maps the associations of variables under the condition. These properties render it invariant to the proportion of the conditioned samples in the population. The resulting graph identifies the associations between variables that are correlated specifically under the condition variables, and so solving the

ambiguity between associations in the general population (background associations) and conditioned associations.

## 3) Method – Iterative ridge regression (IRR)

We suggest a method that graphically maps conditioned associations between variables. Each association can be illustrated as a directed edge between the variables. The edges can be weighted according to the assumed prediction strength between the variables under the condition. The result will be a network structure (built as a directed Graph) that will map the variable associations under the condition.

Exploring the network structure is done by regressing the condition variable using the remaining variables, observing the weight change of the regressor variables when omitting one, and then again regressing the condition variable by the remaining variables.

We will show that the weight change in the remaining variables is relative to their predictive value over the omitted variable under the condition.

Prediction capabilities of one variable by another, under the condition, can be used as a cue for conditioned association, which can later be tested separately by a simple hypothesis testing tool.

### a) Defining the problem

Let $S = \{x_1, x_2, \dots x_n\}$ be a set of random binary variables (features), random binary variable $c$ (Condition), and a data sample set D, where each sample represents a vector for the value of $x_1, x_2, \dots x_n$ and their corresponding $c$ value. Next, generate a graph (network) $G = \{V, E\}$ where each node corresponds to a variable of $\{x_1, x_2, \dots x_n\}$; add an edge $\boldsymbol{x_j} \rightarrow \boldsymbol{x_k}$, if the association $\{\boldsymbol{x_j}, \boldsymbol{x_k}\}$ under the condition $c$ is significantly better than in the general population. The direction of the edge is dependent upon the prediction direction between $\boldsymbol{x_j}$ and $\boldsymbol{x_k}$ if $\boldsymbol{x_j}$ can predict the value of $\boldsymbol{x_k}$ under the condition $c$, there will be an edge directed from $\boldsymbol{x_j}$ to $\boldsymbol{x_k}$, and vice versa.

**b) Selection or regularized regression method**

In a classic regression problem, we want to find a weight set W that satisfies the equation:

$$\underset{W}{\operatorname{argmin}} E\left(Y - \sum_{i=1\to n} w_i x_i\right)^2 \tag{1}$$

In order to handle the problem of regressing variables with high covariance (typical to the problems explored by the IRR), which cause instability when inverting the sample matrix, a statistical method called Ridge Regression or Tikhonov regularization (Tychonoff and Arsenin 1977) is utilized. This method adds a penalty over the weight vector norm.

$$\underset{W}{\operatorname{argmin}} E\left(Y - \sum_{i=1\to n} w_i x_i\right)^2 + \lambda \parallel W \parallel^2 \tag{2}$$

Where the $\lambda$ parameter stands for the ridge parameter.

Another method, called The Lasso regularization, suggested by Tibshirani (1996), replaces the ridge regression's L2 norm penalty of the weight vector, with L1 norm penalty –

$$\lambda \parallel W \parallel \ell 1$$

While it also has an efficient solution of the weight vector, it also tends toward a zero value for negligible coefficients when using a large enough $\lambda$ value.

As was recently suggested, the Elastic nets (Zou and Hastie 2005) use a mix of L2 and L1 regularization over the weight norm –

$$\lambda((1-\alpha) \parallel W \parallel^2 + \alpha \parallel W \parallel_{\ell 1})$$

where the α parameter value determines whether the penalty leans toward L2 (which displays the ridge regression behavior) or L1 (Lasso).

In their most recent paper, Friedman et al. (2010), display a distinction between the above three regularization methods – Ridge regression, Lasso, and elastic nets – in regards to the effect of the weight change of regressor variables after omitting one of them.

Since the L2 norm penalty amplifies an uneven weight distribution, Ridge regression tends to split the coefficient weight of the omitted variable between the highly correlated variables. Lasso, on the other hand, tends to pick one correlated variable to receive the weight of the omitted variable. Elastic-nets behavior depends on the value of the α parameter.

In our case, the inter-relations between the variables are more relevant than the precision of the model's prediction. Therefore the ridge regression will be more suitable for our needs, while there is a need for a threshold value to filter the negligible interaction,

**c) Iterative interaction discovery**

Once the regularization method is chosen, the conditioned interactions are discovered while iterating over the model variables.

At the first stage, the regression of the condition variable will be done, using the variables in S:

$$W^B = \underset{W}{\operatorname{argmin}} E\left(C - \sum_{x_i \in S} w_i x_i\right)^2 + \lambda \parallel W \parallel^2 \quad (3)$$

Later, in each iteration j, the values of variable $\boldsymbol{x_j}$ will be omitted from the samples' data and the condition variable will again be predicted using the remaining variables.

$$W^j = \underset{W}{\operatorname{argmin}} E\left(C - \sum_{x_i \in S/x_j} w_i x_i\right)^2 + \lambda \parallel W \parallel^2 \quad (4)$$

This results in a new set of weights for the remaining features – $W^j$ , that will be compared with the original set of weights. The weight difference of each feature can be computed: $\Delta W^j = W^B - W^j$

As proved in appendix 7c, $\Delta W^j$ is the coefficients vector that gives the best regularized estimation of $W_j^B x_j$.

$$\Delta W^j = \underset{W}{\operatorname{argmin}} E\left( W_j^B x_j - \sum_{x_i \in S/x_j} w_i x_i \right)^2 + \lambda \parallel W^B - W \parallel^2 \quad (5)$$

At this point, the benefits of ridge regression come in hand. The best estimation of $W_j^B x_j$ is biased towards the selection of $\Delta W_j$, which evenly distribute the weights between variables that are correlated with $\boldsymbol{x_j}$ under C.

For instance, if $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are fully correlated with $\boldsymbol{x_j}$, the result will be $\left[\Delta W^j\right]_1 = \left[\Delta W^j\right]_2 = \frac{1}{2}$, while in L1 norm penalty, the result may be $\left[\Delta W_j\right]_1 = 1$ and $\left[\Delta W_j\right]_1 = 0$, or vice versa.

In this case, one can expect all of the variables that highly correlate with $\boldsymbol{x_j}$ under $c$, to receive significant value in the corresponding $\Delta W^j$ index.

Therefore, each element i in $\Delta W^j$ represents the prediction level of $\boldsymbol{x_j}$ using $\boldsymbol{x_i}$ under the condition $c$. This element can be used as a cue for the conditioned association level between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$.

If $\left[\Delta W^j\right]i$ passes a known threshold $t$, the association $\boldsymbol{x_j} \rightarrow \boldsymbol{x_k}$ can be tested using an independent $\boldsymbol{X^2}$ test, whether it is statistically significant under the condition $c$. The $X^2$ test results are adjusted for false discovery rate using the Hochberg-Benjamini correction (Benjamini and Hochberg 1995).

