Bioinformatics, 2019, 1–9
doi: 10.1093/bioinformatics/btz721
Advance Access Publication Date: 30 September 2019
Original Paper

Sequence analysis

DeepCleave: a deep learning predictor for caspase and matrix metalloprotease substrates and cleavage sites

Fuyi Li [1,2,†], Jinxiang Chen [1,3,†], André Leier [4,5], Tatiana Marquez-Lago [4,5], Quanzhong Liu [3], Yanze Wang [3], Jerico Revote [1], A. Ian Smith [1], Tatsuya Akutsu [6], Geoffrey I. Webb [2], Lukasz Kurgan [7,*] and Jiangning Song [1,2,8,*]

[1]Monash Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, VIC 3800, Australia, [2]Monash Centre for Data Science, Faculty of Information Technology, Monash University, Melbourne, VIC 3800, Australia, [3]College of Information Engineering, Northwest A&F University, Yangling 712100, China, [4]Department of Genetics and [5]Department of Cell, Developmental and Integrative Biology, School of Medicine, University of Alabama at Birmingham, Birmingham, AL, USA, [6]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan, [7]Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA and [8]ARC Centre of Excellence in Advanced Molecular Imaging, Monash University, Melbourne, VIC 3800, Australia

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
Associate Editor: Arne Elofsson

Received on June 7, 2019; revised on August 13, 2019; editorial decision on September 16, 2019; accepted on September 25, 2019

Abstract

Motivation: Proteases are enzymes that cleave target substrate proteins by catalyzing the hydrolysis of peptide bonds between specific amino acids. While the functional proteolysis regulated by proteases plays a central role in the 'life and death' cellular processes, many of the corresponding substrates and their cleavage sites were not found yet. Availability of accurate predictors of the substrates and cleavage sites would facilitate understanding of proteases' functions and physiological roles. Deep learning is a promising approach for the development of accurate predictors of substrate cleavage events.

Results: We propose DeepCleave, the first deep learning-based predictor of protease-specific substrates and cleavage sites. DeepCleave uses protein substrate sequence data as input and employs convolutional neural networks with transfer learning to train accurate predictive models. High predictive performance of our models stems from the use of high-quality cleavage site features extracted from the substrate sequences through the deep learning process, and the application of transfer learning, multiple kernels and attention layer in the design of the deep network. Empirical tests against several related state-of-the-art methods demonstrate that DeepCleave outperforms these methods in predicting caspase and matrix metalloprotease substrate-cleavage sites.

Availability and implementation: The DeepCleave webserver and source code are freely available at http://deepcleave.erc.monash.edu/.

Contact: lkurgan@vcu.edu or Jiangning.Song@monash.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Protease substrate cleavage plays important roles in a variety of biological processes, such as cell cycle, pathway regulation and protein degradation (Hilt and Wolf, 1995; López-Otín and Overall, 2002). Knowledge of protease-specific substrate cleavage is important for understanding the mechanisms and biological functions of proteases. In contrast to relatively expensive and time-consuming conventional experimental methods for identifying protease substrate cleavage events, computational methods provide a more cost- and time-efficient alternative that is suitable for proteome-wide annotation and which can be used to guide hypothesis-driven experimental design.

Several computational predictors of the protease-specific substrates and cleavage sites that rely on machine learning algorithms have been developed in the past two decades (Li *et al.*, 2018a). They include Pripper (Piippo *et al.*, 2010), Cascleave (Song *et al.*, 2010), PROSPER (Song *et al.*, 2012), LabCaS (Fan *et al.*, 2013), ScreenCap3 (Fu *et al.*,

