Probabilistic Prediction of Protein Phosphorylation Sites Using Kernel Machines

Mark Menor  
Department of Information and Computer Sciences  
University of Hawai’i at Mānoa  
1680 East-West Road  
Honolulu, HI 96822  
mmenor@hawaii.edu

Guylaine Poisson  
Department of Information and Computer Sciences  
University of Hawai’i at Mānoa  
1680 East-West Road  
Honolulu, HI 96822  
guylaine@hawaii.edu

Kyunghim Baek*  
Department of Information and Computer Sciences  
University of Hawai’i at Mānoa  
1680 East-West Road  
Honolulu, HI 96822  
kyungim@hawaii.edu

ABSTRACT
Phosphorylation is an important post-translational modification of proteins that is essential to the regulation of many cellular processes. The in vivo and in vitro discovery of phosphorylation sites is an expensive, time-consuming and laborious task. In this preliminary study, we assess the viability of using our proposed probabilistic Classification Relevance Units Machine (CRUM) for in silico phosphorylation site prediction. We conduct a comparison with the popular Support Vector Machine (SVM) and the Relevance Vector Machine (RVM) that, unlike the SVM, has not been applied to phosphorylation site prediction. The resulting CRUM and RVM predictors offer comparable predictive performance to the SVM. The main advantages of CRUM and RVM over the SVM are:

1. An estimation of the posterior probability of the site being phosphorylatable, providing biologists an important measurement of the uncertainty of the prediction.
2. A more parsimonious model, leading to a reduction in prediction run-time that is important in predictions on large-scale data.

Furthermore, the CRUM training algorithm has lower run-time and memory complexity and has a simpler parameter selection scheme than the RVM learning algorithm. Therefore we conclude that the CRUM is the most viable kernel machine for probabilistic prediction of protein phosphorylation sites.

Categories and Subject Descriptors  
I.3 [Life and Medical Sciences]: Biology and Genetics;  
I.5 [Pattern Recognition]: Models; I.5.1 [Models]: Statistical

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

March 25-29, 2012, Riva del Garda, Italy.  
Copyright 2011 ACM 978-1-4503-0857-1/12/03 ...$10.00.

General Terms
 Algorithms, Experimentation, Performance

Keywords
Phosphorylation, Classification, Kernel machine

1. INTRODUCTION
Phosphorylation of proteins is essential to the regulation of many cellular process, including metabolism, growth, apoptosis, and membrane transport [15]. Phosphorylation is a reversible post-translational modification where enzymes called kinases attach phosphate groups to particular amino acids in a protein sequence [3]. During dephosphorylation, on the other hand, enzymes called phosphatases remove attached phosphate groups from the protein. The presence of phosphate groups on the protein alters the 3-D structure of the protein, thereby catalyzing or inhibiting the protein’s function. Phosphorylation is then analogous to an off-and-on switch, turning certain cellular processes on and off over time. Determining which sites of a protein sequence is phosphorylatable is thus important in the understanding of biochemical processes, particularly of diseases, making kinases a major target for drug design [6].

Protein phosphorylation sites are identified via in vivo or in vitro experiments, with the mass spectrometry approach being popular in recent years [18]. The sites identified by these experiments are listed in the annotations in a variety of protein databases, including Swiss-Prot [4], PhosphoBase [19], and PhosphoSite [14]. Regardless of the method used, this is an expensive, time-consuming and laborious task [18]. To help mitigate this computational problem, in silico, methods that identify highly possible sites have been an active research area for the last 10 years.

The majority of existing phosphorylation site prediction tools use methods from supervised learning. These include support vector machines (SVM) used by KinasePhos [28], Musite [11], and PPRED [1], and a variety of neural network methods used by NetPhosK [3] and GANNPhos [24], among others. These tools differ in their choice of site features and encodings, and their choice of learning algorithms. Due to the difficulty in constructing an independent testing dataset and the lack of standard datasets, in this study we elect to compare different learning methods on the phosphorylation site prediction problem rather than comparing our proposed systems with existing systems directly. We focus...
on kernel machines, including the SVM, that has been extensively studied in the machine learning community.

In the next section, we briefly describe the three kernel machines for classification considered in this study: SVMs, Relevance Vector Machines (RVM), and our newly proposed Classification Relevance Units Machine (CRUM). The RVM and CRUM have not been previously used to solve the phosphorylation site prediction problem. A simplifying modification of the CRUM is also described. Details on the construction of the dataset, model selection, and performance metrics are described in Section 3. Then in Section 4, the experimental results and analysis of the SVM, RVM, and CRUM comparisons are presented. Finally, we conclude our work with discussions on the advantages and shortcomings of the proposed CRUM predictors, and future prospects of overcoming the limitations.

