BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 9 2005, pages 1831–1837
doi:10.1093/bioinformatics/bti281

Sequence analysis

Prediction of caspase cleavage sites using Bayesian bio-basis function neural networks

Zheng Rong Yang

Department of Computer Science, Exeter University, Exeter, Devonshire, UK

Received on May 11, 2004; revised on December 20, 2004; accepted on January 18, 2005
Advance Access publication January 25, 2005

ABSTRACT

Motivation: Apoptosis has drawn the attention of researchers because of its importance in treating some diseases through finding a proper way to block or slow down the apoptosis process. Having understood that caspase cleavage is the key to apoptosis, we find novel methods or algorithms are essential for studying the specificity of caspase cleavage activity and this helps the effective drug design. As bio-basis function neural networks have proven to outperform some conventional neural learning algorithms, there is a motivation, in this study, to investigate the application of bio-basis function neural networks for the prediction of caspase cleavage sites.

Results: Thirteen protein sequences with experimentally determined caspase cleavage sites were downloaded from NCBI. Bayesian bio-basis function neural networks are investigated and the comparisons with single-layer perceptrons, multilayer perceptrons, the original bio-basis function neural networks and support vector machines are given. The impact of the sliding window size used to generate sub-sequences for modelling on prediction accuracy is studied. The results show that the Bayesian bio-basis function neural network with two Gaussian distributions for model parameters (weights) performed the best and the highest prediction accuracy is 97.15 ± 1.13%.

Availability: The package of Bayesian bio-basis function neural network can be obtained by request to the author.

Contact: z.r.yang@ex.ac.uk

INTRODUCTION

Apoptosis (programmed cell death) is a gene-directed mechanism activated as a suicidal event to eliminate excess, damaged, or infected cells (Rohn *et al*., 2004). The function of apoptosis is vital to life as it serves to regulate and control both cell death and tissue homeostasis during the development and the maturation of cells. It was reported that ∼100 000 cells die by apoptosis every second for the purpose of regulating tissues. A family of cysteine proteases called caspases, that are expressed initially in the cell as proenzymes, is the key to apoptosis (Rohn *et al*., 2004). As indicated in Chou *et al*. (2000), cell death is also important in optimizing the function in the immune and central nervous systems. In organisms, cell death and renewal are important functions for the control of the flux of fresh cells at a constant level. The importance of apoptosis study is that many diseases result from apoptosis malfunction. For instance, cancer can occur if apoptosis is blocked (Adams and Cory, 1998; Evan and Littlewood, 1998). Unwanted apoptosis may result in ischemic damage (Reed and Paternostro, 1999).

The activation of caspases (cysteinyl aspartate-specific proteases) is the key to apoptosis. Caspases can initiate a cascade of cleavage activities causing disruption of the components within the cell, as well as the disabling of critical repair processes. Although structural information is hard to obtain, homology alignment of the protein sequences has shown that caspases have highly conserved aspartic acid residues in the substrates (Chou *et al*., 2000).

Because apoptosis is critical to some diseases and cell growth, caspases have been widely studied. For instance, how caspase 7 cleaves tumour necrosis factor receptor-I (TNFR1), to which ligand binding can promote cell survival, or how it activates the apoptosis caspase cascade were studied in Ethell *et al*. (2001). In the study of Alzheimer's disease (AD), caspase cleavage has also been paid attention to. For instance, the identification of caspases that can cleave presenilin-1 (PS1) and presenilin-2 (PS2) has been done by Van de Craen *et al*. (1999), where it has been found that PS1 can be cleaved by two groups, the group that contains caspases 8, 6 and 11 and the group that contains caspases 1, 3 and 7, while PS2 can be proteolysed by caspases 1, 3, 6 and 8. One of the important issues in signalling pathway research is the regulation of cell survival and programmed cell death which are closely related to different signalling molecules (West *et al*., 2002). A decision between cell survival or death is made based not only on the levels of activation of these signalling molecules, but also on the subcellular targeting. As caspase cleavage can induce subcellular translocation, how subcellular targeting regulates the function of caspase-activated protein kinases in apoptosis was studied in Jakobi (2004).

As caspase cleavage is the key to programmed cell death, the study of caspase inhibitors could represent effective new drugs against some disease where blocking apoptosis is desirable (Chou *et al*., 2000). Without a careful study of caspase cleavage specificity effective drug design could be difficult.

Neural learning algorithms have been widely applied for the recognition of various functional sites in proteins. For instance, multilayer perceptrons have been applied for the prediction of HIV and Hepatitis C virus protease cleavage sites (Thompson *et al*., 1995; Cai and Chou, 1998; Narayanan *et al*., 2002), phosphorylation site prediction (Blom *et al*., 1999; Berry *et al*., 2004), the prediction of signal peptide cleavage sites (Nielsen *et al*., 1997) and the prediction of protein–protein interaction sites (Gutteridge *et al*., 2003). A self-organizing map has been applied for mining the rules from HIV protease data (Yang and Chou, 2003). These neural learning algorithms have to use an encoding method to preprocess amino acids which are represented using non-numerical letters. Having understood that the distributed encoding method leads to inefficiency in modelling protein sequences, bio-basis function neural networks were developed for the recognition of functional sites in proteins with success (Thomson *et al*., 2003; Yang and Chou, 2004; Berry *et al*., 2004).

This study, therefore, investigates the use of bio-basis function neural networks in the prediction of caspase cleavage sites. The Bayesian method, in particular, is used to enhance bio-basis function neural networks for dealing with the priors of the parameter structure in bio-basis function neural networks. Single layer perceptrons, multilayer perceptrons, the original bio-basis function neural networks and support vector machines are used for comparison.