2014), CleavPredict (Kumar *et al.*, 2015), PROSPERous (Song *et al.*, 2018a), iProt-Sub (Song *et al.*, 2018b) and Procleave (http://procleave.erc.mo-nash.edu), etc. These methods rely on a variety of different features extracted from the input protein sequences, such as amino acid frequencies, information extracted position-specific scoring matrices, and a wide range of physicochemical properties of amino acids (Chen *et al.*, 2018, 2019). These features are used to train predictive models utilizing several different types of machine learning algorithms. While the strategy that depends on the feature-based sequence encoding has resulted in the development of several well-performing predictors, it has a few shortcomings. First, the already large feature space must be enlarged by combination of existing features and manual design of additional features in order to further improve the prediction performance. The design of new and informative features is typically done via a trial-and-error approach that requires a substantial amount of manual work. Second, the manually developed features could be irrelevant to this prediction and/or redundant (correlated), which negatively impacts the training of the accurate predictive models with the machine learning algorithms. Thus, inclusion of the new features usually involves application of feature selection techniques to reduce the risk of using irrelevant and redundant features. Third, the design of the new features and the use of feature selection methods have to be coupled with the selection of a suitable machine learning algorithm. To sum up, the design of these methods is rather complex and requires handing of three tasks: feature design, feature selection and algorithm selection.

Deep learning-based approach to building the predictive models alleviates these issues. In contrast to the conventional feature-based methods, deep learning is a form of representation learning. That is, it automatically learns a suitable representation from the raw input data, such as protein sequences, without the need to design and select features. Furthermore, the use of the deep learning models, especially when combined with the application of transfer learning, may produce competitive predictive quality when compared with more conventional machine learning-based methods. Consequently, several deep learning-based methods for the protein sequence analysis were published in recent years. For instance, deep neural networks were used for the prediction of protein crystallization (Elbasir *et al.*, 2018), PTM sites (Wang *et al.*, 2018), phosphorylation sites (Luo *et al.*, 2019; Wang *et al.*, 2017), promoters (Umarov *et al.*, 2019) and protein function (Zhang *et al.*, 2019). However, deep learning has not been so far used for the prediction of the protease-specific substrate cleavage sites (Li *et al.*, 2018a).

We introduce DeepCleave, the first deep learning framework for the caspase and matrix metalloprotease substrate cleavage site prediction. Our approach does not require manual feature engineering and transfers generic protease-family models using transfer learning to generate accurate protease-specific predictors. The use of the transfer learning addresses the problem of relatively small sample sizes of the protease-specific substrate cleavage site datasets. Empirical tests illustrate that the use of the transfer learning improves the quality of the protease-specific substrate cleavage sites prediction when compared to the deep network designed without the transfer learning. Extensive empirical benchmark on an independent test dataset demonstrates that DeepCleave outperforms current state-of-the-art computational approaches. A user-friendly and free webserver that implements DeepCleave is available at http://deepcleave.erc.monash.edu/.

2 Materials and methods

2.1 Design and assessment process

We summarize the design and assessment process of DeepCleave in Figure 1A. There are four major steps in this process: (i) dataset collection, (ii) model training, (iii) performance evaluation and (iv) webserver construction. In the first step, we collect the benchmark and independent test datasets from the MEROPS database (Rawlings *et al.*, 2018). In the second step, we design and optimize a convolutional neural network (CNN) (LeCun *et al.*, 2010) using the training dataset. In the third step, we comparatively evaluate the trained CNN models on the independent test against the existing state-of-the-art methods. In the fourth step, we implement and release the DeepCleave webserver and the corresponding source code.

2.2 Dataset collection

We extract the experimentally validated protein substrate annotations from the release 12.0 of the MEROPS database (Rawlings *et al.*, 2018). We reduce sequence similarity of the corresponding substrate sequences using the CD-HIT program (Fu *et al.*, 2012) with the identity threshold of 50% at full protein sequence level. We randomly partition the remaining sequences into the training dataset and the independent test dataset (not used for training) with a ratio of 7:3. Next, we further reduce identity between the test proteins and the proteins from the training dataset to 20% by clustering both datasets together using CD-HIT with the identity threshold of 50% and removing the test proteins that are in clusters with the training proteins. This ensures that the remaining test proteins share <20% identity with the training dataset, while we keep the 50% pairwise similarity within the training set to enlarge the amount of the training data. We note that the 20% cut-off is stricter than the similarity levels maintained in other related studies, which include 80 (Fu *et al.*, 2014), 70 (Song *et al.*, 2010; Song *et al.*, 2018a, b) and 50% (Wang *et al.*, 2017). Details concerning the size and composition of the training and test datasets are summarized in Supplementary Tables S1–S3. We use the training dataset exclusively to optimize the design and parameters of DeepCleave. We utilize the independent (low similarity) test dataset to validate the predictive performance of DeepCleave and compare it with the existing methods.