2. KERNEL MACHINES

We assume that the data is a set of N independent input vectors \( X = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^d \), along with their corresponding targets \( t = \{t_1, t_2, \ldots, t_N\} \). In this preliminary study, we only consider the binary classification such that the target values are binary, \( t_i \in \{0, 1\} \) for the CRUM and RVM, and \( t_i \in \{-1, 1\} \) for the SVM. The value of \( t_i \) is set to 1 if site \( x_i \) is phosphorylatable, otherwise \( t_i \) is set to 0 or -1 for CRUM/RVM and SVM, respectively.

2.1 Support Vector Machine

The goal of an SVM is to find the hyperplane decision boundary that perfectly separates the two classes in the training dataset with maximum margin [27]. This means the distance from the closest training point to the hyperplane should be maximized. The SVM has been generalized to allow for overlapping class distributions [8] and non-linear decision boundaries via use of kernel functions [5]. A commonly used kernel function, and the one used in this study, is the Gaussian radial basis function (RBF) kernel with parameter \( \gamma: k(x, x') = \exp(-\gamma ||x - x'||^2) \). To use the SVM, a user must specify values for the parameter \( C \) that controls the trade-off between minimizing training error and model complexity, and all the kernel parameters; namely, \( \gamma \) in the case of the Gaussian RBF kernel.

Upon completion of training, we can classify a new data point \( x \) using the following formula:

\[
\hat{t}(x) = \text{sgn} \left( \sum_{i=1}^{N} w_i k(x, x_i) + b \right)
\]

where the \( w_i \)'s and \( b \) are learned through the SVM training [5] and \( \text{sgn}(x) \) is the sign function that evaluates to -1 if \( x < 0 \) and 1 if \( x > 0 \). As a consequence of the SVM training, many of the \( w_i \)'s will vanish. Therefore only a subset of the training data is required to make predictions. The training data \( x_i \) whose associated \( w_i > 0 \) are called support vectors (SVs).

2.2 Relevance Vector Machine

Unlike the SVM, the RVM seeks to model the posterior distribution \( p(C_i|\theta) \) that a site \( x \) is phosphorylatable (i.e. is a member of the positive class \( C_+ \)) using the following model,

\[
p(C_i|\mathbf{x}) = \sigma \left( \sum_{i=1}^{N} \alpha_i k(x, x_i) + b \right)
\]

where \( \sigma(t) = (1 + e^{-t})^{-1} \), and the \( \alpha_i \)'s and \( b \) are learned through the RVM training. Empirically it has been seen that more \( w_i \) terms vanish with the RVM than with the SVM, resulting in a sparser, more parsimonious model [25]. Similar to the idea of SVs, the vectors \( x_i \) whose associated \( w_i > 0 \) are called relevance vectors (RVs).

3. METHODS

3.1 Implementation

To conduct the computational experiments, we use the LIBSVM [7] implementation of the SVM that is freely available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/. For the RVM, Tipping’s MATLAB implementation of the fast RVM learning algorithm described in [26] is used, and is freely available at www.vectoranomaly.com. Since the RVM implementation is in MATLAB and a MATLAB interface is available for LIBSVM, we implemented CRUM, and conducted all dataset processing and experiments in the MATLAB environment.
3.2 Dataset Construction

The PhosphoELM [10] database of known, experimentally verified eukaryotic phosphorylatable serine (S), threonine (T), and tyrosine (Y) sites is used as the dataset. The residues surrounding the phosphorylation site of window size $W$ were extracted. For example if $W = 9$, we extract sequences of length 19 where the center residue is the phosphorylation site S, T, or Y. Window sizes suggested by NetPhos were used: a window of 9 for T and Y sites and 11 for S sites [2]. This resulted in three positive datasets, one for each type of site.

Since independent observations are assumed by the models under consideration, each positive dataset is reduced based on homology. Following PredPhospho [17], individually for each dataset, if any two sites have more than 70% identity according to a Needleman-Wunsch alignment [22], one of the two sites is removed from the dataset.