The simulation is carried out on 13 protein sequences containing various experimentally determined caspase cleavage sites. These 13 protein sequences are downloaded from NCBI (http://www.ncbi.nih.gov). Jackknife simulation is used to assess the model performance. The impact of the sliding window size on model performance is also studied.

SYSTEM AND METHODS

The parameter priors were not used in the bio-basis function neural network proposed in 2003 (Thomson *et al*., 2003). A recent empirical finding that is opposed to our belief in the parameter structure (Yang and Chou, 2004) motivated us to investigate the use of the parameter prior in the Bayesian framework in this study.

Bio-basis function neural networks

The bio-basis function neural network is composed of three layers, i.e. input, bio-basis and output layers. The input layer is composed of D neurons corresponding to D residues in a capsase sub-sequence. Each caspase sub-sequence is obtained by scanning a protein sequence with a fix-sized sliding window. The residues scanned by the sliding window at a specific position in the sequence are denoted as a caspase sub-sequence. A caspase sub-sequence is referred to as a positive one if there is a cleavage site in the middle of it, otherwise it is considered negative. Each bio-basis is supported by a caspase sub-sequence with known property, i.e. with or without a cleavage site. The sub-sequence used by a bio-basis is referred to as a support sub-sequence (SS). The output neuron is used for decision-making. Each input neuron is used to accept amino acids from one specified residue in sub-sequences. The input amino acid is delivered to the relevant residue of each SS. Pairwise homology alignment is used to calculate the similarity between an input caspase sub-sequence and an SS. The similarity is then normalized using a specifically designed bio-basis function (Thomson *et al*., 2003). All the similarities are then weighted to produce an output for decision-making.

Suppose a caspase sub-sequence is referred to as $\mathbf{s}_m$ and its target $y_m \in \{0,1\}$, where '1' means that $\mathbf{s}_m$ has a cleavage site and '0', not a mathematical description of a bio-basis function neural network classifier, for predicting if there is a caspase cleavage site in $\mathbf{s}_m$, is defined as (Thomson *et al*., 2003)

$$\hat{y}_m = \sum_{k=1}^{K} w_k f(\mathbf{s}_m, \mathbf{z}_k) = y_m - e_m, \tag{1}$$

where $\hat{y}_m$ is the prediction for $\mathbf{s}_m$, w_k the weight connecting the k-th bio-basis function (supported by $\mathbf{z}_k$) to the output unit, e_m the error, $f(\mathbf{s}_m, \mathbf{z}_k)$ the bio-basis function, which quantifies the normalized similarity between $\mathbf{s}_m$ and $\mathbf{z}_k$, is defined in Thomson *et al*. (2003). A feature matrix F has M rows for M outputs and K columns for K bio-bases, where the entry in the m-th row and k-th column means the output from the k-th bio-basis for the m-th input. We denote the target vector as $\mathbf{y} = (y_1, y_2, \ldots, y_M)^t$, the error vector as $\mathbf{e} = (e_1, e_2, \ldots, e_M)^t$ and the parameter vector as $\mathbf{w} = (w_1, w_2, \ldots, w_K)^t$ and a vector–matrix notation of a linear classifier in the feature space formed by $\mathbf{F}$ is defined as

$$\mathbf{y} = \mathbf{F}\mathbf{w} + \mathbf{e}. \tag{2}$$

Bayesian method for bio-basis function neural networks

In the system defined in Equation (2), both the errors and the weights are assumed to follow certain probability density functions. These functions are generally not known and regarded as priors. As the errors are generally assumed to follow a Gaussian, $e_m \sim N(0, \sigma_e = 1/\sqrt{\rho_e})$ ($\rho_e \sim N(0,1)$ is the hyperparameter), this paper investigates three priors for the weights. They are of uniform distribution, single Gaussian and two Gaussians. The prior of a uniform distribution has been widely used in linear systems. The prior of a single Gaussian has been used in Bayesian neural networks (Nabney, 2003). The prior of two Gaussians is motivated by the empirical finding in the earlier study (Yang and Chou, 2004), where the weights are distributed in two distinct probability density functions for a discriminant task. When dealing with multiple classification problems, the prior of two Gaussians can be easily extended to the prior of multiple Gaussians.

We use ϑ to refer to the hyperparameters governing the error structure prior and weight structure prior. The Bayes formula of the posterior probability is shown as follows

$$p(\mathbf{w}, \vartheta|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w}, \vartheta)p(\mathbf{w}, \vartheta)}{p(\mathbf{y})}. \tag{3}$$

Note that $p(\mathbf{w}, \vartheta|\mathbf{y})$ is the posterior, $p(\mathbf{y}|\mathbf{w}, \vartheta)$ the conditional probability, $p(\mathbf{y})$ the normalization factor and $p(\mathbf{w}, \vartheta) = p(\mathbf{w}|\vartheta)p(\vartheta)$, where $p(\mathbf{w}|\vartheta)$ is the probability of the weights given the hyperparameters and $p(\vartheta)$ the a priori probability of the hyperparameters. The posterior probability is then

$$\begin{aligned} L = p(\mathbf{w}, \vartheta|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w}, \vartheta)p(\mathbf{w}, \vartheta)}{p(\mathbf{y})} &\propto p(\mathbf{y}|\mathbf{w}, \vartheta)p(\mathbf{w}, \vartheta) \\ &= p(\mathbf{y}|\mathbf{w}, \vartheta)p(\mathbf{w}|\vartheta)p(\vartheta). \end{aligned} \tag{4}$$

The method called MAP (maximum a posteriori) can be used for the parameter estimation.