Passing both criteria, an edge $x_j \rightarrow x_k$ will be added to our graph G.

(See appendix 7a and 7b for Pseudo code and demonstration of the method).

**d) Complexity of the IRR algorithm**

Since the IRR algorithm uses both Ridge regression and $\boldsymbol{X^2}$ test, both have many robust and efficient implementations, its calculation time is a linear function over the sample size.

Overall, using an N size variable set and an M size sample set, the IRR regresses the sample set N times, then tests again within the set all of the N variables for significant associations.

In this case the IRR's asymptotic complexity is $O(N^2M)$.

## 4) Comparing the IRR algorithm with Bayesian Networks

Herein is a performance comparison of the IRR algorithm, with state of the art Bayesian network algorithm. Both algorithms try to locate a-priory induced associations between variables that correlate under a condition.

### a) Data set construction

The sample set contains 1000 samples with 20 variables. Each variable can hold a binary value of -1 or 1.

Each sample is assigned with a binary condition value; the total size of the condition vector is 1000 samples. All sample values were initialized to -1, afterwards 0.2 of the samples had their values inversed to simulate background noise.

Out of the 20 variables, 4 variables were randomly selected to hold the conditioned pattern – their values will be correlated under the condition and random otherwise.

Out of the condition vector, a predefined ratio (0.8) of the samples were set as a condition and their values were set to 1. Out of the conditioned samples, a predefined ratio (0.2) was randomly selected to hold the pattern. These samples will have a positive (1) value in the conditioned pattern variables.

After the construction of the sample set, sampling noise was induced by inversing the values of a predefined ratio (Noise Level) of the sample set.

### b) Simulation overview

Both the IRR and the BNT algorithms were used to extract the conditioned pattern variables. We used Kevin P. Murphy's BNT Matlab package (Murphy 2001) for BNT construction. The BNT was constructed using the K2 algorithm (Cheng, Bell and Liu 1997) that outperformed other network construction algorithms (Leray and Francois 2004).

As described in the chapters above, The BNT result DAG, was searched for the Connected Component containing the condition variable. All the variables in this Connected Component (ignoring edge directions) were tested as the result pattern.

The BNT and the IRR result patterns were tested against the induced pattern, and scored using the Jaccard Index. For each noise level, 100 independent simulation executions were performed, and the mean and standard deviation of the Jaccard score were used as a basis for performance comparison between the IRR and the BNT (see exact simulation details in appendix 7d).

**c) Results**

**[suggested place for table 1]**

**[suggested place for figure 4]**

**d) Discussion of the comparison results**

The simulation results show that the IRR performs well when trying to locate conditioned patterns in data that have up to 25% noisy (i.e. reversed) values–. At these noise levels, the IRR significantly ($p<0.01$) outperforms known state of the art algorithms such as the K2 BNT algorithm.

These results emphasize the effect of association ambiguity of the BNT, as shown in the following simulation case:

In this case, a 4 sized pattern that includes variables number 1, 16, 17 and 20, was induced to a 20 random variables data set. These variables associate only under the condition; otherwise, they are randomly correlated.

Figure 5 shows that the IRR result is a fully connected graph, which includes the pattern variables. The BNT graph, which by nature is a DAG (directed a-cyclic graph), contains the pattern variables but also two other variables (var12 and var15) that were randomly associated with the pattern variables, regardless of the condition.

As described before, we can only assume about the conditional nature of the BNT result by exploring the connected component of the graph containing the condition, and ignoring the direction of the edges, thus include variables 12 and 15 in the suggested pattern result.

In this case the Jaccard index score of the IRR graph will be 1 (fully compatible with the pattern), while the BNT result will be 0.67.

**[suggested place for figure 5]**

## 5) Application of the IRR for prediction of HIV resistance mutation patterns

We used IRR to explore patterns of HIV RNA mutations that emerge in HIV infected people after the initiation of antiretroviral drug treatment. These mutations, termed drug-resistant mutations, are preferentially selected because they render the virus resistant to the drugs admitted. In fact, this was the trigger for the development of IRR in the first place.

In most cases, the treatment is targeted to interfere with the activity of specific HIV proteins, such as viral protease, reverse-transcriptase and integrase. The mutations occur in parts of the viral-RNA sequences that map into the protein active sites upon translation. These are the sites that are affected by the HIV treatment. Characterization of the associations between resistance mutations and treatment regimens can be beneficial in the selection and optimization of treatment. It can be used to predict the appearance of resistance mutations or to assess the functional behavior of the mutant protein and reveal interactions among the drugs and among different coexisting mutations.

Deforche et al. (2006), have explored the interactions between resistance mutations using Bayesian Networks modeling. Their data set contained amino-acid samples of the protease site from HIV patients and their treatment history (focusing specifically on the protease inhibitor (PI) Nelfinavir, or NFV). First, $X^2$ test was applied to identify mutations that significantly correlated with the treatment. A new data set was then created, where the presence/absence of a given mutation in each sample was represented by the Boolean value of a variable assigned to this mutation. An additional variable was assigned to the NFV treatment status of this sample.

BNT modeling was used to explore associations between the variables described above. The resultant network recovered some of the known interactions between the NFV-induced mutations. It also pointed to new associations that were later found to be biologically meaningful.

We will briefly describe the application of IRR to the same biological process using our own data.

**a) Data and Method**

A total of 1745 sample protease sequences were obtained from the depository of the National HIV Reference Laboratory (NHRL). 1261 samples were of individuals infected with subtype-C HIV, of which 170 were treated with NFV, and 454 samples were of subtype-B, of which 29 patients were treated with NFV.

Using IRR, we separately analyzed each subtype's samples, since C and B subtypes display different resistance mutation patterns (Grossman, et al. 2004, Kantor, et al. 2005, Rhee, et al. 2006) , we separated the IRR analysis to each subtype's samples.

For each of the two subtypes, the amino acid sequences were compared to the subtype consensus sequence (that is, the sequence that is the best representative of the wildtype virus). The subtype consensus sequences were equal in samples from drug-naïve HIV carriers and in the general sample population, reflecting the fact that variants constitute a small minority in the sequence population. After the consensus comparison, a mutations variable set was established by including every diversion from consensus that occurred at least twice

For each mutation, a binary variable was created, indicating for each sample whether the mutation appeared in that sample (positive value, set to +1) or not (-1). Similarly, the condition variable binary value indicates the NFV treatment status for each sample.

A binary frequency matrix of the mutation variables and the condition variable over the sample population was used as input for the IRR algorithm.

