Data and text mining

Neural network prediction of peptide separation in strong anion exchange chromatography

Cheolhwan Oh, Stanislaw H. Žak¹, Hamid Mirzaei², Charles Buck, Fred E. Regnier² and Xiang Zhang*

Bindley Bioscience Center, ¹School of Electrical and Computer Engineering and ²Department of Chemistry, Purdue University, West Lafayette, IN 47907, USA

Received on March 27, 2006; revised on October 4, 2006; accepted on November 4, 2006
Advance Access publication November 8, 2006
Associate Editor: Satoru Miyano

ABSTRACT

Motivation: The still emerging combination of technologies that enable description and characterization of all expressed proteins in a biological system is known as proteomics. Although many separation and analysis technologies have been employed in proteomics, it remains a challenge to predict peptide behavior during separation processes. New informatics tools are needed to model the experimental analysis method that will allow scientists to predict peptide separation and assist with required data mining steps, such as protein identification.

Results: We developed a software package to predict the separation of peptides in strong anion exchange (SAX) chromatography using artificial neural network based pattern classification techniques. A multi-layer perceptron is used as a pattern classifier and it is designed with feature vectors extracted from the peptides so that the classification error is minimized. A genetic algorithm is employed to train the neural network. The developed system was tested using 14 protein digests, and the sensitivity analysis was carried out to investigate the significance of each feature.

Contact: zhang100@purdue.edu

1 INTRODUCTION

In the bottom-up approach to proteomics, proteins are generally converted to a mixture of tryptic peptides that are then fractionated chromatographically and analyzed by mass spectrometry (Hattan et al., 2005). Although liquid chromatography is widely used in proteomics, it remains a challenge to predict what kind of peptides will be present in a certain chromatographic fraction using peptide sequence information. Thus, proteomics scientists are forced to optimize experimental conditions by trial-and-error. Several mathematical models have been developed to simulate peptide separations in the reversed-phase (RP) chromatography where peptides are eluted in linear gradient mode (Petritits et al., 2003; Baczek et al., 2005). To our knowledge, no effort has thus far been focused on modeling strong anion exchange (SAX) chromatography using artificial neural networks.

In this study, the feasibility of applying machine learning methods to strong anion exchange chromatography was explored. We focused on a single experimental case: a peptide mixture loaded onto a SAX column. Some peptides are retained on the column while the rest are not retained and flow through the column. Peptides not retained on the column are collected and named flow-through peptides. The retained peptides are then washed off the SAX column in a single step and named the elution group. Therefore, the original peptide mixture is separated into two groups, retained (elution group) and not retained (flow-through group). Our objective is to develop a machine learning method to predict, based on the peptide sequence, whether a peptide will be retained on the SAX column.

An artificial neural network trained with a genetic algorithm was used to predict peptide separation behavior on a SAX column. We used a multi-layer perceptron as a pattern classifier and a genetic algorithm to train this network. In contrast to the conventional training technique, such as the error back-propagation algorithm, the genetic algorithm is not gradient-based (Chong and Žak, 2001) and has increased chance for a better solution because it provides a way (the mutation operation) to escape from local optimizers.

