Author's response to reviews

Title: Logic Minimization and Rule Extraction for Identification of Functional Sites in Molecular Sequences

Authors:

Raul Cruz-Cano (raulcruz@umd.edu)
Mei-Ling T Lee (mitlee@umd.edu)
Ming-Ying Leung (mleung@utep.edu)

Version: 2 Date: 5 March 2012

Author's response to reviews: see over
March 4, 2012

Subject: Submission of revised manuscript MS: 4280528655743347
Title: Logic Minimization and Rule Extraction for Identification of Functional Sites in Molecular Sequences
Authors: Raul Cruz-Cano, Mei-Ling T Lee and Ming-Ying Leung

Dear BioData Mining Editorial Team:

Thank you for your email on January 4, 2012. The editor’s instructions and the reviewers’ suggestions have been carefully considered. We have revised the manuscript with additional experiments and data according to their advice. The point-by-point description of the changes made in response to their comments is included in the next pages.

We would like to take this opportunity to thank the reviewers for their constructive criticisms in helping to improve the manuscript. We are looking forward to your favorable decision.

Sincerely,
Raul Cruz-Cano
Reviewer #1 report
Title: Logic Minimization and Rule Extraction for Identification of Functional Sites in Molecular Sequences
Version: 1 Date: 23 November 2011
Reviewer: hui liu
Reviewer's report:
Commit:
• Major Compulsory Revisions
  1. Is the question posed by the authors new and well defined?
the question is explicit and popular
  2. Are the methods appropriate and well described, and are sufficient details
provided to replicate the work?
The authors apply logic minimization techniques to classification and rule
discovery. And even ESPRESSO has limitation that 22 input variables should
been selected firstly. So SVM is adopted for reducing the number of variables.
Then the SVM, along with ANN, provide performance benchmarks to compare
with the logic minimization approach for classification. The proposed logic
minimization method provides sets of rules which can predict TFBS on human
DNA and O-glycosylation sites on proteins with sensitivity and PPV comparable
to those from ANN and SVM. Furthermore, the logic minimization method has the
additional capability of generating interpretable rules that allow biological
scientists to correlate the predictions with other experimental results and form
new hypothesis for further investigation.
Actually there are several methods designed to extract rules from trained
classifier, for example SVM and ANN as mentioned in the manuscript. Then in
order to validate performance of generating interpretable rules, the authors
should train SVM and ANN and extract rules to compare with the proposed
method. On the other hand, to test the classify performance, the SVM or ANN
should used the original features instead of the selected 22 features.

After considering the reviewer's comments, we have incorporated the following points in
the revised manuscript to clarify the problems of using SVM and ANN for rule extraction.
As noticed in [35], the lack of software for efficient rule extraction by the SVM approach
is an issue that remains as a challenge. On the other hand, it is possible to find software
for extracting rules from ANN. “Recognize, Predict, Forecast!™” is a software which
uses Self-Organizing Neural Networks (SONN) to solve classification and regression
problems [28]. This software presents the ANN as a graph helpful for understanding
what input characteristics are used by the ANN in making a prediction. Moreover, in
trained SONNs, only the essential connections are drawn for making the rule extraction
task possible for a scientist with a basic understanding of ANN.

One difficulty in using this software is that it requires either 50% or 75% of all the
available data for training. In the case of the TFBS, it would be extremely expensive to
study half or three quarters of the almost 20,000 possible TFBS and then make
predictions in the remaining candidates. Although “Recognize, Predict, Forecast!™” is
useful for validating the extracted rules, this limited testing capability makes it unsuitable
for performance comparisons with the feed-forward ANN, SVM, and the proposed method. In any case, SONNs are designed to be understandable at the expense of accuracy; hence, it is reasonable to assume that their performance for more realistic training test sets would not be better than the performance of the feed-forward ANNs using only 150 training examples (see Table 9).

For the TFBS, using all the original features during training, an SONN with three layers selects to connect only the inputs 26, 29, 33 and 40 with positive values. This is basically rule 2 of Table 8.