The IRR algorithm used a value of 50 as the ridge parameter and $p<0.05$ as the $X^2$ test significance range.

**b) Results**

Graphical representations of the HIV mutation data constructed via IRR are displayed in figures 6a (subtype B analysis) and 6b (subtype C analysis). Associations that showed exceptionally significant $X^2$ scoring ($p<0.00000$ ) and IRR scoring > 0.01 are marked by red edges.

**[suggested place for figure 6]**

A detailed comparison with previously established results and a discussion of the potential significance of new, previously unknown associations is in preparation and will be published soon (Bar-Yaakov, Intrator and Grossman 2011). Briefly, out of the 31 mutations in the IRR-generated networks, 13 have been previously identified as resistance mutations. The IRR network contains several directed paths that match known resistance mutation pathways. In addition, the IRR network shared 24 mutations and a number of pathways with the Bayesian network generated through the analysis of Deforche et al. (2006). Importantly, the IRR method revealed novel associations, such as V15I-D30N, that are biologically plausible and that provide new insights into mechanisms of drug-resistance development.

Our results display some inconsistencies with previous publications. These, too, and implications regarding limitations of the method and possible remedies will be discussed in more detail in future publication (Bar-Yaakov, Intrator and Grossman 2011).

## 6) Summary

This paper introduced Iterative Ridge Regression (IRR), a modification of the dependency network method, which produces a directed graph containing only condition related associations. We have described a robust and computationally efficient estimator algorithm used to uncover such a network.

We showed that the IRR algorithm performs better than current state of the art Bayesian networks when the purpose is to identify conditioned associations.

We also briefly describe, here and in a forthcoming publication, a real life application of the IRR method to the analysis of HIV mutation patterns that emerge in HIV infected patients treated with a particular antiretroviral drug. We have demonstrated that our method can recover from a pool of viral sequences known associations of such treatment with selected resistance mutations as well as documented associations between different mutations. Moreover, the IRR method revealed novel associations that are statistically significant and biologically plausible.

## 7) Appendix

### a) IRR Pseudo code

A pseudo-code for describing the iterative ridge regression (IRR) construction of variable correlation under condition. See text for details of the algorithm steps.

```
/* variables initialization*/
K = number of variables
N = number of samples
Variable List = 1..K variables
Sequence Samples = 1..N sequences
Condition Vector = 1..N boolean (index i is true is sample i was treated)
Variable interaction matrix = KxK real
Variable Matrix = NxK matrix

/* build the variables frequency matrix */
Foreach Sequence Sample i = 1..N
        Foreach Variable j = 1..K
                if Sequence Sample i contains Variable j
                        Variable Matrix(i,j) = 1
                else
                        Variable Matrix(i,j) = -1
                endif
        endforeach
endforeach

/* build the Variables interaction matrix */

Base Weight Vector = apply Ridge Regression to estimate Condition Vector with
                        Variable matrix as data;
for i = 1..K
        Omitted VariableMatrix = NxK-1
        Omitted Variable Matrix(1..N,1..i-1) = Variable Matrix(1..N,1..i-1)
```

```
        Omitted Variable Matrix(1..N,i..k-1) = Variable Matrix(1..N,i+1..k)
        Omitted Weight Vector = apply Ridge Regression to estimate Condition
                                                Vector with Ommited Variable matrix as data;
        Weight Diff Vector = Omitted Weight Vector - Base Weight Vector
        Variables interaction matrix(1..k,i) = Weight Diff Vector
end

threshold = get a p value threshold from the distribution of all interactions

/* network buildup */

Variables Interaction Graph = Create empty Graph matrices with variables as vertices
foreach i = 1..K , j =1..K
        if  Variables interaction matrix(i,j) > threshold  and
                Variable I and j is significant (FDR corrected) under condition
                add directed arc from variable i to j
        endif
endforeach

return Variables Interaction Graph
```

**b) Detailed Algorithm demonstration –**

Follows a simulated example of finding a conditioned association. The left column shows the condition variable value, applied to each of the samples, while the columns to its right show the existence of several mutations in each of the samples.

**[suggested place for table 2]**

At first stage, the 0 values are replaced with -1 for more robust calculations.

Variables C and D are fully correlated, but with low correlation with the condition (-0.169). In the full sample set, variables A and B have low correlation between themselves (0.314) and with the Condition (0.169). Yet it is clearly visible that under the condition, variables A and B are highly correlated, hence creating a pattern. When we omit Variable D we will receive the following weights values after ridge regression:

**[suggested place for table 3]**

The change in the weights of the other variables is relatively small, and not affected by the initial weight of the feature.

However, after omitting one of the pattern's members we will receive the following weights:

**[suggested place for table 4]**

The weight change in the other pattern member (Variable B) is significantly bigger than the change in the other weights, and positively directed.

Iteratively applying the same step over the whole set of variables results in an NxN table consisting of the weight change of each of the features when a different feature omitted.

**[suggested place for table 5]**

Pattern validation using $\boldsymbol{X^2}$ test: after a filtered interaction table, each cell holds a value that describes an "IRR significant" directed interaction between two features (mutations). These filtered interactions will comprise our Region of Interest (ROI), which will be the basis for an independent statistical test – this test will filter out the significant interaction from the previous "IRR significant" interaction.

For each $a \to b$ interaction, we will test whether the $a \to b$ association frequency under the condition is significantly different from its frequency in the general sample's population – i.e. $\boldsymbol{p}(a, b/c) \gg p(a, b)$ . Resulting P values will be adjusted according to the Benjamini-Hochberg algorithm for P value adjustment. We will sort the interactions that pass our threshold P value.

Network construction – we will now construct an interaction network out of the filtered interactions, each mutation will become a vertex, while every filtered $a \to b$ interaction will add a directed edge between the $a$ and $\boldsymbol{b}$ vertices. Since the significant interactions found are between variables $\boldsymbol{a}$ and $\boldsymbol{b}$ the result network will be: $\leftrightarrow \boldsymbol{b}$ .