Uniform distribution of parameters

If the weights are assumed to follow a uniform distribution and ρ_e is a constant, applying negative log on L leads to

$$\tilde{L} = -\ln L = C_1||\mathbf{e}||^2 + C_2, \tag{5}$$

where C_1 and C_2 are the two constants and $\mathbf{e} = \mathbf{y} - \mathbf{F}\mathbf{w}$. Maximizing L is equivalent to the least squares function and the pseudoinverse (Duda *et al*., 2002) can be used to estimate the weights

$$\mathbf{w} = (\mathbf{F}^t\mathbf{F})^{-1}\mathbf{F}^t\mathbf{y}, \tag{6}$$

where $\mathbf{F}^t$ means the transpose of $\mathbf{F}$ and $(\mathbf{F}^t\mathbf{F})^{-1}$ the inverse matrix of $\mathbf{F}^t\mathbf{F}$. We refer to this system as BBF0 which has been used in Thomson *et al*. (2003).

Single Gaussian of parameters

If the weights are assumed to follow a single Gaussian, $w_k \sim N(u_w, \sigma_w = 1/\sqrt{\rho_w})$, where the hyperparameters that control the weight distribution are assumed to follow Gaussians, i.e. $u_w \sim N(0,1)$ and $\rho_w \sim N(0,1)$. Applying negative log of L leads to

$$\begin{aligned} \tilde{L} = -\ln L = &\frac{1}{2}[\rho_e||\mathbf{e}||^2 + \rho_w||\mathbf{w} - \mathbf{u}_w||^2 - M\ln\rho_e \\ &- K\ln\rho_w + \mathbf{u}_w^t\mathbf{u}_w + \rho_e^2 + \rho_w^2 + C], \end{aligned} \tag{7}$$

where C is a constant and $\mathbf{u}_w = u_w\mathbf{i}_K$. Note that $\mathbf{i}_r = \underbrace{(1, 1, \ldots, 1)}_{r}{}^t$ is an r-order identity vector. Letting the partial derivative of $\tilde{L}$ with respect to ρ_e be zero leads to

$$\rho_e = \frac{-||\mathbf{e}||^2 + \sqrt{||\mathbf{e}||^4 + 8M}}{4}. \tag{8}$$

Note that there should be two solutions to the quadratic equation $2\rho_e^2 + ||\mathbf{e}||^2\rho_e - M = 0$. Because $\rho_e > 0$ and $||\mathbf{e}||^2 > 0$, we only consider the positive solution. Letting the partial derivative of $\tilde{L}$ with respect to ρ_w be zero leads to

$$\rho_w = \frac{-||\mathbf{w} - \mathbf{u}_w||^2 + \sqrt{||\mathbf{w} - \mathbf{u}_w||^4 + 8K}}{4}. \tag{9}$$

Note that only the positive solution is used again. Letting the partial derivative of $\tilde{L}$ with respect to u_w be zero leads to

$$u_w = \frac{\rho_w}{1 + K\rho_w}\mathbf{w}^{\mathrm{t}}\mathbf{i}_K. \tag{10}$$

Letting the partial derivative of $\tilde{L}$ with respect to $\mathbf{w}$ be zero leads to

$$\mathbf{\Phi}\mathbf{w} = \boldsymbol{v}, \tag{11}$$

where $\mathbf{\Phi} = \rho_e\mathbf{F}^{\mathrm{t}}\mathbf{F} + \rho_w\mathbf{I}$ and $\boldsymbol{v} = \rho_e\mathbf{F}^{\mathrm{t}}\mathbf{y} + \rho_w\mathbf{I}\mathbf{u}_w$. The estimations of the weights is $\mathbf{w} = \mathbf{\Phi}^{-1}\boldsymbol{v}$ as $\mathbf{\Phi}$ is a squared matrix. We refer to this system as BBF1.

Multiple Gaussians of parameters

We then consider the situation that the weights follow two Gaussians. We use α and β to refer to non-cleaved and cleaved caspase sub-sequences, respectively. The weights connecting the bio-bases supported by non-cleaved sub-sequences follow one Gaussian $w_{\alpha k} \sim N(u_\alpha, \sigma_\alpha = 1/\sqrt{\rho_\alpha})$ and the weights connecting the bio-bases supported by cleaved sub-sequences, the other Gaussian $w_{\beta k} \sim N(u_\beta, \sigma_\beta = 1/\sqrt{\rho_\beta})$. The hyperparameters ρ_e, u_α, ρ_α, u_β and ρ_β are also assumed to follow Gaussians; $\rho_e \sim N(0,1)$, $u_\alpha \sim N(0,1)$, $\rho_\alpha \sim N(0,1)$, $u_\beta \sim N(0,1)$ and $\rho_\beta \sim N(0,1)$. As each bio-basis has an associated class label, the feature matrix can be expressed as $\mathbf{F} = \mathbf{F}_\alpha \bigcup \mathbf{F}_\beta$, where

$$\begin{aligned}
\mathbf{F}_\alpha &= \begin{pmatrix} f(\mathbf{s}_1, \mathbf{z}_1) & f(\mathbf{s}_1, \mathbf{z}_2) & \cdots & f(\mathbf{s}_1, \mathbf{z}_{K_\alpha}) \\ f(\mathbf{s}_2, \mathbf{Z}_1) & f(\mathbf{s}_2, \mathbf{z}_2) & \cdots & f(\mathbf{s}_2, \mathbf{z}_{K_\alpha}) \\ \vdots & \vdots & \vdots & \vdots \\ f(\mathbf{s}_M, \mathbf{z}_1) & f(\mathbf{s}_M, \mathbf{z}_2) & \cdots & f(\mathbf{s}_M, \mathbf{z}_{K_\alpha}) \end{pmatrix}, \\
\mathbf{F}_\beta &= \begin{pmatrix} f(\mathbf{s}_1, \mathbf{z}_1) & f(\mathbf{s}_1, \mathbf{z}_2) & \cdots & f(\mathbf{s}_1, \mathbf{z}_{K_\beta}) \\ f(\mathbf{s}_2, \mathbf{Z}_1) & f(\mathbf{s}_2, \mathbf{z}_2) & \cdots & f(\mathbf{s}_2, \mathbf{z}_{K_\beta}) \\ \vdots & \vdots & \vdots & \vdots \\ f(\mathbf{s}_M, \mathbf{z}_1) & f(\mathbf{s}_M, \mathbf{z}_2) & \cdots & f(\mathbf{s}_M, \mathbf{z}_{K_\beta}) \end{pmatrix},
\end{aligned} \tag{12}$$