2.3 Model training

We use the substrate sequences as input and we employ the one-hot encoding to present these sequences for the CNN. We train CNN using the training dataset to self-learn features that best represent information that is relevant for the protease substrate cleavage site prediction from the one-hot encoded input chains. The output from the DeepCleave predictor consists of two numeric scores for each residue in the input protein sequence: one that quantifies propensity for cleavage site and the other that quantifies propensity for non-cleavage site. The two scores are combined together to produce a binary prediction, i.e. every residue is predicted as either cleavage site (if cleavage site score > non-cleavage site score) or non-cleavage site (otherwise). We provide further details in the following sections.

2.3.1 One-hot encoding of the input protein sequence

The CNN model requires the input with a fixed length, while the lengths of the substrate sequences vary widely. Thus, we use a local sliding window approach with a fixed window size of 30 (P15-P15′ sites: 15 residues upstream and downstream of the cleavage site). We pad the positions of the window that extend beyond the protein sequence at either terminus with symbol X. We encode the protease subsequence using the one-hot encoding which produces a 21-dimensional vector (20 types of common amino acids and X) with a value of 1 corresponding to the amino acid in the sliding window and 0 at all other positions. Consequently, the input used to predict the cleavage site for the residue in the middle of the window is 21×30 matrix. Each residue/matrix is labeled as 1 (if this is a native cleavage site) or 0 (otherwise) for the purpose of training the CNN network.

2.3.2 Architecture of the deep CNN

We use the Keras package (Gulli and Pal, 2017) with a Theano backend (Team *et al.*, 2016) to implement the DeepCleave model. Classical CNNs consist of a convolution layer, max-pool layer and fully connected layers from lower layers to higher layers. The lower layers learn simple sequence features which aggregate into more complex features in the higher network levels. The topology of the CNN used in DeepCleave is shown in Figure 1B. It consists of three convolutional layers, attention layer, two fully connected layers and