**c) Analytical proof of the IRR method**

Given a random variable set $S = \{x_1, x_2, \dots x_n\}$ and a condition variable $c$, denote $W^B$ as the basis weight vector of ridge regressing $c$ using S:

$$W^B = \underset{W}{\operatorname{argmin}} E\left(c - \sum_{x_i \in S} w_i x_i\right)^2 + \lambda \parallel W \parallel^2$$

Let $W^j$ be the resulting weight vector, after the ridge regression of $c$ using $S$ excluding $x_j$:

$$W^j = \underset{W}{\operatorname{argmin}} E\left(c - \sum_{x_i \in S/x_j} w_i x_i\right)^2 + \lambda \parallel W \parallel^2$$

In the case where $W_j^B \neq 0$, i.e. $x_j$ has significant part in estimating $c$, we receive:

$$E\left(c - \sum_{x_i \in S} W_j^B x_j\right)^2 + \lambda \parallel W^B \parallel^2 < E\left(c - \sum_{x_i \in S/x_j} {W^j}_i x_i\right)^2 + \lambda \parallel W^j \parallel^2$$

**Lemma** – Denote $\Delta W^j$ to be the weight difference between the two results: $\Delta W^j = W^B - W^j$ .

We wish to prove that $\Delta W^j$ is the coefficients vector that gives the best estimation of $W_j^B x_j$ , while maintaining the regularization expression small:

$$\Delta W^j = \underset{W}{\operatorname{argmin}} E\left(W_j^B x_j - \sum_{x_i \in S/x_j} w_i x_i\right)^2 + \lambda \parallel W^B - W \parallel^2$$

**Proof** – Suppose there is an alternative set of weights: $\widetilde{W^j} \neq \Delta W^j$ that offers better regularized estimation of $W_j^B x_j$ than $\Delta W^j$, hence:

$$E\left( W_j^B x_j - \sum_{x_i \in S/x_j} \widetilde{W_i^j} x_i \right)^2 + \lambda \parallel W^B - \widetilde{W^j} \parallel^2$$

$$< E\left( W_j^B x_j - \sum_{x_i \in S/x_j} \Delta W_i^j x_i \right)^2 + \lambda \parallel W^B - \Delta W^j \parallel^2$$

Denote $\widetilde{\Delta W^j}$ as their weight difference: $\widetilde{\Delta W^j} = \widetilde{W^j} - \Delta W^j$ . After decomposing $\widetilde{W^j}$ :

$$E\left( W_j^B x_j - \sum_{x_i \in S/x_j} (\widetilde{\Delta W_i^j} + \Delta W_i^j)\, x_i \right)^2 + \lambda \parallel W^B - \widetilde{W^j} \parallel^2$$

$$< E\left( W_j^B x_j - \sum_{x_i \in S/x_j} \Delta W_i^j x_i \right)^2 + \lambda \parallel W^B - \Delta W^j \parallel^2$$

Hence:

$$E\left( W_j^B x_j - \sum_{x_i \in S/x_j} \Delta W_i^j x_i + \sum_{x_i \in S/x_j} \widetilde{\Delta W_i^j} x_i \right)^2 + \lambda \parallel W^B - \Delta W^j + \widetilde{\Delta W^j} \parallel^2$$

$$< E\left( W_j^B x_j - \sum_{x_i \in S/x_j} \Delta W_i^j x_i \right)^2 + \lambda \parallel W^B - \Delta W^j \parallel^2$$

Since:

$$E\left( c - \sum_{x_i \in S} W_i^B x_i \right)^2 + \lambda \parallel W^B \parallel^2 < E\left( c - \sum_{x_i \in S/x_j} W^j{}_i x_i \right)^2 + \lambda \parallel W^j \parallel^2$$

After decomposing:

$$E\left( c - \sum_{x_i \in S/x_j} W_i^B x_i + W_j^B x_j \right)^2 + \lambda \parallel W^B \parallel^2 < E\left( c - \sum_{x_i \in S/x_j} W_i^B x_i + \sum_{x_i \in S/x_j} \Delta W_i^j x_i \right)^2 + \lambda \parallel W^j \parallel^2$$

Since we have a better regularized estimation for $W^B{}_j x_j$ :

$$E\left(c - \sum_{x_i \in S/x_j} W_i^B x_i + \sum_{x_i \in S/x_j} \Delta W_i^j \, x_i + \sum_{x_i \in S/x_j} \widetilde{\Delta W_i^j} x_i\right)^2 + \lambda \parallel W^B - \Delta W^j + \widetilde{\Delta W^j} \parallel^2$$

$$< E\left(c - \sum_{x_i \in S/x_j} W_i^B x_i + \sum_{x_i \in S/x_j} \Delta W_i^j x_i\right)^2 + \lambda \parallel W^B - \Delta W^j \parallel^2$$

Therefore it can use $\widetilde{\Delta W^j}$ for better minimization of the condition variable estimation without using $x_j$:

$$E\left(c - \sum_{x_i \in S/x_j} W_i^j x_i + \sum_{x_i \in S/x_j} \widetilde{\Delta W_i^j} x_i\right)^2 + \lambda \parallel W^j + \widetilde{\Delta W^j} \parallel^2$$

$$< E\left(c - \sum_{x_i \in S/x_j} W^j{}_i x_i\right)^2 + \lambda \parallel W^j \parallel^2$$

And this contradicts with the argmin property of $W^j$. ■

In the above procedure, we have proved that by simple reduction of $W^j$ from $W^B$ we receive the best regularized estimation of the omitted variable by its co-variables.

**d) Exact details of the IRR – BNT simulation**

i) Data Simulation

(1) Parameters –

(a) Number of Features – **N**

(b) Number of Samples – **M**

(c) **Condition ratio out of samples** – ratio of samples that are conditioned.

(d) Pattern description –

(i) **Pattern size** – number of features in the pattern.

(ii) **Pattern-Condition ratio** – the ratio of the conditioned samples which contains the pattern.

(e) **Background noise level** – ratio of random data members which have positive value (before pattern induction).

(f) **Sampling noise level** – ratio of random data members, which will have their value inverted.

(2) Data creation

(a) Allocate an **N** by **M** data matrix (Samples).

(b) Allocate an **M** size vector (Condition vector).

(c) Reset the sample matrix and the condition vector values to -1.

(d) Randomly select cells out of the sample's matrix, until selecting total cells ratio of **background noise level**. Inverse their value to 1.

(e) Randomly select cells out of the condition vector, until selecting an amount of cells corresponding to the **Condition ratio** multiplied by the samples set size. These are the **condition indices**, Inverse their value to 1.

(f) Pattern induction –

(i) Randomly select total of **pattern size** indices out of the features indices. These are the **Pattern indices**.

(ii) Randomly select samples, until reaching total of **(Pattern-Condition ratio** * size of **condition indices)** , which their corresponding indices in the Condition vector are positive. These are the **Conditioned samples** associated with the condition vector.

(iii) Change the values of the **Conditioned samples** in the **Pattern indices** to 1.

(g) Sampling noise induction –

(i) Randomly select values until reaching a ratio of sampling **noise level** out of the samples values, and inverse their values.

ii) Single Simulation execution

(1) IRR execution and scoring

(a) Use the sample matrix and the condition vector as an input to the IRR algorithm.

(b) IRR result is the estimated conditioned pattern; use Jaccard Index to evaluate the similarity between the original induced pattern and the estimated IRR result pattern.