where K_α and K_β are the numbers of non-cleaved and cleaved sub-sequences, respectively. Correspondingly, we have $\mathbf{w} = \mathbf{w}_\alpha \bigcup \mathbf{w}_\beta$, $\mathbf{y} = \mathbf{y}_\alpha \bigcup \mathbf{y}_\beta$ and $\mathbf{e} = \mathbf{e}_\alpha \bigcup \mathbf{e}_\beta$. Applying negative log on L leads to

$$\begin{aligned}
\tilde{L} = -\ln L = \frac{1}{2}[&\rho_e||\mathbf{e}||^2 + \rho_\alpha||\mathbf{w}_\alpha - \mathbf{u}_\alpha||^2 + \rho_\beta||\mathbf{w}_\beta - \mathbf{u}_\beta||^2 \\
&-M\ln\rho_e - K_\alpha\ln\rho_\alpha - K_\beta\ln\rho_\beta + \mathbf{u}_\alpha^{\mathrm{t}}\mathbf{u}_\alpha \\
&+\mathbf{u}_\beta^{\mathrm{t}}\mathbf{u}_\beta + \rho_e^2 + \rho_\alpha^2 + \rho_\beta^2 + C],
\end{aligned} \tag{13}$$

where C is a constant, $\mathbf{u}_\alpha = \mu_\alpha\mathbf{i}_{K_\alpha}$ and $\mathbf{u}_\beta = \mu_\beta\mathbf{i}_{K_\beta}$. Letting the partial derivative of $\tilde{L}$ with respect to ρ_e be zero leads to the same result as above Equation (8). We use ξ to represent α and β. Letting the partial derivative of $\tilde{L}$ with respect to ρ_α or ρ_β be zero leads to

$$\rho_\xi = \frac{-||\mathbf{w}_\xi - \mathbf{u}_\xi||^2 + \sqrt{||\mathbf{w}_\xi - \mathbf{u}_\xi||^4 + 8K_\xi}}{4}. \tag{14}$$

Letting the partial derivative of $\tilde{L}$ with respect to u_α or u_β be zero leads to

$$u_\xi = \frac{\rho_\xi}{1 + K_\xi\rho_\xi}\mathbf{w}_\xi^{\mathrm{t}}\mathbf{i}_{K_\xi}. \tag{15}$$

Letting the partial derivative of $\tilde{L}$ with respect to $\mathbf{w}_\alpha$ or $\mathbf{w}_\beta$ be zero leads to

$$\rho_e\mathbf{F}_\xi^{\mathrm{t}}\mathbf{F}_\xi\mathbf{w} + \rho_\xi\mathbf{w}_\xi = \rho_e\mathbf{F}_\xi^{\mathrm{t}}\mathbf{y}_\xi + \rho_\xi\mathbf{u}_\xi \tag{16}$$

or

$$(\rho_e\mathbf{F}^{\mathrm{t}}\mathbf{F} + \rho_\zeta\mathbf{I})\mathbf{w} = \rho_e\mathbf{F}^{\mathrm{t}}\mathbf{y} + \rho_\zeta\mathbf{I}\mathbf{u}, \tag{17}$$

where $\rho_\zeta\mathbf{I}$ is defined as follows, the first α diagonal elements are assigned value of ρ_α and the last β diagonal elements are assigned value of ρ_β

$$\rho_\zeta\mathbf{I} = \begin{pmatrix} \rho_\alpha & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \rho_\alpha & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \rho_\alpha & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \rho_\beta & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \rho_\beta & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \rho_\beta \end{pmatrix}. \tag{18}$$

Suppose $\mathbf{\Phi} = \rho_e\mathbf{F}^{\mathrm{t}}\mathbf{F} + \rho_\zeta\mathbf{I}$ and $\boldsymbol{v} = \rho_e\mathbf{F}^{\mathrm{t}}\mathbf{y} + \rho_\zeta\mathbf{I}\mathbf{u}$, Equation (17) can be rewritten as

$$\mathbf{\Phi}\mathbf{w} = \boldsymbol{v}. \tag{19}$$

The estimation of the weights is $\mathbf{w} = \mathbf{\Phi}^{-1}\boldsymbol{v}$. We refer to this system as BBF2.

Expectation–maximization algorithm for the estimation of the parameters in BBF1 and BBF2

As the partial derivatives of $\tilde{L}$ with respect to some parameters are not in the closed forms, the learning of the parameters in BBF1 and BBF2 including hyperparameters can be implemented using the principle of the expectation–maximization (EM) algorithm. Each parameter is assigned a random value at the beginning. In the t-th learning cycle of the E-step, hyperparameters are estimated as follows

$$\begin{aligned}
\rho_e(t+1) &= \frac{-||\mathbf{e}(t)||^2 + \sqrt{||\mathbf{e}(t)||^4 + 8M}}{4}. \\
\rho_\hbar(t+1) &= \frac{-||\mathbf{w}_\hbar(t) - \mathbf{u}_\hbar(t)||^2 + \sqrt{||\mathbf{w}_\hbar(t) - \mathbf{u}_\hbar(t)||^4 + 8K_\hbar}}{4}, \\
u_\hbar(t+1) &= \frac{\rho_\hbar(t+1)}{1 + K_\hbar\rho_\hbar(t+1)}\mathbf{w}_\hbar^t(t)\mathbf{i}_{K_\hbar},
\end{aligned} \tag{20}$$