For the glycosylation data, the sizes of the training datasets, although larger, were not extremely dissimilar from those used for the systems mentioned in Table 10. The SONN was unable to handle the S O-glycosylated dataset perhaps due to the large number of variables for such a small number of positive examples. Despite using half of the dataset with all the variables for training, its sensitivity was only around 45%. Adding layers to the SONN did not help its performance. For this reason, we decided not to pay attention to the rules produced by the SONN for this dataset. The results for the T O-glycosylated dataset were more encouraging, providing a sensitivity of around 72% and PPV of 77%. These numbers are in the same range as those presented in Table 10. A SONN with four layers selected inputs 15, 38, 99,101, 143, 164, 206, and 227 as the significant variables. All of these variables appear in Table 12. Even better, the architecture of the network shows that input 38, 143, 164, and 227 were associated before being conglomerated with the rest of the relevant inputs. This set of variables strongly resembles most of the highest ranked variables listed in Table 14.

These results indicate that the SONN method confirms the rules extracted by logic minimization for the TFBS and for the T O-glycosylated datasets. It is also worth mentioning that we manually checked dozens of examples that are correctly classified as TFBS or as O-glycosylated S and T sites and verified that they actually follow the rules obtained by the logic minimization method.

As mentioned on page 21, both ANN [26] and SVM [30] have been previously applied to the O-glycosylation datasets without using any feature selection. The results in [30] indicate that SVM provides better prediction accuracies than ANN and hence it is discussed in several occasions in the section corresponding to this problem. It is also worth noting that the inclusion of all the features would not contribute to gaining new knowledge about this biological case study, i.e. there is no better understanding of which combination of features are significant in the O-glycosylation process or the differences between the O-glycosylation in T sites and S sites.

3. Are the data sound and well controlled?
Yes
4. Does the manuscript adhere to the relevant standards for reporting and data deposition?
no. need additional necessary experiments
Additional experiments have been performed as described above. These experiments have confirmed the most significant rules discovered by the proposed method for the TFBS and the T O-glycosylated datasets.

5. Are the discussion and conclusions well balanced and adequately supported by the data?
No. refer to 2
The discussions and conclusions of the paper have been modified to reflect the additional information obtained from the new set of experiments.

Once again, thank you for the suggestion; the additional experiments did improve the quality of the paper considerably.

6. Do the title and abstract accurately convey what has been found?
Yes
7. Is the writing acceptable?
Unreadable
The writing has been reviewed in details and corrected extensively in consultation with technical writers. The number of modifications to the written English is so vast that it would be impractical to highlight all of them in the revised manuscript. Instead only the material added in response to the reviewers’ comments in the “Reviewer’s report” sections has been highlighted.

Level of interest: An article whose findings are important to those with closely related research interests
Quality of written English: Not suitable for publication unless extensively edited
Statistical review: No, the manuscript does not need to be seen by a statistician.
Declaration of competing interests:
I declare that I have no competing interests
Reviewer #2
Title: Logic Minimization and Rule Extraction for Identification of Functional Sites in Molecular Sequences
Version: 1 Date: 31 December 2011
Reviewer: Jesus Aguilar-Ruiz
Reviewer's report:
The paper deals with classification and rule discovery in bioinformatics. The authors apply logic minimization, first encoding the original data into binary format. They apply the method to two problems: TFBS and O-glycosylation prediction. Results are compared to that of ANN and SVM. These are comparable. The advantage of the method resides in the form of results (rules), much more intelligible.

1) Pages 4 and 5. Applying Rule 2 from Table 2 to pattern 16 of Table 1, the result is 1, when it should be "unknown". This means that "unknown" could be interpreted as "doesn't matter", and I am not sure that this is correct, because the valid output might be 0.

Yes, the valid output could be 0 but given the rules extracted from the examples with known output (Table 3), it was determined that the best guess for the output corresponding to this particular input pattern is 1.