2 METHODS

2.1 Experimental methods

2.1.1 Proteolysis Each of 14 model proteins, glyceraldehyde-3-phosphate dehydrogenase (rabbit muscle), myoglobin (horse heart), cytochrome c (horse heart), alpha-crystallin chain A (bovine eye lens), Ig gamma-2 chain C region (human), serum albumin (human), concanavalin A (Jack bean), beta-galactosidase (Escherichia coli), alpha-amyrase (Bacillus licheniformis), ribonuclease A (bovine pancreas), beta-lactoglobulin (bovine milk), glucose oxidase (Aspergillus niger), carbonic anhydrase II (bovine erythrocyte) and ovalbumin (chicken egg), were dissolved in 50 mM HEPES buffer (pH 8.0) to the final concentration of 10 mg/ml. To denature and reduce protein samples, urea and DTT were added to a final concentration of 6 M and 10 mM, respectively. Mixtures were incubated for 1 h at 37°C, iodoacetamide was added to a final concentration of 20 mM and the reaction was allowed to proceed for an additional 30 min at 4°C. Cysteine was then added to a final concentration of 10 mM to quench extra iodoacetamide. Samples were diluted 6-fold with 50 mM HEPES (pH 8.0); and 10 mM CaCl₂. Sequence grade trypsin was added (2%), and the reaction mixture incubated at 37°C for at least 8 h. Proteolysis was quenched by adding TLCK (trypsin:TLCK ratio of 1:1 (w/w)).
Multidimensional LC-MS/MS  A total of 20 ml tryptic digest of each model protein was injected to the SAX column (Agilent) preconditioned with 50 mM HEPES buffer (pH 8.0). Unbound peptides were directed to a C18 RP column in-line with the SAX column. The SAX column was then set off-line and the RP column bound peptides were desalted using buffer A (99.5% deionized water, 0.5% acetonitrile and 0.1% formic acid) for 10 min. Peptides were eluted from RP column using a 60 min gradient from 100% buffer A to 60% buffer B (95% acetonitrile, 5% deionized water and 0.1% formic acid). Eluted peptides were analyzed using a QSTAR workstation (Applied Biosystems, Framingham, MA) equipped with an ESI source. Both MS and MS/MS analyses were conducted for unbound peptides. After the unbound peptide reversed-phase analysis was done, the SAX column was set in-line again and the bound peptides were eluted using 0.5 M NaCl in 50 mM HEPES buffer (pH 8.0). Eluted peptides were recaptured on the RP column and were desalted using buffer A for 20 min. Acidic peptides recaptured on the RP column were eluted using a 60 min gradient form 100% A to 60% B and were analyzed in exact same manner as for the unbound peptides.

2.1.3 Peptide identification  MS/MS data were searched against the protein database using MASCOT software (Perkins et al., 1999). Trypsin was used as digestion protein and the number of allowed missed-cleavages was set to one. Carboxamidomethylation of the cysteine was specified as a constant modification. No variable modification was considered. Monoisotopic mass value was used for peptide search. Peptide and fragment mass tolerance was set to 0.4. Default setting was used for all other variables.

2.2 Artificial neural networks

2.2.1 Feature extraction  We extracted six features from the peptide sequences: molecular weight (f1), sequence index (f2), length (f3), N-terminal amino acid residue (f4), C-terminal amino acid residue (f5) and charge (f6). Molecular weight (f1) is the sum of the molecular weight of each amino acid residue. We designed the sequence index (f2) to reflect the influence of the order of amino acids in the peptide sequences. The sequence index has the form

\[
SI = \sqrt{\frac{\sum_{i=1}^{N-1} (m_i + m_{i+1})^2}{N-1}}
\]

where \(m_i\) is the pI value of the \(i\)th amino acid in the peptide, and \(N\) is the number of amino acids in the peptide. Length (f3) is the number of amino acid residues in a peptide: N-pK\(_a\) value (f4), C-pK\(_a\) value (f5) and charge (f6). Molecular weight (f1) is the sum of the molecular weight of each amino acid residue. Charge contributions were calculated using the formulas:

\[
N_i = [A^-]/([A^-] + [AH]) = \alpha/(1 + \alpha)
\]

and

\[
P_l = [BH^+]/([BH^+] + [B]) = 1/(1 + \alpha),
\]

where \(\alpha = 10^{pK_a - pK}\). Values of the ionization constant (pK\(_a\)) for each amino acid residue were derived from the literature (Rickard et al., 1991).

2.2.2 Normalization  We normalized the feature values to reflect the influence of each feature as equally as possible. Every feature value of each category was normalized using the formula,

\[
\text{normalized} = 2 \times (\text{raw} - \text{min})/(\text{max} - \text{min}) - 1,
\]

where ‘raw’ and ‘normalized’ are the values of the feature before and after normalization, and ‘min’ and ‘max’ are minimum and maximum values of the corresponding feature category. Thus, all the normalized feature values reside in the interval \([-1,1]\).