(2) Bayesian network (**BNT**) execution and scoring

(a) We used the BNT package (Murphy 2001) for executing a Bayesian network analysis over the simulated data.

(b) We compared the IRR with the K2 Bayesian network algorithms (Cheng, Bell and Liu 1997), showed to outperform or match other BNET algorithms (Leray and Francois 2004)

(c) Input for the BN was created by concatenating the sample matrix and the condition vector – the condition vector becomes another feature in the sample features.

(d) We used the DAG result of the Bayesian Network Power Constructor (BNPC) algorithm (Cooper and Herskovits 1992), for topological sorting used to deduct the variable order, needed as an input for the K2 algorithm.

(e) The scoring was done by searching the connected component of the DAG result that contains the Condition feature.

(f) We assemble a variable set containing the variables in the connected component.

(g) The set is regarded as the result estimated pattern of the algorithm.

(h) We use Jaccard Index to evaluate the similarity between the original induced pattern and the estimated Bayesian Network result pattern.

iii) Comparison method and parameters

(1) In each comparison iteration, we simulated an input data matrix with parameters values as follows –

(a) Number of Features – **N** = 20

(b) Number of Samples – **M** = 1000

(c) **Condition ratio out of samples** = 0.8

(d) Single Pattern induced–

(i) **Pattern size** = 4

(ii) **Pattern-Condition ratio** = 0.2.

(e) **Background noise level** = 0.2

(f) Sampling **noise level** = variable across the simulation

(2) We used the simulated data as an input to the IRR, BNPC and K2 algorithms.

(3) For each set of parameters, we used 100 executions to get an average score on each sampling noise level.

## 8) DISCLOSURE STATEMENT

No competing financial interests exist.

## Bibliography


Banerjee, Onureena, LE Ghaoui, and Alexandre d'Aspremont. "Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data." *Journal of Machine Learning Research* 9 (2008).

Bar-Yaakov, Nimrod, Nathan Intrator, and Zehava Grossman. "Interactions Among PI-Induced Resistance Mutations Revealed by Iterative Ridge Regression ." *preprint*, 2011.

Benjamini, Yoav, and Yosef Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society. Series B* 57 (1995): 289-300.

Cheng, J., D.A. Bell, and W Liu. "An algorithm for Bayesian belief network construction from data." *Proceedings of AI and STAT'97*, 1997: 83–90.

Chickering, D. M., D. Heckerman, and C. Meek. "Large-Sample Learning of Bayesian Networks is NP-Hard." *Journal of Machine Learning Research* 5 (2004): 1287-1330.

Cooper, G., and E. Herskovits. "A Bayesian method for the induction of probabilistic networks from data." *Machine Learning*, no. 9 (1992): 309-347.

Deforche, K., et al. "Analysis of HIV-1 pol sequences using Bayesian Networks: implications for drug resistance." *Bioinfomatics* 22 (2006): 2975-2979.

Edwards, D. M. *Introduction to Graphical Modelling, 2nd ed.* New York: Springer, 2000.

Friedman, J, T. Hastie, and R. Tibshirani. "Sparse inverse covariance estimation with the graphical lasso." *Biostatistics* 9 (2008): 432-441.

Friedman, J., T. Hastie, and R. Tibshirani. "Regularization paths for generalized linear models via coordinate descent." *Journal of Statistical Software* 33 (2010): 1–22.

Friedman, N., M. Linial, I. Nachman, and D. Pe er. "Using Bayesian Networks to Analyze Expression Data." *Journal of Computational Biology* 7 (2000): 601-620.

Grossman, Z., et al. "Mutation D30N is not preferentially selected by human immunodeficiency virus type 1 subtype C in the development of resistance to nelfinavir." *Antimicrobial agents and chemotherapy* 48 (2004): 2159–2165.

Heckerman, David, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. "Dependency Networks for Inference, Collaborative Filtering, and Data Visualization." *Journal of Machine Learning Research* 1 (2000): 49-75.

Hoerl, AE. "Application of ridge analysis to regression problems." *Chemical Engineering Progress*, no. 58 (1962): 54-59.

Jain, A. K., M. N. Murty, and P. J. Flynn. "Data Clustering: A Review." *Computing Surveys* 31 (1999): 264–323.

Jensen, FV. *An Introduction to Bayesian Networks.* Berlin: Springer, 1996.

Kantor, R., et al. "Impact of HIV-1 Subtype and Antiretroviral Therapy on Protease and Reverse Transcriptase Genotype: Results of a Global Collaboration." *PLoS Medicine* 2 (2005): 325-337.

Leray, P., and O. Francois. *BNT structure learning package: documentation and experiments.* Laboratoire PSI, Universite et INSA de Rouen, 2004.

Markowetz, Florian, and Rainer Spang. "Inferring cellular networks – a review." *BMC Bioinformatics* 8 (2007): S5.

Meinshausen, N., and P. Buhlmann. "High-dimensional graphs and variable selection with the Lasso." *Annals of Statistics* 34 (2006): 1436-1462.

Murphy, Kevin. *Bayes Net Toolbox for Matlab.* 2001. http://people.cs.ubc.ca/~murphyk/Software/BNT/bnt.html.

Nikulin, M.S. "Chi-square test for continuous distributions with shift and scale parameters." *Theory of Probability and Its Applications* 18 (1973): 559–568.

Pearl, J. "Comment: Graphical Models, Causality and Intervention." *Statistical Science* 8 (1993): 266-269.

Pearl, J. "Fusion, propagation, and structuring in belief networks." *Artificial Intelligence* 29 (1986): 241–288.

Pearl, Judea. *Causality: Models, Reasoning, and Inference.* London: Cambridge University Press, 2009.

Pe'er, D., A. Regev, G. Elidan, and N. Friedman. "Inferring subnetworks from perturbed expression profiles." *Bioinformatics* 17 (2001): S215-S224.

Rhee, S.Y., et al. "HIV-1 pol mutation frequency by subtype and treatment experience: extension of the HIVseq program to seven non-B subtypes." (AIDS) 20, no. 5 (2006): 643-652.

Shafer, Robert W. "Rationale and Uses of a Public HIV Drug-Resistance Database." *Journal of Infectious Diseases* 194 (2006): s51-s58.

Sing, T., et al. "Characterization of novel HIV drug resistance mutations using clustering, multidimensional scaling and SVM-based feature ranking." In *Knowledge Discovery in Databases: PKDD 2005*, by A., Torgo, L., Brazdil, P., Camacho, R., Gama, J. (Eds.) Jorge, 285:296. New York: Springer, 2005.

Tibshirani, Robert. "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society. Series B* 58 (1996): 267-288.

Tychonoff, A. N., and V. Y. Arsenin. *Solution of Ill-posed Problems.* Washington: Winston & Sons, 1977.