It is a situation similar to an ANN dealing with an unseen input pattern. In theory the output could be any value in the range of the output variable but the ANN generalizes the information obtained from the training set (i.e., the input patterns with known output) and takes an educated guess about what the most likely value for this output would be.
[see page 5 of the revised manuscript]

2) Page 8. Authors use SVM for feature selection. Why this technique? Any justification?
Yes, this point needs clarification. The main reason for the selection of SVM is its dual nature as classifier/feature selector, a similar characteristic shared by our proposed method which is a classifier/rule extractor. Additional reasons are:
- Existing software is readily available.
- SVMs are a well-respected and popular technique.
- SVMs are fast and computationally inexpensive.
- SVMs provide deterministic and therefore consistent results.
[see page 9]

3) Discretization is usually a way of losing information. How can the authors guarantee that discretization is not affecting the results?
This complex matter requires careful examination for each application. In the same way that the standardization or the use of logarithmic scale might or might not affect the results of some experiments for a given classifier, discretization might or might not affect the results of the proposed algorithm for a given dataset. In this revised manuscript, we have included discussions on some
alternative rule extraction approaches such as self-organizing neural network (SONN). Results from the SONN approach and the findings presented in [18] confirm that our method is able to produce rules that accurately describe the datasets for the two biological problems in this manuscript. This suggests that the method can be used to model real-life problems with characteristics similar to the case studies presented in the paper.

[see page 11]

4) Authors propose to move the threshold to remove contradictions in the output. Is it possible to assure that this threshold can always be found?
This is an important point requiring a more complete analysis. In theory, it is easy to build an example in which the threshold cannot be found. Suppose that we have a variable $X$ with values 0, .5, and 1. Also assume that the classification for these examples is 0, 1, and 0, respectively. In this case, there is no threshold which can divide the classes based on variable $X$. However, real-life applications usually have many variables, providing a much better chance to separate the classes.

Also notice that:
- Even if the threshold cannot be found, the user will gain a better understanding of the dataset. Now he/she would know that the variable $X$ does not have direct relationship with the desired output.
- The user can try different transformations of the data to solve the problem. For example if instead of $X$ we use $f(X) = |X - .5|$ then $f(0) = .5$ (class=0), $f(.5) = 0$ (class=1) and $f(1) = .5$ (class=0), now a threshold of $.25$ would separate the classes just fine. Another alternative would be to separate the variable in more than 2 partitions using more than one threshold. For example we can use three binary variables to represent if $X < .25$, $.25 \leq X \leq .75$ and $X > .75$.

[see page 10]

5) Page 14. Is the encoding used efficient enough? Have the authors thought on natural or Gray binary encoding as these are shorter?
Yes, other encoding schemes were considered but we reached the conclusion that, while these encoding schemes would lead to shorter training and test samples, the main advantage of the proposed algorithm, i.e., its ability to provide rules that can be interpreted immediately by a scientist who might not have training in digital design, mathematics, machine learning, etc., would be lost. In other words, there would not be a clear relationship between each binary variable and the natural language term required to express the corresponding nucleotide or amino acid at that position.
Let's examine the example shown below:

<table>
<thead>
<tr>
<th>Input X</th>
<th>Binary Encoding 1 (bit per variable)</th>
<th>Natural Binary Encoding</th>
<th>Output Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>001</td>
<td>00</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>010</td>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>100</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

If this example is encoded using three binary variables then the digital minimization system will provide the rule: IF the right most bit is 1 or the middle bit is 1 THEN the output is 1. This immediately can be interpreted as IF X is A or X is B then Y is 1. For the natural binary encoding the resulting rule is IF the left-most bit is zero THEN the output is 1. Notice that for an user untrained in digital logic it would not be obvious that this bit represents the cases in which X is equal to A or B. This example is very simple and the meaning of the left-most bit in the natural binary encoding is easy to find but for problems with dozens of variables, the user would need a deep understanding of digital logic to translate the extracted rules into meaningful terms.

In future developments of the software, we can consider using these encoding schemes and adding a decoder to interpret the rules created by the system.

[see page 17]

6) The experimentation is based on some sort of cross-validation for TFBS prediction, and hold-out for Glycosylation. Is this appropriate in terms of maintaining a balanced criterion for experimentation?

Yes, we believe that this is the case because these different criteria are the best approximation to the situations researchers studying each of the problems would face. In the case of the O-glycosylation data, the researchers know that only S or T can be O-glycosylated; hence, they can focus only on pieces of protein that have an S or T in the center. The TFBS do not have this limitation and a sliding window which considers all the possible sub-strings of DNA as potential TFBS should be employed.