where $\vartheta(t+1)$ is the newly estimated value for ϑ at t-th learning cycle. In the t-th cycle of the M-step, network parameters are estimated as follows

$$\begin{aligned}
\mathbf{\Phi}(t+1) &= \rho_e(t+1)\mathbf{F}^{\mathrm{t}}(t+1)\mathbf{F}(t+1) + \rho_\hbar\mathbf{I}(t+1), \\
\boldsymbol{v}(t+1) &= \rho_e(t+1)\mathbf{F}^{\mathrm{t}}(t+1)\mathbf{y} + \rho_\hbar\mathbf{I}(t+1)\mathbf{u}(t+1), \\
\mathbf{w}(t+1) &= \mathbf{\Phi}(t+1)^{-1}\boldsymbol{v}(t+1).
\end{aligned} \tag{21}$$

Note that $\hbar$ means α and β in BBF2 while w in BBF1.

The learning algorithm is designed as follows

Step 1. Randomize the parameters and the hyperparameters.

Step 2. Input the training sub-sequences to the model using the existing parameters to estimate the model error.

Step 3. Estimate the hyperparameters.

Step 4. Estimate the parameters using the new values assigned to the hyperparameters.

Step 5. Check if the stop criterion is satisfied. If so, stop; otherwise, go to Step 2.

Stop criterion

From Equation (8), we can determine the stop criterion. When the error is approaching zero, ρ_e will be approaching a limit as follows

$$\lim_{||\mathbf{e}|| \to 0} \rho_e = \sqrt{\frac{M}{2}}. \tag{22}$$

As the error will never be zero, $\sqrt{M/2}$ will be the maximum value for ρ_e or the stop criterion could be any value of ρ_e which satisfies

$$\left|\rho_e - \sqrt{\frac{M}{2}}\right| < \varepsilon, \tag{23}$$

where ε is a small positive number. In practice, ε may vary with the value of M even for the same task. We normalize the equation by the number of training caspase sub-sequences

$$\left|\frac{\rho_e}{M} - \frac{1}{\sqrt{2}}\right| < \varepsilon'. \tag{24}$$

In the algorithm we choose $\varepsilon' = 0.1$.

Sub-sequence selection

In this study, the total number of non-cleaved sub-sequences is about 36 453, but the number of the cleaved sub-sequences is only 18. In order to see whether matched training data matters, the same number of non-cleaved training sub-sequences as the cleaved training sub-sequences are selected for modelling. Note that each selected non-cleaved training sub-sequence has the highest matching rate with a cleaved training sub-sequence and for each cleaved training sub-sequence, only one non-cleaved training sub-sequence is selected. Unfortunately, the prediction accuracy turned to the cleaved sub-sequences with zero specificity.

On the other hand, using all 36 453 sub-sequences certainly makes it difficult, in modelling, in terms of the computer memory. A trial-and-error method is used and it has been found when the ratio of the non-cleaved sub-sequences over the cleaved sub-sequences is ∼450, the computer program can be run in our current operating systems (512 MB RAM). Therefore, one non-cleaved training sub-sequence is randomly selected from among every five available non-cleaved ones leading to about 8200 non-cleaved training sub-sequences for modelling.

Modelling procedure

Step 1. Select the first sequence for testing.

Step 2. Scan the remaining 12 sequences for obtaining training sub-sequences.

Step 3. Divide the training sub-sequences into 10 subsets.

Step 4. Create 10 models using these 10 subsets. Each model is constructed using 9 subsets and validated on the remaining subset.

Step 5. Determine the best model in terms of the validation performance.

Step 6. Scan the testing sequence to generate testing sub-sequences.

Step 7. Use the best validation model for testing.

Step 8. Select the next sequence for testing.

Step 9. Repeat the Steps 2–8 till all the sequences are tested.

Measurement

Let TN, TP, FP and FN denote the true negatives (correctly identified non-cleaved sub-sequences), the true positives (correctly identified cleaved sub-sequences), the false positives (wrongly identified non-cleaved sub-sequences) and the false negatives (wrongly identified cleaved sub-sequences) respectively, and the three indicators used for measurements are:

$$\begin{gathered}\text{True negative fraction: TNf} = \text{TN}/(\text{TN} + \text{FP}) \\ \text{True positive fraction: TPf} = \text{TP}/(\text{TP} + \text{FN}) \\ \text{Total accuracy: TA} = (\text{TN} + \text{TP})/(\text{TN} + \text{FP} + \text{TP} + \text{FN}).\end{gathered} \tag{25}$$

IMPLEMENTATION

The packages are implemented using Java in Linux system with 512 MB RAM and 2 GHz.

RESULTS

The data were downloaded from NCBI (http://www.ncbi.nih.gov). Shown in Table 1 is the information of the sequences. Each sequence is composed of a couple of experimentally determined cleavage sites. Jackknife simulation is used. In each simulation run, one protein is ruled out and the remaining proteins are used for constructing a classifier. The constructed classifier is used to test the singled-out protein. The mean prediction accuracy and the standard deviation of the measurements are calculated. Each protein sequence was scanned by a sliding window with a fixed size, which varies from 10 to 20 with a gap of 2 in this study to investigate its impact on model performance.