Zou, Hui, and Trevor Hastie. "Regularization and variable selection via the elastic net." *Journal of the Royal Statistical Society: Series B* 67, no. 2 (2005): 301-320.

## Tables and Figures

| Noise level | IRR | K2 |
|---|---|---|
| 0 | 0.9340 | 0.7220 |
| 0.02 | 0.8980 | 0.6327 |
| 0.04 | 0.9163 | 0.6750 |
| 0.06 | 0.9180 | 0.7153 |
| 0.08 | 0.8828 | 0.6020 |
| 0.1 | 0.8923 | 0.6173 |
| 0.12 | 0.9012 | 0.6504 |
| 0.14 | 0.8593 | 0.5330 |
| 0.16 | 0.8418 | 0.5280 |
| 0.18 | 0.8160 | 0.6558 |
| 0.2 | 0.8001 | 0.3687 |
| 0.22 | 0.7030 | 0.3755 |
| 0.24 | 0.6382 | 0.3135 |
| 0.26 | 0.5423 | 0.2052 |
| 0.28 | 0.4298 | 0.2015 |
| 0.3 | 0.3153 | 0.1360 |
| 0.32 | 0.2223 | 0.0965 |
| 0.34 | 0.1480 | 0.0492 |
| 0.36 | 0.1093 | 0.0827 |
| 0.38 | 0.0470 | 0.0329 |
| 0.4 | 0.0295 | 0.0170 |

**Table 1: Mean Jaccard index scoring over 100 iterations of the IRR and Bayesian Network K2 algorithms along various noise levels**

| Condition | Var. D | Var. C | Var. B | Var. A |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |

**Table 2: Example data set of four random variables (Variables A, B, C, D) and a condition variable, over 12 samples. Positive sample value is marked with gray.**

| Variable | Ridge Reg. Weights | Weights After Omission of Variable D | Weight Difference |
|---|---|---|---|
| A | 0.0302 | 0.0302 | 0.0000 |
| B | 0.0302 | 0.0302 | 0.0000 |
| C | 0.0016 | 0.0018 | 0.0002 |
| D | 0.0016 | N.A. | N.A. |

**Table 3: Weights of the variables after ridge regressing the condition variable, and after ridge regressing the condition variable without variable D.**

| Variable | Ridge Reg. Weights | Weights After Omission of Variable A | Weight Difference |
|---|---|---|---|
| A | 0.0302 | N.A. | N.A. |
| B | 0.0302 | 0.0319 | 0.0017 |
| C | 0.0016 | 0.0008 | -0.0008 |
| D | 0.0016 | 0.0008 | -0.0008 |

**Table 4: Weights of the variables after ridge regressing the condition variable, and after ridge regressing the condition variable without variable A.**

| Ridge Regression Weights | Weight Diff | | | | |
|---|---|---|---|---|---|
| All Variables | Variable | Var. A omitted | Var. B omitted | Var. C omitted | Var. D omitted |
| 0.0302 | A | 0 | 0.0016677 | 0 | 0 |
| 0.0302 | B | 0.00167 | 0 | 0 | 0 |
| 0.0016 | C | 0 | 0 | 0 | 0.000279 |
| 0.0016 | D | 0 | 0 | 0.0002792 | 0 |

**Table 5: Weights of the variables after ridge regressing the condition variable after omitting each of the variables. IRR Significant associations are marked in gray.**

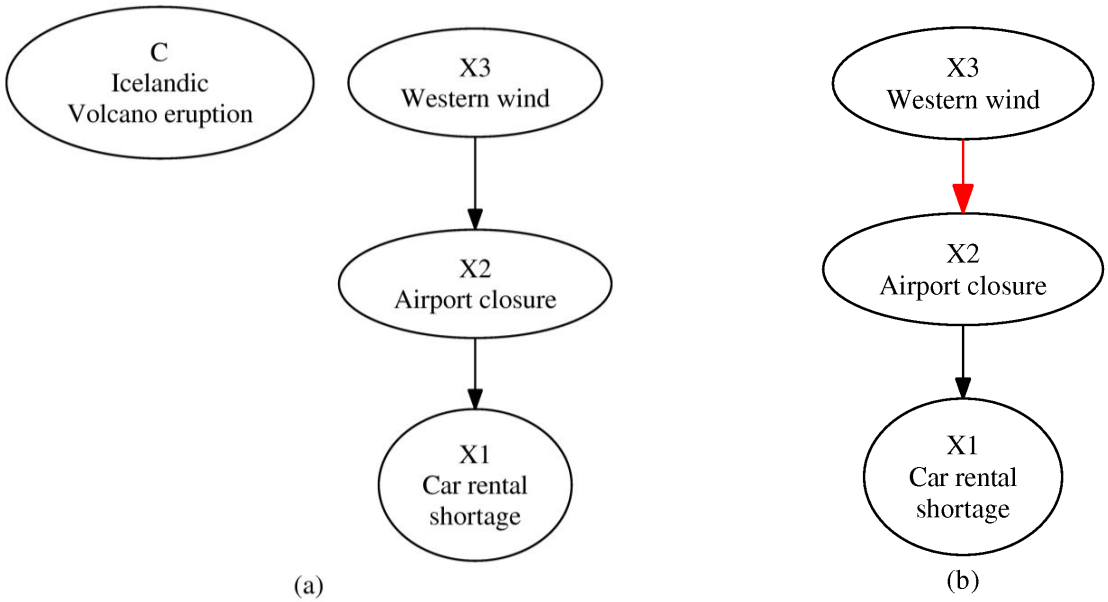


**Figure 1 – Graphical representation of the dependencies between three random variables. (a) is the general case and (b) cases under the condition c (Icelandic Volcano eruption) . The edge $x_3 \rightarrow x_2$ is unique to cases under Condition c.**

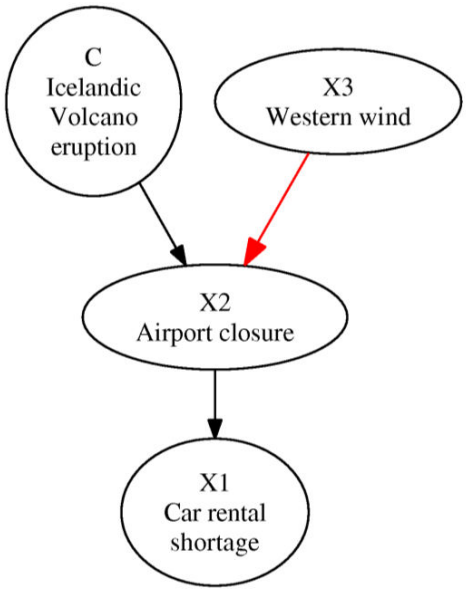


**Figure 2 – Bayesian Network representing the joint distribution mapping of X1, X2, X3 and the condition C. The edge $x_3 \rightarrow x_2$ was retrieved using the BNT.**