2.2.3 Designing the neural network  To predict peptide behavior, we use a multi-layer perceptron composed of the input, hidden and output layers (Fig. 1). Each layer contains neurons where nonlinear transformation is performed. Each neuron in each layer is connected to every neuron in the adjacent layer(s). The training or testing vectors are presented to the input layer, and processed by the hidden and output layers. Detailed analysis of multi-layer perceptrons has been presented by Hassoun (1995) and by Zak (2003). The output of the network is

\[
z_k = g\left(\sum_{j=1}^{m} w_{kj} y_j - t_k^o\right),
\]

where

\[
y_j = g\left(\sum_{i=1}^{l} w_{ij} x_i - t_j^o\right),
\]

and \(g\) is a nonlinear activation function of the form,

\[
g(v) = \frac{1 - e^{-v}}{1 + e^{-v}}.
\]

The values of the network parameters, \(w_{ki}^o\), \(w_{ij}^o\), \(t_i^o\), and \(t_j^o\) are determined by an appropriate training method. First, we construct the cost function that properly reflects the classification errors,

\[
f = \sum_{p=1}^{N} \sum_{l=1}^{m} (z_l(x_p) - d_{lk})^2,
\]

where \(x_p\) is a feature vector of the \(p\)th training sample, \(z_l(x_p)\) is the actual output of the \(l\)th neuron of the output layer for \(x_p\), \(d_{lk}\) is the corresponding desired output, and \(N\) is the number of the training samples. Note that \(f\) is a function of \(w_{ki}^o\), \(w_{ij}^o\), \(t_i^o\), and \(t_j^o\), where \(i = 1, \ldots, l\), \(j = 1, \ldots, m\), and \(k = 1, \ldots, n\).

The neural network was trained so that the cost function \(f\) was minimized via application of a genetic algorithm. The process began with constructing the fitness function that is to be maximized. In our experiments, we used \(1/(1 + f)\) as a fitness function to make the original minimization problem suitable for application of the genetic algorithm. Then, an initial population of candidate solutions was generated. These candidate solutions are called chromosomes. In our implementation, the chromosomes are represented by binary strings. The objective is to find a chromosome that maximizes the fitness function. After the initial population was generated, the next step was to create a mating pool using a selection operator. The selection operation is
a procedure that picks out the chromosomes based on their fitness scores. We employed a roulette wheel selection method, where the probability that each chromosome was selected was proportional to its fitness score. The next population was formed by employing the crossover and the mutation operators acting on the chromosomes in the mating pool. We adopted a single point crossover method. First, two chromosomes, which are called parent chromosomes, in the mating pool were chosen randomly. Also, the crossover point in the chromosomes was chosen randomly. Then, two new chromosomes were created by exchanging the parts between the crossover points and the end points of the parent chromosomes. The mutation operation was carried out in such a way that each bit in each chromosome was flipped by a given mutation rate. In addition, we incorporated the elitism strategy into our system. With the selection, crossover, and mutation operations described above, the best chromosomes in the current generation may not be preserved into the next generation. To prevent this situation, the best chromosomes were automatically put in the next generation in the first step. Then, the remaining spots of the next generation were filled by the crossover and the mutation operations. The above procedure was repeated until the predetermined stopping criteria were met.

Although the training error can be minimized with the method described above, it may not guarantee the minimization of the classification error in the validation process. To this end, we chose the chromosome that gives the minimum testing error and converted it to the neural network parameters. An example is shown in Figure 2. The root mean square error (RMSE) was used to measure the training and testing errors.

![Figure 2. An example of learning curves.](https://academic.oup.com/bioinformatics/article-abstract/23/1/114/189931)

3 RESULTS

We have developed a neural network system trained with a genetic algorithm to predict peptide separation in the SAX. The system has been tested with pure protein digests, where 14 proteins were individually digested into tryptic peptides. Each protein digest was then analyzed on the SAX followed by the RP-MS. One hundred fifty peptides were identified.

To evaluate the performance of our system, we carried out the following procedure. First, the whole set of peptides was divided into two groups: a designing set and a testing set. We designed the neural network with the peptides in the training set and the testing set as described in section 2.2 (see Fig. 2). The performance of the neural network classifier with parameters determined in the designing procedure was evaluated with the peptides in the validation set. Out of 150 peptides, we used 90 peptides for a training set, 30 peptides for a testing set and the remaining 30 peptides for a validation set. We repeated the designing and validation procedures 100 times to obtain the average performance. In each time, different sets of randomly selected peptides were assigned to the training, the testing and the validation sets, respectively.