[see page 21]

7) Page 15. There are several "magic" numbers in the paper. For example, the sliding windows of size 56. Why 56? Why not 60? Justify.

The TFBS dataset used in this research consists of 73 experimentally verified transcription factor binding sites with a length of 14 nucleotides. Each of the 14 nucleotides has four possible values (A, C, G, or T); hence each TFBS is represented by 56 binary variables (14 * 4 = 56 bits). This has been clarified on page 17.

8) How are the 22 top variables selected by SVM-RFE? Clarify.

As mentioned on page 11, the process of repetitively eliminating from the dataset the variables producing the smallest change in the generalization capability of a SVM is called recursive feature elimination (RFE). Basically, the first variable
selected is the variable that reduces the generalization capability of the SVM the most. Once that this variable is identified and removed from the dataset the process is repeated leading to the second most significant variable. We selected the top 22 variables in this iterative manner. The change in the generalization capability of the SVM can be measured by keeping track of the changes in the magnitude of its margin. The margin is the distance between the decision surface provided by the SVM and its closest data point. Reference [1] provides a more detailed explanation of the process.

[see pages 12]

9) Page 16. It is mentioned that ESPRESSO found that only the top 7 variables were necessary. However, in Table 9 only the top 6 variables are shown. Is this consistent?
This is a typo in Table 9, Tables 7 and 8 are correct, seven variables are necessary for ESPRESSO. Thanks for catching this mistake.

10) Page 17. The number of positive predictions reveals the difficulty of the problem. However, 250 over 7 true TFBS is not an excellent result. Yes, there is room for improvements in the performance of the logic-minimization algorithm, but we note the following points:
1) The ANN and the SVM provide comparable prediction accuracy without generating any additional knowledge, in the form of rules, which can help elucidate sequence characteristics around TFBS.
2) The 250 predictions are just a small fraction of the approximately 20,000 possible TFBS that were evaluated by the logic-minimization method.
3) In the revised manuscript, the extracted rules for TFBS have been confirmed by an alternative method (SONN).

We will touch on the matter of the performance of the logic minimization method again in the answer to comment 18.

11) Page 19. Are the 11 amino acids encoded as a 21-bit long binary sequence? Not 20-bit long? I also have doubts about this very long encoding method. The extra bit is used to represent the case “missing value” in the data (page 22). Since there are no missing values in the TFBS case study, this additional bit was not required. Again, the long encoding is necessary to preserve the most distinctive advantage of the proposed method: its immediate interpretability. In fact, our investigation of the O-glycosylation dataset was motivated by inquiries from a group of biological scientists interested in finding rules to characterize the flanking amino acid sequences around O-glycosylation sites. The advantage of immediate interpretability will be discussed again in the answer to point 17. [see page 22]

12) The top 20 (of the 231 variables) selected by SVM-RFE were used. Why the top 20? It is said that the balance between performance and computational resources justify this decision. I guess that it is necessary to run the method...
several times to find a good balance. How can it be measured that balance for a new unseen dataset? And, specifically, what does "good balance" mean?

The point of balance will depend on each user. A user with a high-performance computer can use many more variables than a user with a standard desktop who needs to find the rules quickly. Each individual user of the algorithm will have to decide what kind of compromise works best for him/her. Preliminary studies with a few variables can lead to accurate estimations of the time and resources needed for larger versions of the problem at hand.

It is also worth considering that by keeping only this relatively small number of variables for the rule extraction, it is easier to observe all of them at once and hence possible to spot unusual patterns, such as the “10 shared variables” mentioned in comment 17. A similar argument has been suggested in [5].

13) The performance for the SVM using the top 20, 30, 40 or 50 variables are similar (stated by the authors). Can this be guaranteed for any dataset? The algorithm described in Figure 1 can be applied unchanged regardless of the number of significant variables. However, its implementation might have to be adapted to the dataset being studied. For example, a different logic minimization software (more powerful than ESPRESSO) might be required. In any case, the essence of the algorithm can remain unchanged regardless of the number of significant variables.