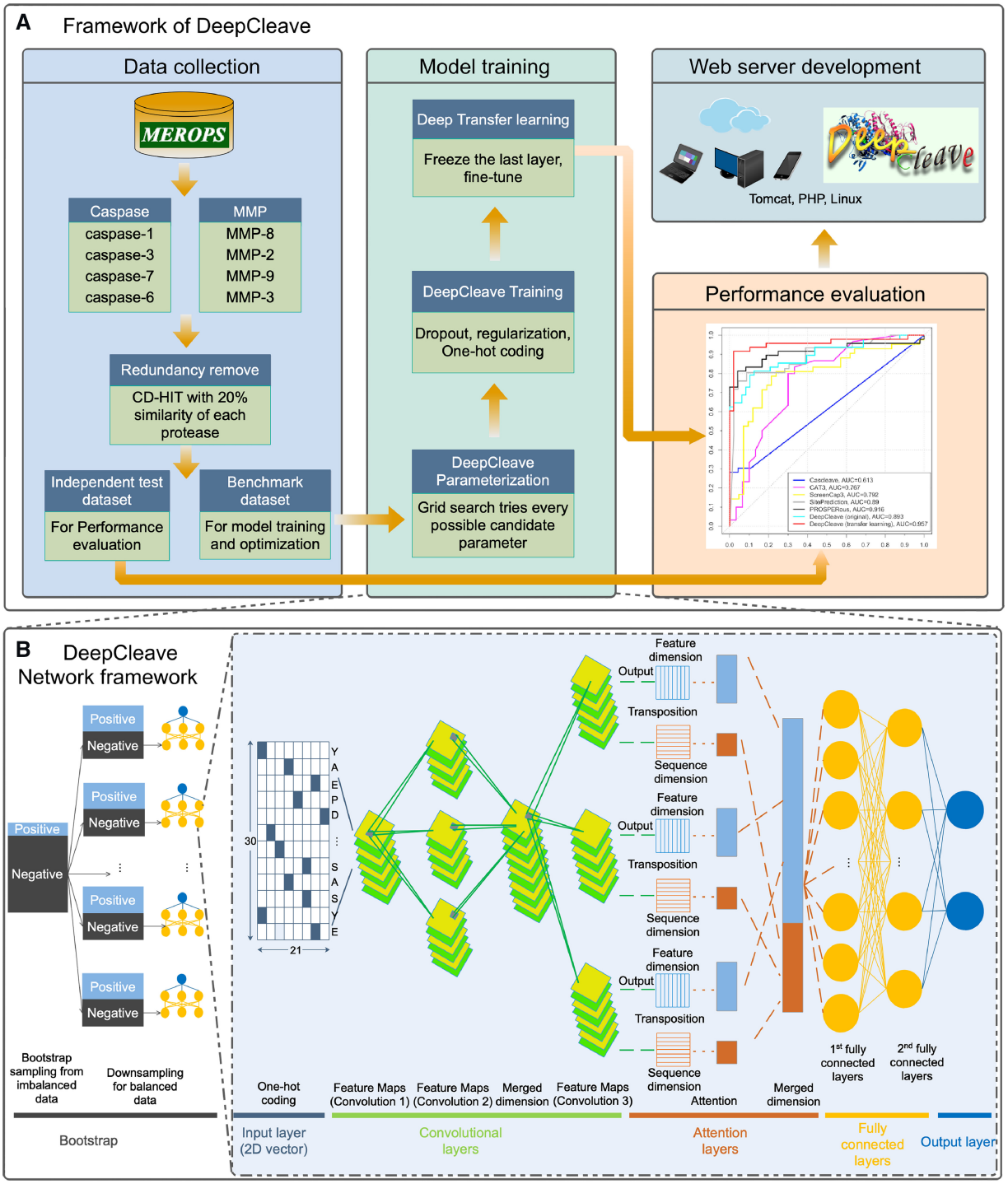

Fig. 1. Development flowchart and the deep network architecture. (**A**) The flowchart of the development and assessment process. (**B**) The topology of the deep CNN. The CNN takes input sequences and sequentially transforms them into a 'flattened' output vector using convolutional, pooling and fully connected layers. The elements of the output vector (softmax layer) represent the probabilities of the cleavage sites. During the training process, the internal parameters of the neural network layers are iteratively adjusted to improve accuracy. Typically, lower layers (left side of B) learn simple features, which then influence the high-level representations (right side of B)

the output layer. We describe these layers in the subsequent paragraphs.

The three convolutional layers that aim to capture features from the one-hot encoding matrix. In the first convolutional layer, we use kernel size $= 1 \times 200$ (convolutional filter in first convolutional layer) to extract simple features from the one-hot encoding matrix. The convolution of the kernel matrix and the input portion of the neuron window size is the output of the neurons on each convolutional layer. The second convolutional layer uses three parallel convolution blocks, each with a different convolution window size (kernel sizes $= 3 \times 150$, 6×150 and 9×150; convolutional filter $= 150$ in the second convolutional layer) to convert the features from the first convolutional layer in a parallel manner. We apply three different kernel sizes to diversify the extracted features, ultimately leading to a potentially more robust and more accurate predictive model. This strategy diversifies the high-level features that are

extracted from the features generated in the previous layer. Next, we utilize a merge layer to combine the feature representations generated by the three convolution blocks into a higher-dimensional feature representation. The third convolutional layer also uses three convolution blocks with different convolution window sizes (kernel sizes $= 5 \times 200$, 10×200 and 15×200; convolutional filter $= 200$ in the third convolutional layer) to further diversify and improve the extracted features. A detailed visualization of three different kernel sizes in the first and the second convolutional layer is provided as an example in Supplementary Figure S1.