Analysis on the significance of network features was carried out. Several sensitivity analysis or saliency measure techniques on the features used with the neural network classifier have been proposed for this purpose (e.g. Moody, 1994; Belue and Bauer, 1995; Mak and Blaming, 1998). These methods give estimates for the significance of single features, but they do not detect the influences of the combinations of the features. Hence, we adopted an exhaustive approach. That is, we designed a neural network with each combination of features and evaluated its performance. The results with neural networks constructed with at least three features are summarized in Table 1. The RMSEs were measured in the design procedure, and the classification success rates (SR) were measured in the validation process. The best SR we obtained was 84%. As was expected, the higher SR was obtained when the RMSE was smaller.

To investigate the significance of each feature and the combination of features, we define the sensitivity of the feature(s) as follows.

\[
S(A | B) = SR(B) - SR(B - A),
\]

where A is the set of feature(s) for which we require sensitivity, B is the set of features used as inputs in the construction process of the neural network classifier, which contains A. For example, the sensitivity of the combination of features 1 and 2 in the classifier constructed with features 1, 2, 3, 4, 5, 6 is

\[
S(1, 2 | 1, 2, 3, 4, 5, 6) = SR(1, 2, 3, 4, 5, 6) - SR(3, 4, 5, 6) = 0.79 - 0.76 = 0.3.
\]

This sensitivity metric measures how much the classification performance is deteriorated when the specific feature(s) are eliminated from the input layer of the neural network classifier. As this value increases, the corresponding feature(s) can be considered more significant.

First, we can see that N-pK_a value (f4) and C-pK_a value (f5) have adverse effects on the classification. When each of them was removed from the set of the features, the classification performance was enhanced, i.e.

\[
S(4 | 1, 2, 3, 4, 5, 6) = 0.79 - 0.80 = -0.01,
S(5 | 1, 2, 3, 4, 5, 6) = 0.79 - 0.82 = -0.03,
\]

and

\[
S(4, 5 | 1, 2, 3, 4, 5, 6) = 0.79 - 0.84 = -0.5.
\]

In fact, among the 15 neural network classifiers constructed using 4 features, the classifier with the features 1, 2, 3, 6 has the best performance (see Table 1). In all the combinations that contain features 4 and 5, the sensitivity S(4, 5 | X) have negative values. Therefore, we can safely remove the features N-pK_a value.
Let us define the average sensitivity \( S_{\text{AVG}}(A) \) of feature \( A \) as the average of all possible \( S(A|X) \), where \( X \) is a set of features that contains \( A \). The average sensitivity of each feature was calculated as follows:

\[
S_{\text{AVG}}(1) = 0.0344, \quad S_{\text{AVG}}(2) = 0.0488, \\
S_{\text{AVG}}(3) = 0.0213, \quad S_{\text{AVG}}(4) = -0.0175, \\
S_{\text{AVG}}(5) = -0.0263, \quad S_{\text{AVG}}(6) = 0.0356.
\]

From these results, we conclude that the order of significance of features is sequence index \( f_2 \), charge \( f_6 \), molecular weight \( f_1 \), length \( f_3 \), N-pK_a value \( f_4 \) and C-pK_a value \( f_5 \), from the most significant feature with respect to sensitivity.

The average success rate reached 84% when features \{f_1, f_2, f_3, f_6\} were used for the construction of the neural network classifier. Also, the same success rate was achieved when \{f_1, f_2, f_3\}, \{f_1, f_2, f_6\} or \{f_2, f_3, f_6\} were used. These results actually reflect the separation mechanism of SAX chromatography, where the peptide sequence and charge determine the separation.