For the negative datasets, we use any sites centered on S, T, or Y that are not annotated as a phosphorylation site. This makes the assumption that such sites are indeed negatives, which is not known for certain since they may be undiscovered phosphorylation sites. Due to the rarity of phosphorylation sites, only a few false negative sites are expected in the training dataset and will affect our training and evaluation to a small degree [1]. The negative datasets use the same window size as their associated positive dataset. Also in the same manner, we reduced each negative dataset through homology. Since the negative dataset far exceeds the positive dataset in size, only a random selection of the negative sites are chosen, roughly equal in size to the positive dataset.

For an unbiased comparison between models, we need a test dataset. We therefore reserve 10% of each dataset as the test dataset. The remaining 90% forms the training and validation datasets.

The selection of the site features plays a key role in the success of a predictor. Further experimentation will be done, but for the present study sequence features are used. The BLOSUM encoding proposed in NetPhosYeast [16] is used where a residue is represented by a 20-dimensional vector corresponding to that residue’s row or column of the BLOSUM62 matrix. The values of the BLOSUM62 matrix were scaled to fall between 0 and 1, using min-max normalization. A value $v$ is normalized to $v' = (v - \text{min})/(\text{max} - \text{min})$, where in this case $\text{min} = -4$ and $\text{max} = 11$. Since the sequences of our T and Y datasets are 18 residues long (excluding the center residue that is constant), the BLOSUM encoding results in a 360-dimensional numerical vector for each site. Similarly, our S sequences are encoded as 440-dimensional vectors.

3.3 Parameter Search

All models considered contain parameters that are not learned and thus must be determined by external means. These parameters are $C$ and $\gamma$ for the SVM, $\gamma$ for the RVM, and $M$ for the CRUM. A search over the parameter space is conducted and each model is evaluated using cross-validation. The parameters that result in the lowest prediction error according to cross-validation are used to train the final predictor for the comparison. For the T and Y datasets, 10-fold cross-validation is used. Since the S dataset is significantly larger, 3-fold cross-validation is used. The parameter searches are conducted as follows. For the SVMs, we conduct a course grid search over $C = 2^i$ and $\gamma = 2^j$ where $i,j \in \{-10, -9, \ldots, 9, 10\}$. Based on the cross-validation results, a region is heuristically determined for a more precise parameter search. The parameter searches for RVM and CRUM are conducted in a similar fashion over a single dimension, $\gamma$ or $M$, respectively.

The use of clustering in CRUM allows for additional methods for model selection. Giving the k-means clustering estimates a probabilistic interpretation, the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) can be computed [9]. This allows CRUM’s $M$ to be selected as the model with the best AIC or BIC value. This has clear computational advantages: 1) the more costly supervised learning algorithm is not invoked and 2) only a single clustering needs to be conducted per $M$ value evaluated, compared to a 10-fold cross-validation that would require clustering and supervised training on 10 different datasets.

3.4 Predictor Assessment

With the parameters determined, the final SVMs, RVMs, and CRUMs are trained on the entire training dataset. The resulting predictors are then assessed based on predictive performance measures on the test datasets: sensitivity ($S_h$), specificity ($Sp$), Matthews correlation coefficient (MCC), and accuracy ($Ac$). As described in Section 3.2, the test dataset is independent of the training dataset and so had no bearing on the training or on the determination of the parameters $C$, $\gamma$, or $M$. Thus the test dataset gives unbiased performance measurements on the generalization ability of the considered methods. A posterior probability threshold of 0.5 is used on the RVM and CRUM predictors. In other words, we classify a site as phosphorylatable if the RVM or CRUM predictor reports a probability higher than 0.5.

The predictors are also assessed based on their computational complexity. The number of critical vectors (CVs), which are the SVs, RVs, and RU vectors for the SVM, RVM, and CRUM, respectively, are reported, as this measure gives an indication of the run-time and memory complexity of the resulting predictors, as given in Equations (1), (2), and (3). We also conduct run-time benchmarking trials for the training and prediction algorithms, and report the average of 10 trials. In this case, note that there is a bias toward the SVM implemented in faster compiled C++, whereas both CRUM and RVM are implemented in the slower interpreted language of MATLAB. Due to the presence of multiple local minima in the clustering phase, the CRUM predictors includes 10 replicates of k-means to mitigate this problem and is included in the training run-times reported.

The computer used on the smaller T and Y datasets is a 2.53 GHz Intel Core 2 Duo with 4 GB of memory. Due to the memory requirements for the RVM, we used a PC with 2.83 GHz Intel Core 2 Duo with 8 GB of memory to analyze the large S dataset. The run-time benchmarking trials are ran on these computers.