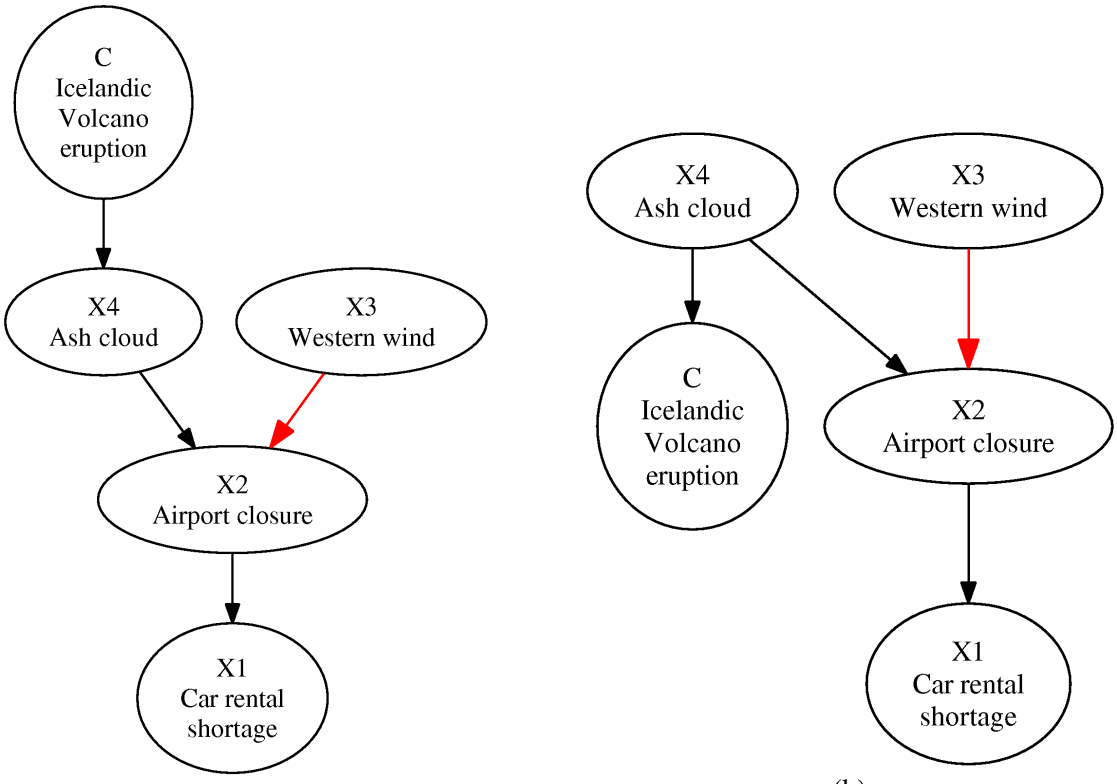


**Figure 3 – Alternate Bayesian Network representation of our data. These networks maintain the d-separation criteria between the variables, though now the novel conditioned association ($x_3 \rightarrow x_2$) should be searched within the whole connected component containing C.**

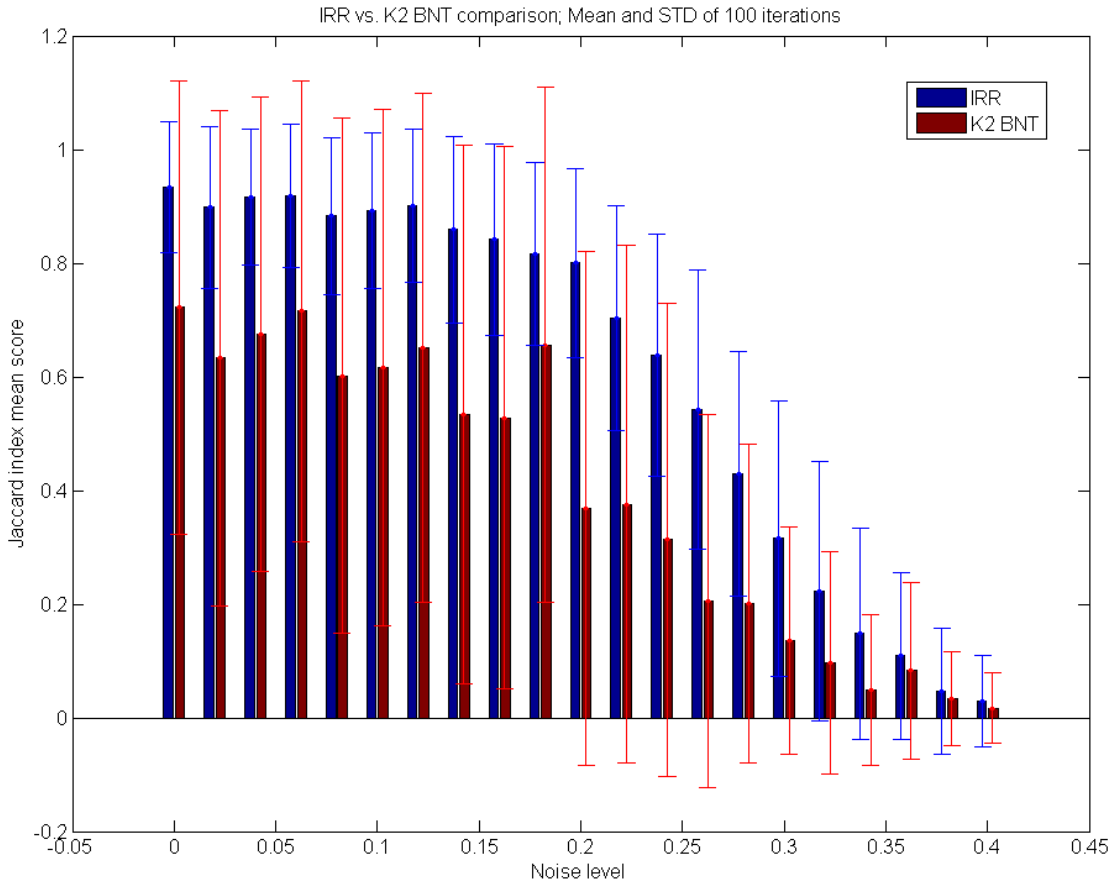


**Figure 4: Graphic representation of the Jaccard index scoring mean and STD of the IRR and Bayesian Network K2 algorithms, over 100 iterations. The IRR algorithm significantly outperforms the Bayesian Network K2 algorithm ($p<0.01$). The STD values show more stable performance by the IRR as compared with the K2 algorithm.**