The histogram of the classification success rate of the individual peptides is shown in Figure 3. In each simulation, 30 peptides were randomly selected from 150 peptides as members of the validation set. The remaining 120 peptides were used for the design of the ANN. We repeated the simulation 100 times and counted the number of successful classifications of each peptide in the validation set. We performed the above procedure respectively with the different feature sets \{f_1, f_2, f_3, f_6\}, \{f_1, f_2, f_3\}, \{f_1, f_2, f_6\} and \{f_2, f_3, f_6\}, and calculated the overall classification success rate of the individual peptides. We used the majority criterion to decide whether a peptide should be classified as a member of the elution group or a member of the flow-through group. That is, we counted the number of simulations where a peptide \( i \) was classified into the flow-through group \( (N_{\text{ft}}) \) and the number of simulations where the same peptide was classified into the elution group \( (N_{\text{el}}) \). If \( N_{\text{ft}} > N_{\text{el}} \), the peptide was assigned to the flow-through group. Otherwise, it was assigned to the elution group. Applying this method to the simulations, we were able to correctly classify 127 peptides, which is 84.7% of the total number of peptides (Fig. 3).

To further analyze the prediction performance of ANN, we studied the dependency of the classification success rate on molecular weight and calculated charge of peptides. The results are displayed in Figures 4 and 5. We categorized peptides into multiple groups based on their molecular weights or calculated charges. The classification success rate was defined as number of peptides that were correctly classified divided by the total number of peptides in the same group. The majority criterion was employed again. As shown in Figures 4 and 5, there was no clear correlation between the classification success rate and peptide molecular weight or peptide charge. Although the classification success rates are 1 with the peptides in the outer bins in both Figures 4 and 5, the confidence

---

**Table 1. Classification results with selected features**

<table>
<thead>
<tr>
<th>Features</th>
<th>RMSE</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4, 5, 6</td>
<td>0.76</td>
<td>0.79</td>
</tr>
<tr>
<td>1, 2, 3, 4, 5</td>
<td>0.80</td>
<td>0.76</td>
</tr>
<tr>
<td>1, 2, 3, 4, 6</td>
<td>0.70</td>
<td>0.82</td>
</tr>
<tr>
<td>1, 2, 3, 5, 6</td>
<td>0.71</td>
<td>0.80</td>
</tr>
<tr>
<td>1, 2, 4, 5, 6</td>
<td>0.74</td>
<td>0.78</td>
</tr>
<tr>
<td>1, 3, 4, 5, 6</td>
<td>0.80</td>
<td>0.77</td>
</tr>
<tr>
<td>2, 3, 4, 5, 6</td>
<td>0.78</td>
<td>0.78</td>
</tr>
<tr>
<td>1, 2, 3, 4, 6</td>
<td>0.73</td>
<td>0.80</td>
</tr>
<tr>
<td>1, 2, 3, 5</td>
<td>0.74</td>
<td>0.80</td>
</tr>
<tr>
<td>1, 2, 3, 6</td>
<td>0.66</td>
<td>0.84</td>
</tr>
<tr>
<td>1, 2, 4, 5</td>
<td>0.79</td>
<td>0.78</td>
</tr>
<tr>
<td>1, 2, 4, 6</td>
<td>0.70</td>
<td>0.83</td>
</tr>
<tr>
<td>1, 2, 5, 6</td>
<td>0.70</td>
<td>0.82</td>
</tr>
<tr>
<td>1, 3, 4, 5</td>
<td>0.98</td>
<td>0.69</td>
</tr>
<tr>
<td>1, 3, 4, 6</td>
<td>0.76</td>
<td>0.80</td>
</tr>
<tr>
<td>1, 3, 5, 6</td>
<td>0.78</td>
<td>0.78</td>
</tr>
<tr>
<td>1, 4, 5, 6</td>
<td>0.77</td>
<td>0.79</td>
</tr>
<tr>
<td>2, 3, 4, 5</td>
<td>0.80</td>
<td>0.78</td>
</tr>
<tr>
<td>2, 3, 4, 6</td>
<td>0.74</td>
<td>0.81</td>
</tr>
<tr>
<td>2, 3, 5, 6</td>
<td>0.73</td>
<td>0.80</td>
</tr>
<tr>
<td>2, 4, 5, 6</td>
<td>0.92</td>
<td>0.70</td>
</tr>
<tr>
<td>3, 4, 5, 6</td>
<td>0.83</td>
<td>0.76</td>
</tr>
<tr>
<td>1, 2, 3</td>
<td>0.67</td>
<td>0.84</td>
</tr>
<tr>
<td>1, 2, 4</td>
<td>0.71</td>
<td>0.81</td>
</tr>
<tr>
<td>1, 2, 5</td>
<td>0.73</td>
<td>0.81</td>
</tr>
<tr>
<td>1, 2, 6</td>
<td>0.64</td>
<td>0.84</td>
</tr>
<tr>
<td>1, 3, 4</td>
<td>0.98</td>
<td>0.66</td>
</tr>
<tr>
<td>1, 3, 5</td>
<td>0.94</td>
<td>0.69</td>
</tr>
<tr>
<td>1, 3, 6</td>
<td>0.75</td>
<td>0.81</td>
</tr>
<tr>
<td>1, 4, 5</td>
<td>0.97</td>
<td>0.68</td>
</tr>
<tr>
<td>1, 4, 6</td>
<td>0.74</td>
<td>0.80</td>
</tr>
<tr>
<td>1, 5, 6</td>
<td>0.77</td>
<td>0.79</td>
</tr>
<tr>
<td>2, 3, 4</td>
<td>0.75</td>
<td>0.81</td>
</tr>
<tr>
<td>2, 3, 5</td>
<td>0.72</td>
<td>0.81</td>
</tr>
<tr>
<td>2, 3, 6</td>
<td>0.70</td>
<td>0.84</td>
</tr>
<tr>
<td>2, 4, 5</td>
<td>0.89</td>
<td>0.71</td>
</tr>
<tr>
<td>2, 4, 6</td>
<td>0.88</td>
<td>0.71</td>
</tr>
<tr>
<td>2, 5, 6</td>
<td>0.88</td>
<td>0.70</td>
</tr>
<tr>
<td>3, 4, 5</td>
<td>0.98</td>
<td>0.67</td>
</tr>
<tr>
<td>3, 4, 6</td>
<td>0.80</td>
<td>0.78</td>
</tr>
<tr>
<td>3, 5, 6</td>
<td>0.83</td>
<td>0.75</td>
</tr>
<tr>
<td>4, 5, 6</td>
<td>0.92</td>
<td>0.69</td>
</tr>
</tbody>
</table>