The attention layer aims to selectively discover relevant features from a large number of features generated in the convolutional layers. Inspired by the implementation of attention mechanism in previous studies (Luo *et al.*, 2019; Wang *et al.*, 2017), we implement the DeepCleave's attention layer to learn two types of feature representations from the output of each convolution block of the third convolution layer. One representation considers the direction of the sequence and the other focuses on the direction of the features. The 'transposition' in Figure 1 means transposed matrix. The feature representation matrix in the direction of features is the transposed matrix in the direction of sequence. This results in the total of six feature representations that are combined together in the merge layer.

The merged attention layer is followed by two fully connected layers. These two layers reassemble more localized features produced in the merge layer to produce features that cover the entire context of the input matrix. They also act as classifiers that map the resulting feature space onto the corresponding labels using nonlinear transformations. The two fully connected layers use 149 and 8 neurons, respectively.

The output layer has two neurons that quantify propensity for cleavage site and for the non-cleavage site. The two neurons are fully connected to the previous layer and the activation function is softmax. The softmax activation is commonly used in the final output layer to distribute the probability throughout each of the multiple output nodes (Armenteros, 2019; Luo *et al.*, 2019; Wang *et al.*, 2017, 2018). We use transfer learning to convert generic protease family models into protease-specific models. We implement the transfer learning by keeping the layers before the 2nd fully connected layer of the base network (protease-family level model) frozen and training the 2nd fully connected layer and output layer for protease-specific cleavage sites prediction.

2.3.3 Training of the CNN

We employ the 'Adam' optimizer (Kingma *et al.*, 2014) with the classification cross-entropy as a loss function to train our model. Moreover, we use grid search to adjust the DeepCleave hyperparameters. We utilize several strategies that are detailed below to prevent over-fitting into the training dataset. They include the use of ReLU activation function, L2 regularization, dropout (Sainath *et al.*, 2013) and 'early stopping' (Yao *et al.*, 2007).

ReLU, which fixes the gradient disappearance problem in the backpropagation training algorithm, is defined as

$$\mathrm{ReLU} = \begin{cases} 0 & if\ x < 0 \\ x & else \end{cases}.$$

L2 regularization imposes large penalties on sparse spiked weight vectors, preferring uniform parameters. This results in the neural nodes taking advantage of a larger number of the inputs coming from the upper network layer, rather than only a part of the input. After the L2 term is added, the absolute value of the weights tends to overall decrease, especially if there is no particularly large value, that is, the network tends to learn relatively small weights.

Dropout refers to the random removal of some neurons ('erasure' of these neurons from the network) when training a large neural network. Since randomly removed neurons are different in each batch of the training process, the corresponding networks are also different, resulting in 'new' models. Dropout reduces a potentially harmful co-adaptation of neurons because this way neurons do not depend on the presence of other specific neurons. Therefore, the network is forced to learn new features that are used in combination to improve predictions. As such, dropout is a useful to ensure that the prediction network model is robust to the loss of individual features (Krizhevsky *et al.*, 2017).

The 'early stopping' strategy stops training when the loss on the training set is not decreasing (i.e. the degree of reduction is less than a certain threshold). This solves the problem of manually setting the number of epochs and reduces chances of overfitting the network to the training set.

2.4 Balancing the training dataset

The number of cleavage sites is much smaller than the number of non-cleavage sites. We apply bootstrapping (Wallace *et al.*, 2011) to tackle this imbalance problem in the training dataset. We visualize this strategy in the 'Bootstrap' part of Figure 1B. Let $\boldsymbol{P}$ and $\boldsymbol{N}$ be the positive set (cleavage sites) and negative set (non-cleavage sites), respectively, and $\#\boldsymbol{P}$ and $\#\boldsymbol{N}$ be the number of the corresponding positive and negative residues. We selected the same numbers ($\#\boldsymbol{P}$) of negative and positive residues to train a model in each bootstrap iteration. We split the negative residues into $n = N/\mathrm{P}$ subsets and we apply n bootstrap iterations to traverse the negative residue and train one prediction model. We repeat this procedure five times to train five prediction models. We use the average output of these five models as the final prediction.