Table 1. Thirteen proteins which are cleaved by caspase

Proteins	Gene	Length	Cleavage sites
O00273	*DFFA*	331	117(C3), 224(C3)
Q07817	*BCL2L1*	233	61(C1)
P11862	*GAS2*	314	279(C1)
P08592	*APP*	770	672(C6)
P05067	*APP*	770	672(C6), 739(C3 or C6 or C8 or C9)
Q9JJV8	*BCL2*	236	64(C3 and C9)
P10415	*BCL2*	239	34(C3)
O43903	*GAS2*	313	278(C)[a]
Q12772	*SREBF2*	1141	468(C3 and C7)
Q13546	*RIPK1*	671	324(C8)
Q08378	*GOLGA3*	1498	59(C2), 139(C3), 311(C7)
O60216	*RAD21*	631	279(C3 or C7)
O95155	*UBE4B*	1302	109(C3 or C7), 123(C6)

C2, Capsase 2; C3, Capsase 3; C7, Capsase 7; C6, Capsase 6; C8, Capsase 8; C9, Capsase 9.

[a] No further information is provided in NCBI.

Table 2. Algorithms for the investigation

Notation	Algorithm
SLP	Single layer perceptron
bSLP	Bayesian single layer perceptron
MLP	Multilayer perceptron
SVM	Support vector machines
BBF0	bio-basis function neural networks using a uniform distribution of weights
BBF1	Bayesian bio-basis function neural network using a single Gaussian of weights
BBF2	Bayesian bio-basis function neural network using two Gaussians of weights

Algorithms

Seven algorithms are used for the investigation (Table 2). When using SLP, bSLP, MLP (Duda *et al*., 2002) and SVM (Vapnik, 1995) each sub-sequence is encoded using the distributed encoding method (Qian and Sejnowski, 1988). SLP and bSLP are used for comparison because they are linear machines in the encoding space and bio-basis function neural networks are also linear machines in the space spanned by the bio-bases. MLP had totally biased prediction accuracy towards non-cleaved sub-sequences no matter how the number of hidden neurons vary. This is not surprising as this phenomenon has been investigated in the earlier study in Wilson and Sharda (1994), where the prediction accuracy was always biased towards the class with the majority of the inputs.

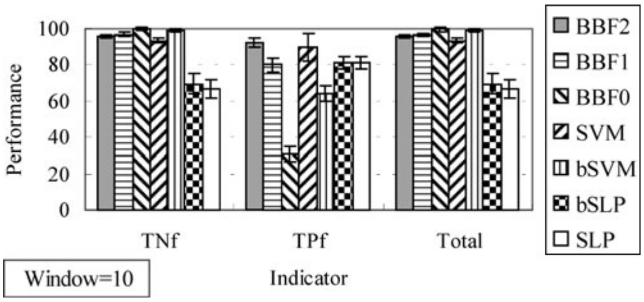

Fig. 1. The performance comparison for window size of 10. The horizontal axis denotes the three measurements and the vertical axis, the performance.

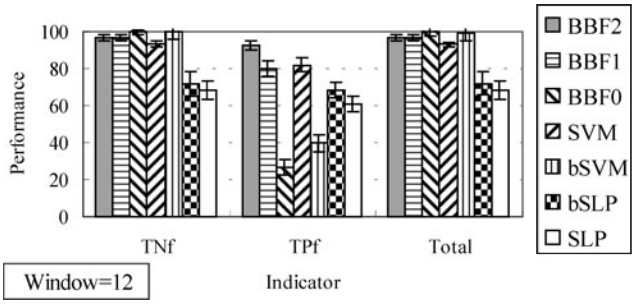

Fig. 2. The performance comparison for window size of 12. The horizontal axis denotes the three measurements and the vertical axis, the performance.

In the simulation, we limit the learning cycles to 1000 unless the stopping criterion or the steady state of the mean parameters is satisfied when estimating parameters for BBF1, BBF2 and bSLP. For SLP, the learning rate was 0.1 and the momentum factor was 0.08. Blosum62 matrix (Henikoff and Henikoff, 1992) was used for pairwise homology alignment calculation (Thomson *et al.*, 2003). When using SVM, linear and non-linear kernels (radial basis function, polynomial function and sigmoid function) were used for comparison. All non-linear function produced similar results as MLP while linear kernel worked the best with the C value as 100 for trading off between training error and regularization ability. The package SVM$^{\text{light}}$ (Joachims, 1999, http://svmlight.joachims.org/) was used.

Figure 1 shows the performance comparison for window size 10. It can be seen that although BBF0 demonstrated higher total accuracy than BBF2, it demonstrated a very low true positive fraction. The two best models are BBF2 and SVM. Their TNfs are 96 and 94%, their TPfs are 92 and 90% and their total accuracies are 96 and 93%. The P-values of the t-test on TNf, TPf and total accuracy between BBF2 and SVM are 0.0001, 0.8181 and 0.0002, respectively. The hypotheses that BBF2 shows similar TNf and total accuracy as SVM have been denied. At the same time the hypothesis that BBF2 shows similar TPf as SVM cannot be denied because the P-value is approaching one (0.8181). This means that BBF2 outperformed SVM in reducing the false cleaved sub-sequences significantly, while maintaining a similar performance in recognizing the cleaved sub-sequences.

In Figure 2, the pattern shown in Figure 1 still holds, where BBF2 and SVM are the best models. The P-values of the t-test on TNf, TPf and total accuracy are 7.0×10^{-6}, 0.4367 and 5.0×10^{-6}, respectively. Compared with the models using window size of 10, it can be seen that the hypotheses that BBF2 shows similar TNf and total accuracy as SVM have been strongly denied.

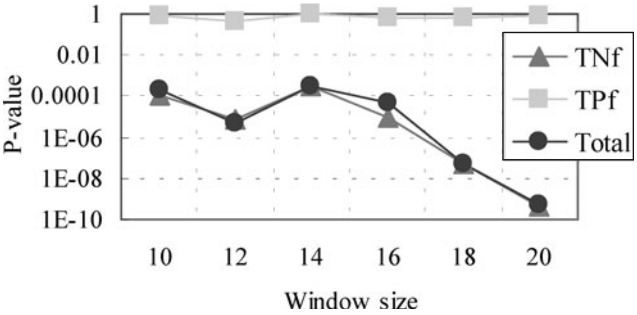

Fig. 3. The trend of the P-value of the t-test for comparing BBF2 with SVM.