14) Page 20. 45% of the scores? Why this number? Why hold-out for this experimentation? Thanks for bringing up this question. This point needs to be clarified. The 45% is not meant to be interpreted as an optimal number in any way. It just shows how the scores of the rules can be used to tune, up to a certain point, the balance of PPV and sensitivity according to the needs of the user. Users who wish for higher PPV and can tolerate lower sensitivity would select a higher percentage to be considered a positive prediction, while those who wish to have a higher sensitivity can lower the score needed to be considered a positive prediction. An explanation has been added on page 23.

15) It is surprising that there is no mention in the text (except the figures in the table) to the PPV for ANN and SVM (about 95%), much better than those for Digital Rules (80% and 66%). Any justification? Yes, this omission has been addressed. Nevertheless keep in mind that, given the superior sensitivity of the proposed method for those same cases, this fact does not mean that SVM or ANN offer a better performance than the logic minimization method. Users who prefer high sensitivity would be better served by it. Moreover, if the user is also interested not only in the prediction results, but
also in gaining a better understanding of the problem at the same time, the rules provided by logic minimization can be immediately interpreted while the SVM and ANN are not. In the latter cases, the user would have to find a separate rule extraction method; learn how to use it, etc.

[see page 24]  

16) Page 21. The number of relevant binary variables are correlated with the number of training examples. This is a statement of the authors. Where is the proof? This statement was based on a series of experiments, not on a mathematical proof. In these experiments, binary rules were generated at random and training and test sets were created using these rules. Then new variables with random values were inserted to all the examples in the dataset. We discovered that the number of variables correctly identified by the SVM-RFE was positively correlated with the number of training examples provided for it. Since this set of complex preliminary experiments would require a number of pages to be described we prefer to withdraw this statement from the current paper and save it for a more in-depth discussion in a future manuscript. If the reviewers would like to examine the results of these experiments, we can make them available as supplementary files.

17) Tables 11 and 12 share 10 variables (50%). What is the meaning and the impact of it? The sharing of these variables means that the 10 common variables are always involved in the O-glycosylation process, while the rest of the variables can help to understand what makes the O-glycosylation process different at an S site or a T site. This serves as an example showing the value of interpretability offered by the proposed method. [see page 25]  

18) Page 23. "...PPV comparable to those from ANN and SVM." Not always true. In general, the paper needs a great improvement in the sense that the experimentation and results do not prove that the proposed method provides good performance for any dataset. In the revised manuscript, we have added additional experimental results and edited it extensively to explain the importance of our proposed method and how it improves over the existing techniques. We also need to publish our study for these real-life datasets in order to present these problems to the computational intelligence (CI) community. We expect that good performance will be achieved once researchers with CI background begin to study these problems and further improve the machine learning methods employed.

Furthermore, results from the SONN approach and the findings reported in [18] confirm that the proposed method provides useful insights into the case studies. Its relative ease of implementation using existing software tools and its ability to generate interpretable rules makes the proposed method worthy of divulgence.
Level of interest: An article of limited interest
Since the algorithm presented in this paper can be adapted to different problems, it should be of interest for the community of scientists interested in finding new ways to mine useful information (e.g., extract rules) from biological datasets. A search in Google Scholar for the term “rule extraction” leads to 12,400 hits (1000+ since 2011). Many of these articles have been published in high impact journals dedicated to data mining and soft computing, so this community is active. With the easy interpretability of our results, we expect that this article will promote interests in applying machine learning methods among biologists who might not have training in mathematical analysis or digital design. On the other hand, we would also like to encourage efforts in enhancing machine learning methods for applications in biological data mining. We hope that this paper will contribute to both of these aims.

Quality of written English: Not suitable for publication unless extensively edited
Yes, the writing has been reviewed in details and corrected extensively in consultation with technical writers. The number of modifications to the written English is so vast that it would be impractical to highlight all of them in the revised manuscript. Instead only the material added in response to the reviewers’ comments in the “Reviewer’s report” sections has been highlighted.

Statistical review: No, the manuscript does not need to be seen by a statistician.
Declaration of competing interests:
I declare that I have no competing interests for publication unless extensively