1: molecular weight, 2: sequence index, 3: length, 4: N-pK_a value, 5: C-pK_a value, 6: charge.
level for those results should not be as high as for the interior bins
because the number of peptides in the outer bins is relatively small.

4 CONCLUSIONS

We report a method of using a neural network based pattern
classification system trained with a genetic algorithm to predict
peptide separation in SAX chromatography. This approach provides
advantage for protein identification efforts in proteomics studies. In
this study, we obtained an average classification success rate of 84%
in predicting peptide separation on a SAX column using six features
to describe each peptide. Out of the six features, sequence index,
charge, molecular weight and sequence length make significant
contributions to the prediction. \( p_{K_a} \) values of C-terminal and N-
terminal amino acids did not contribute to the classification task.
This may indicate that peptide separation on the SAX is mainly
determined by the entire sequence. The contribution of N-terminal
and C-terminal amino acids is limited.

ACKNOWLEDGEMENTS

The authors are grateful for reviewers’ constructive comments.
This project was supported by seed funding from Bindley
Bioscience Center, Purdue University.

Conflict of Interest: none declared.

REFERENCES

Baczek,T. et al. (2005) Prediction of peptide retention at different HPLC conditions
from multiple linear regression models. J. Proteome Res., 4, 555–563.
perceptrons. Neurocomputing, 7, 111–121.
Wiley, NY.
Cambridge, MA.
Hattan,S.J. et al. (2005) Comparative study of [three] LC-MALDI workflows for the
561–564.
In Cherkassky,V., Friedman,J.H. and Wechsler,H. (eds), From Statistics to Neural
Networks, Theory and Pattern Recognition Applications. Springer-Verlag, Berlin,
NY, pp. 147–165.
sequence databases using mass spectrometry data. Electrophoresis, 20, 3551–3567.
Petritts,K. et al. (2003) Use of artificial neural networks for the accurate prediction
of peptide liquid chromatography elution times in proteome analyses. Anal. Chem.,
75, 1039–1048.
Rickard,E.C. et al. (1991) Correlation of electrofocusing mobilities from capillary
electrophoresis with physicochemical properties of proteins and peptides. Anal.
Biochem., 197, 197–207.