4. RESULTS

Table 1, Table 2, and Table 3 show the details of the comparative analysis of the predictors on the test dataset for S, T, and Y site predictions, respectively. The CRUM predictor names indicate the method used to determine the model complexity parameter $M$, which is either “CV” for cross-validation, AIC, or BIC.

In Table 1, it is shown that the SVM has the best pre-
and the quickest training times over all datasets. However, accuracy of the methods considered for S and T prediction about 3 times slower than SVM’s. It is observed that training is about 2 times faster than RVM’s and only 9 times faster even with the implementation disadvantage. We again see the CRUM and RVM predictors perform closely. The CRUM predictor names indicate the method used to determine the model complexity parameter \( M \): CV for cross-validation, AIC for Akaike Information Criterion, or BIC for Bayesian Information Criterion.

Table 1: Comparison of predictors on test dataset for serine site prediction

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Ac (%)</th>
<th>Mcc</th>
<th>Sn (%)</th>
<th>Sp (%)</th>
<th>#CV</th>
<th>Train (s)</th>
<th>Test (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>75.32</td>
<td>0.51</td>
<td>68.42</td>
<td>81.54</td>
<td>10,536</td>
<td>491.42</td>
<td>32.46</td>
</tr>
<tr>
<td>RVM</td>
<td>74.01</td>
<td>0.48</td>
<td>69.52</td>
<td>78.08</td>
<td>76</td>
<td>3312.55</td>
<td>2.03</td>
</tr>
<tr>
<td>CRUM-CV</td>
<td>73.97</td>
<td>0.48</td>
<td>69.63</td>
<td>77.89</td>
<td>134</td>
<td>2964.50</td>
<td>2.27</td>
</tr>
<tr>
<td>CRUM-BIC</td>
<td>70.34</td>
<td>0.40</td>
<td>63.82</td>
<td>76.21</td>
<td>64</td>
<td>2681.57</td>
<td>1.10</td>
</tr>
<tr>
<td>CRUM-AIC</td>
<td>74.13</td>
<td>0.48</td>
<td>68.42</td>
<td>79.27</td>
<td>157</td>
<td>3135.63</td>
<td>2.74</td>
</tr>
</tbody>
</table>

Table 2: Comparison of predictors on test dataset for threonine site prediction

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Ac (%)</th>
<th>Mcc</th>
<th>Sn (%)</th>
<th>Sp (%)</th>
<th>#CV</th>
<th>Train (s)</th>
<th>Test (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>76.85</td>
<td>0.53</td>
<td>60.45</td>
<td>89.25</td>
<td>3031</td>
<td>25.80</td>
<td>2.16</td>
</tr>
<tr>
<td>RVM</td>
<td>73.75</td>
<td>0.46</td>
<td>60.90</td>
<td>83.87</td>
<td>50</td>
<td>499.68</td>
<td>0.19</td>
</tr>
<tr>
<td>CRUM-CV</td>
<td>74.35</td>
<td>0.48</td>
<td>59.09</td>
<td>86.38</td>
<td>82</td>
<td>129.66</td>
<td>0.25</td>
</tr>
<tr>
<td>CRUM-BIC</td>
<td>69.74</td>
<td>0.38</td>
<td>53.64</td>
<td>82.44</td>
<td>20</td>
<td>73.80</td>
<td>0.07</td>
</tr>
<tr>
<td>CRUM-AIC</td>
<td>73.74</td>
<td>0.47</td>
<td>57.73</td>
<td>86.38</td>
<td>88</td>
<td>139.56</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 3: Comparison of predictors on test dataset for tyrosine site prediction

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Ac (%)</th>
<th>Mcc</th>
<th>Sn (%)</th>
<th>Sp (%)</th>
<th>#CV</th>
<th>Train (s)</th>
<th>Test (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>68.52</td>
<td>0.37</td>
<td>64.52</td>
<td>72.19</td>
<td>2339</td>
<td>11.10</td>
<td>1.24</td>
</tr>
<tr>
<td>RVM</td>
<td>69.75</td>
<td>0.39</td>
<td>65.16</td>
<td>73.96</td>
<td>16</td>
<td>75.95</td>
<td>0.04</td>
</tr>
<tr>
<td>CRUM-CV</td>
<td>70.06</td>
<td>0.40</td>
<td>65.81</td>
<td>73.96</td>
<td>37</td>
<td>53.07</td>
<td>0.07</td>
</tr>
<tr>
<td>CRUM-BIC</td>
<td>70.37</td>
<td>0.41</td>
<td>66.45</td>
<td>73.96</td>
<td>16</td>
<td>36.91</td>
<td>0.03</td>
</tr>
<tr>
<td>CRUM-AIC</td>
<td>70.37</td>
<td>0.41</td>
<td>66.45</td>
<td>73.96</td>
<td>78</td>
<td>78.31</td>
<td>0.17</td>
</tr>
</tbody>
</table>