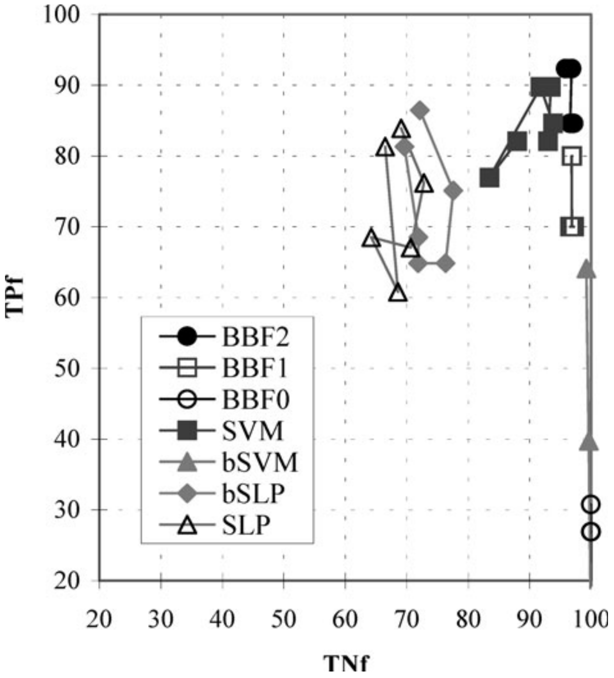

Fig. 4. The comparison among all the models using TNf and TPf.

When the window size increased to 20, BBF2 and SVM were still the best models. Figure 3 shows the P-values of the t-tests with a log scale. It can be seen that the trend shows that BBF2 outperforms SVM increasingly when the window size increases.

Figure 4 shows a comparison among all models using two measurements, TNf and TPf. The best model should be located at the top right corner. A failed model would be located at the left bottom corner. In terms of this, it can be seen that the BBF2 models outperformed all the other models because they are the closest to the top right corner.

DISCUSSION

This paper has presented a method of using Bayesian bio-basis function neural networks for the prediction of caspase cleavage sites in proteins. The experiments showed that BBF2 performed the best. Table 3 shows the prediction summary using BBF2.

Table 3. Prediction summary for BBF2

	10 (%)	12 (%)	14 (%)	16 (%)	18 (%)	20 (%)
TNf	95.86 ± 1.11	96.82 ± 1.64	96.57 ± 1.31	97.06 ± 1.69	97.18 ± 1.18	96.99 ± 1.68
TPf	92.31 ± 2.66	92.31 ± 2.66	84.62 ± 3.61	84.62 ± 3.61	84.62 ± 3.61	84.62 ± 3.61
Total	95.83 ± 1.10	96.79 ± 1.61	96.54 ± 1.25	97.03 ± 1.66	97.15 ± 1.13	96.96 ± 1.64

Table 4. Eighteen cleaved sub-sequences

Proteins	Gene	Cleavage expression (10-residue)
O00273	*DFFA*	VDETDSGAGL (C3), VDAVDTGISR (C3)
Q07817	*BCL2L1*	WHLADSPAVN (C1)
P11862	*GAS2*	ISRVDGKTSP (C1)
P08592	*APP*	EVKMDAEFGH (C6)
P05067	*APP*	EVKMDAEFRH (C6), VVEVDAAVTP (C3, C6, C8 or C9)
Q9JJV8	*BCL2*	AVHRDMAART (C3 and C9)
P10415	*BCL2*	WDAGDVGAAP (C3)
O43903	*GAS2*	ISRVDGKTSP (C)
Q12772	*SREBF2*	KDEPDSPPVA (C3 and C7)
Q13546	*RIPK1*	SLQLDCVAVP (C8)
Q08378	*GOLGA3*	GESPDGPGQG (C2), LCSTDSPLPL (C3), VSEVDGNDSD (C7)
O60216	*RAD21*	PDSPDSVDPV (C3 or C7)
O95155	*UBE4B*	SMDIDGVSCE (C3 or C7), QVDVDSGIEN (C6)

This work did not try Bayesian neural networks which use a single Gaussian prior for the weights. It was expected that Bayesian neural networks would not show good performance based on two factors. First, the single Gaussian prior of the weights did not outperform SLP and bio-basis function neural networks. Second, MLP did not work for both unmatched and matched training datasets.

The major drawback of this work is the lack of data. It is believed that the prediction accuracy will be further increased when more data are available. Nevertheless, this work has established an efficient computational methodology for the prediction of caspase cleavage sites.

The next issue is about sub-sequence length for modelling. It appears that it does not affect the prediction accuracy significantly although the accuracy varies with the size of the sliding window. A further investigation of the caspase structures is needed to determine empirically, the sub-sequence length thereby removing the trial-and-error process of determining the window size.

The third issue is whether the cleaved sub-sequences have very conserved patterns or expressions for classification. Eighteen 10-residue cleaved sub-sequences are listed in Table 4. It can be seen that there is a pattern XXXXD(S/T/G/A/M/V/C)XXXX. The prediction is then made using these expressions and the result is listed in Table 5. The maximum window size is limited to 10 as the result shown above demonstrated that this window size works equally well with the others. It can be seen that the accuracy of identifying cleaved sub-sequences is very low. For instance, the maximum averaged predicted cleaved sub-sequences is 2.89 (=52/18). The mean accuracy of predicting cleaved sub-sequences is then 0.17% (=2.89/17) for each cleaved expression or pattern.