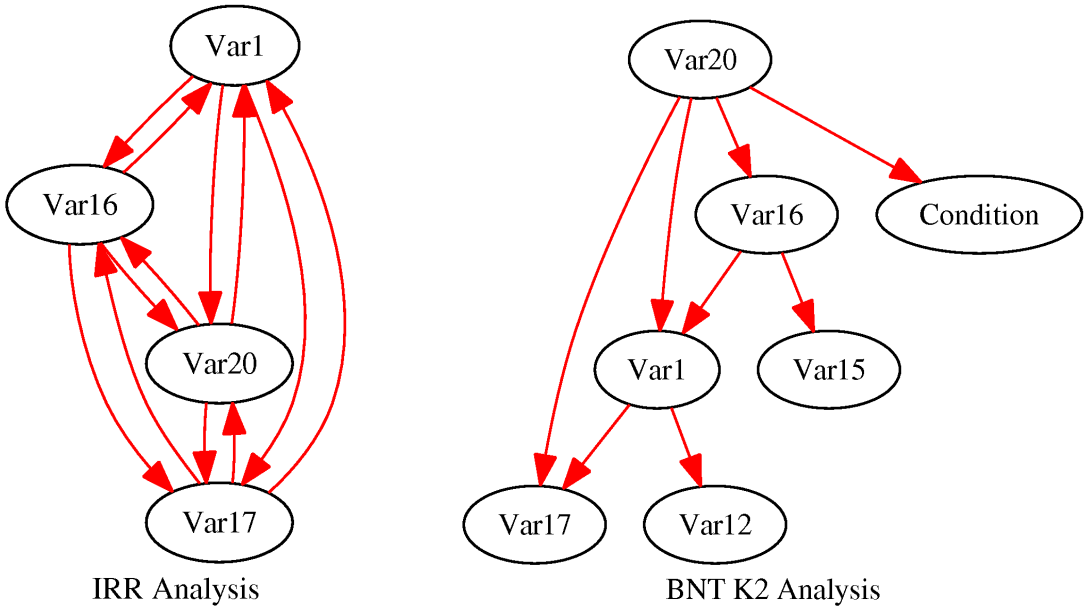


**Figure 5: Graphical result of pattern locating. The pattern contains 4 variables (Var1, Var16, Var17, Var20) in a 20 random variables data set. The IRR has fully located the induced pattern, while the BNT has also located two randomly correlated variables (Var12, Var15), not associated in the pattern under the condition.**

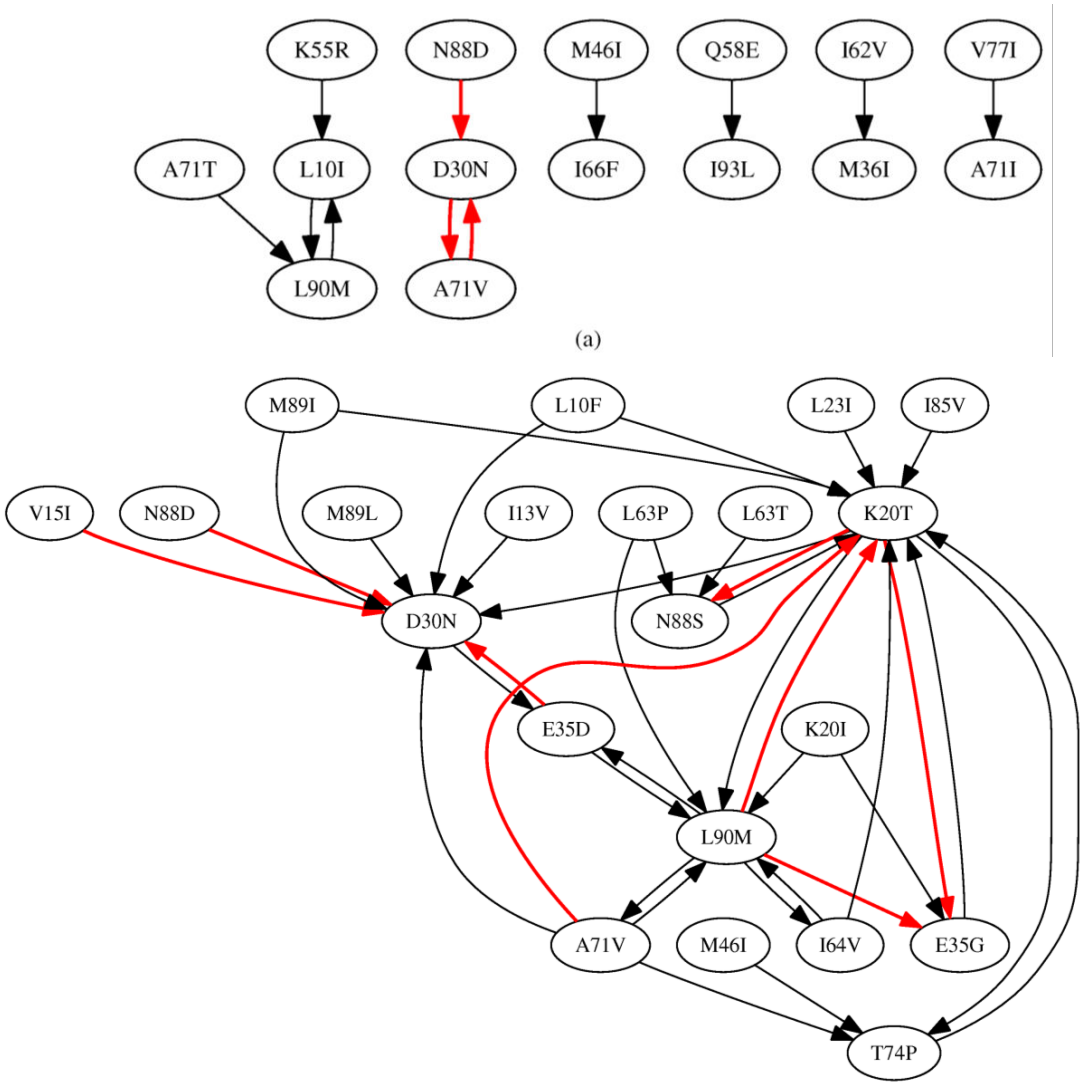


**Figure 6: IRR graphical result of amino acid mutations associations found in subtype B (figure 6a) and subtype C (figure 6b) HIV protease sequences, conditioned by Nelfinavir (NFV) treatment. Red edges suggest exceptionally significant associations with high IRR and $X^2$ scoring. Edge directions are such that the presence of the edge source will increase the probability of the edge target.**