This comes at the significant cost that the user does not know the uncertainty of the SVM’s prediction, or has an inaccurate measure of uncertainty if Platt’s method is used to obtain posterior estimates from SVM outputs [23, 25]. Both the CRUM and RVM provide accurate posterior estimates by virtue of their data likelihood-based objective functions [21, 25], while also significantly reducing the runtime and memory complexity of the prediction algorithm, as indicated by the use of less than 5% of the CVs than the SVM. Arguably, prediction run-time that is more important than training run-time, as training is typically invoked at far less frequency than the prediction algorithm.

In comparison to the RVM, the CRUM achieves comparable predictive performance with a lower cost training algorithm, in both run-time and memory complexity. Furthermore, the use of AIC and BIC, model selection is simpler and faster with the CRUM. It is known that with a finite sample, BIC underestimates the model complexity, while the AIC overestimates it [13], as confirmed here by the cross-validation model complexity falling between BIC’s and AIC’s. However the results also show that this is not an issue with the CRUM, as the built-in regularization mitigates the overfitting problem and the resulting complex AIC model exhibits comparable results to the cross-validation model. Therefore the use of AIC is a cheap, viable option for CRUM model selection.

Overall, we see that the SVM offers the best prediction accuracy of the methods considered for S and T prediction and the quickest training times over all datasets. However, this comes at the significant cost that the user does not know the uncertainty of the SVM’s prediction, or has an inaccurate measure of uncertainty if Platt’s method is used to obtain posterior estimates from SVM outputs [23, 25]. Both the CRUM and RVM provide accurate posterior estimates by virtue of their data likelihood-based objective functions [21, 25], while also significantly reducing the runtime and memory complexity of the prediction algorithm, as indicated by the use of less than 5% of the CVs than the SVM. Arguably, prediction run-time that is more important than training run-time, as training is typically invoked at far less frequency than the prediction algorithm.

In comparison to the RVM, the CRUM achieves comparable predictive performance with a lower cost training algorithm, in both run-time and memory complexity. Furthermore, the use of AIC and BIC, model selection is simpler and faster with the CRUM. It is known that with a finite sample, BIC underestimates the model complexity, while the AIC overestimates it [13], as confirmed here by the cross-validation model complexity falling between BIC’s and AIC’s. However the results also show that this is not an issue with the CRUM, as the built-in regularization mitigates the overfitting problem and the resulting complex AIC model exhibits comparable results to the cross-validation model. Therefore the use of AIC is a cheap, viable option for CRUM model selection.
5. CONCLUSIONS

In this study, probabilistic predictions of protein phosphorylation sites were proposed using our CRUM model as an alternative to the popular SVM and RVM. The experimental results show that:

- CRUM-based predictors provide comparable predictive performance to that of an SVM- or RVM-based predictor, and has faster training times than the RVM.
- The extremely compact models of the CRUM or RVM provide a significant reduction to the run-time and memory complexity of the prediction algorithm, which is advantageous in conducting predictions at a large scale.
- Due to the CRUM’s use of regularization, selection of the model complexity $M$ is not critical and an overestimate, such as using AIC, leads to comparable performance to other methods.
- CRUM’s simple model selection scheme using clustering and AIC is faster than the cross-validation scheme required for the SVM and RVM, and does not sacrifice predictive performance. This allows further experimentation with possible features and encodings to be conducted at a faster pace.

Therefore the CRUM is the most viable candidate for kernel-based probabilistic prediction of protein phosphorylation sites.

In the future we plan to incorporate additional site features, such as structural information, to improve predictive performance. We also plan on extending the CRUM model and training algorithm for multiple outputs. This will allow for kinase-specific phosphorylation site prediction, where a prediction for which kind of kinase phosphorylates the site is also made.

6. ACKNOWLEDGMENTS

MM, GP and KB are supported by NIH Grant P20RR016467 from the National Center for Research Resources. GP and KB are supported by NIH grant P20RR018727 from the National Center for Research Resources. The paper’s contents are solely the responsibility of the authors and do not necessarily represent the official views of the NIH.

7. REFERENCES