Table 5. The prediction accuracy of using the cleaved expressions or patterns

Window	Total predicted cleaved sub-sequences	Averaged predicted cleaved sub-sequences
2	52	2.89
4	4	0.22
6	4	0.22
8	2	0.11
10	2	0.11

The last issue is the relationship between the Bayesian bio-basis function neural networks with the relevance vector machine (RVM) (Tipping, 2000; Li *et al*., 2002). The focus of RVM is to search for a minimum subset of kernels (bases) which can maximize the generalization ability of a classifier assuming that each weight follows a single Gaussian. However, the Bayesian bio-basis function neural networks proposed here recognizes the fact that the positive (cleaved) sub-sequences are generally not intensively and experimentally determined, hence they are scarcely collected. For instance, there are only 18 cleaved sub-sequences in this study. In the study of HIV protease cleavage sites, there are only 114 cleaved sub-sequences available (Thomson *et al*., 2003). This scarcity means that the positive (cleaved) sub-sequences normally do not show great similarity. A selection procedure on positive (cleaved) sub-sequences may not be appropriate. The Bayesian bio-basis function neural network aims to explore the most probable prior of the weight structure so that model performance can be optimized. Nevertheless, the RVM has also been investigated in this study. However, the algorithm always collapses when dealing with matrices. This has, in fact, been observed and analysed in the earlier study (Chen *et al*., 2003).

ACKNOWLEDGEMENTS

The author thanks Dr Joachims for the use of SVM$^{\text{light}}$. The author also thanks the reviewers and the associate editor for their valuable comments.

REFERENCES

Adams,J.M. and Cory,S. (1998) The Bcl-2 protein family: arbiters of cell survival. *Science*, **281**, 1322–1326.

Berry,E. *et al*. (2004) Reduced bio-basis function neural networks in prediction of phosphorylation sites, a comparative study. *Comput. Biol. Chem.*, **28**, 75–85.

Blom,N. *et al*. (1999) Sequence and structure based prediction of eukaryotic protein phosphorylation sites. *J. Mol. Biol.*, **294**, 1351–1362.

Cai,Y.D. and Chou,K.C. (1998) Artificial neural network model for predicting HIV protease cleavage sites in protein. *Advances in Engineering Software*, **29**, 119–128.

Chen,S. *et al*. (2003) Kernel-based nonlinear beamforming construction using orthogonal forward selection with Fisher ratio class separability measure. *IEEE Signal Processing Letters*, **11**, 478–481.

Chou,K.C. *et al*. (2000) Prediction of the tertiary structure of a caspase-9/inhibitor complex. *FEBS Lett.*, **470**, 249–256.

Duda,R.O., Hart,P.E. and Stork,D.G. (2002) *Pattern Classification and Scene Analysis*. 2nd edn. Wiley, NY.

Ethell,D.W. *et al*. (2001) Caspase 7 can cleave tumor necrosis factor receptor-I (p60) at a non-consensus motif, in vitro. *Biochim. Biophys. Acta*, **1541**, 231–238.

Evan,G. and Littlewood,T. (1998) A matter of life and cell death. *Science*, **281**, 1317–1322.

Gutteridge,A. *et al*. (2003) Using a neural network and spatial clustering to predict the location of active sites in enzymes. *J. Mol. Biol.*, **330**, 719–734.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Sci. Acad. USA*, **89**, 10915–10919.

Jakobi,R. (2004) Subcellular targeting regulates the function of caspase-activated protein kinases in apopsis. *Drug Resist. Updat.*, **7**, 11–17.

Joachims,T. (1999) Making large-scale SVM learning practical. In Schölkopf,B., Burges,C. and Smola,A. (eds), *Advances in Kernel Methods—Support Vector Learning*, MIT Press, Cambridge, MA.

Li,Y. *et al*. (2002) Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics*, **18**, 1332–1339.

Nabney,I. (2003) *Netlab: Algorithms for Pattern Recognition*, Springer.

Narayanan,A. *et al*. (2002) Mining viral protease data to extract cleavage knowledge. *Bioinformatics*, **18**, S5–S13.

Nielsen,H. *et al*. (1997) Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Eng.*, **10**, 1–6.

Qian,N. and Sejnowski,T.J. (1988) Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.*, **202**, 865–884.

Reed,J.C. and Paternostro,G. (1999) Postmitochondrial regulation of apoptosis during heart failure. *Proc. Natl Acad. Sci. USA*, **96**, 7614–7616.

Rohn,T.T. *et al*. (2004) Caspase activaion independent of cell death is required for proper cell dispersal and correct morphology in PC12 cells. *Exp. Cell Res.*, **293**, 215–225.

Thomson,R. *et al*. (2003) Characterising proteolytic cleavage site activity using bio-basis function neural networks. *Bioinformatics*, **19**, 1741–1747, 2003.

Thompson,T.B. *et al*. (1995) Neural network prediction of the HIV-1 protease cleavage sites. *J. Theor. Biol.*, **177**, 369–379.

Tipping,M.E. (2000) The relevance vector machine. *Adv. Neural Inf. Process. Syst.*, **12**, 652–658.

Van de Craen,M. *et al*. (1999) Identification of caspases that cleave presenilin-1 and presenilin-2. *FEBS Lett.*, **445**, 149–154.

Vapnik,V. (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag, NY.

West,K.A. *et al*. (2002) Activation of the P13K/Akt pathway and chemotherapeutic resistance. *Drug Resist. Updat.*, **5**, 234–248.

Wilson,R.L. and Sharda,R. (1994) Bankruptcy prediction using neural networks. *Decision Support Systems*, **11**, 545–557.

Yang,Z.R. and Chou,K.C. (2003) Mining biological data using Self-organising Map. *J. Chem. Inf. Comput. Sci.*, **43**, 1748–1753.

Yang,Z.R. and Chou,K.C. (2004) Bio-basis function neural networks for the prediction of the O-linkage sites in glyco-proteins. *Bioinformatics*, **20**, 903–908.