2.5 Optimization of the model training parameters

We tune the training parameters of the CNN to maximize the predictive performance. We use 90% of the training dataset to perform bootstrapping and the remaining 10% of the training dataset for validation. We employ Bayesian optimization (Snoek *et al.*, 2012) to tune the following parameters: learning rate (values in the 0.0001–0.1 range), L2 regularization weight decay (0.0001–0.1), the batch size (64–1024), the dropout probability (0.2–0.8), the convolutional filter (100–250) and the dense filter (8–200). We show the best performing on the validation set parameter values in Table 1.

2.6 Evaluation metrics

We assess the predictive performance with five commonly used measures including sensitivity (Sn), specificity (Sp), precision, accuracy (Acc) and Matthew's Correlation Coefficient (MCC):

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP}) \times (\mathrm{TP} + \mathrm{FN}) \times (\mathrm{TN} + \mathrm{FP}) \times (\mathrm{TN} + \mathrm{FN})}}$$

where TP, TN, FP and FN denote the numbers of true positives (correctly predicted cleavage sites), true negatives (correctly predicted

Table 1. Values of the tuned parameters

Parameters	Tuned setting
Batch size	1024
Learning rate	0.001
L2 regularization	0.001
Dropout rate	0.75
'Early stopping' patience	20
Initializer	he_normal
Convolutional filter	200, 150, 200
Dense filter	149, 8, 2
Activation function	ReLU

non-cleavage sites), false positives (non-cleavage sites incorrectly predicted as cleavage sites) and false negatives (cleavage sites incorrectly predicted as non-cleavage sites), respectively. Moreover, we also plot the Receiver-Operating Characteristic (ROC) curves and calculated the Area Under the Curve (AUC) values based on the scores produced by the output layer.

3 Results and discussion

3.1 Predictive performance of the protease-family and protease-specific deep CNNs

The protease-specific datasets have relatively small sizes when compared to the needs of the deep network training. Small training datasets may cause overfitting when used to train deep networks (Yosinski *et al.*, 2014). To address this, we utilize the deep transfer learning technique that is commonly used in the deep learning studies (Hurtado *et al.*, 2018; Wang *et al.*, 2017, 2018). The deep transfer learning first trains a base network, and then copies the first n layers of this base network to the first n layers of the target network. Next, the remaining layers of the target network are randomly initialized and trained for a target problem. There are two main strategies for training the target network. The first is to back-propagate errors in the entire target problem network to fine-tune them to the new problem. The second is to keep the transferred feature layers frozen, which means that they are fixed during the training of for the target problem. Choosing whether to fine-tune the first n layers of the target network depends on the size of the target dataset. If the dataset is small and the number of parameters is large, fine-tuning may lead to over-fitting and thus these layers should be kept frozen. On the other hand, if the target dataset is large or the number of parameters is small, the over-fitting should not be a concern and the base layers should be fine-tuned (Yosinski *et al.*, 2014). We apply the second strategy to implement the deep transfer learning since the target dataset is small and the number of parameters is relatively large. We first generate protease-family level deep CNNs for the caspases and matrix metallopeptidases (MMPs) (by combining all the caspases/MMPs cleavage data together) given the relatively small sizes of the protease-specific substrate cleavage site data. Next, we copy the all layers before the 2nd fully connected layer of the base network (protease-family level model), keep these layers frozen, and trained the 2nd fully connected layer and output layer to produce protease-specific predictors. We evaluate and compare the predictive performance of the family level and protease-specific deep CNNs in this section. We contrast the protease-specific predictors with current state-of-the-art predictors in Section 3.4.

Figure 2 shows change in the average training loss and accuracy for the two family-level networks over the consecutive training epochs. We monitor accuracy changes when testing for when to stop training. Therefore, the training process of the network for the caspase cleavage site prediction ('caspase base network') converges after about 400 epochs, while the training for the MMP cleavage site prediction ('MMP base network') requires about 1000 epochs to converge. The caspase base network secures a lower training loss than the network for MMPs. Moreover, the caspase base network converges to a higher accuracy (>0.9 after about 400 epochs) than the MMP base network (around 0.8). However, both results indicate that these networks provide high-quality family level predictions of the cleavage sites. These accurate results provide a strong foundation for the transfer learning of the protease-specific predictors.

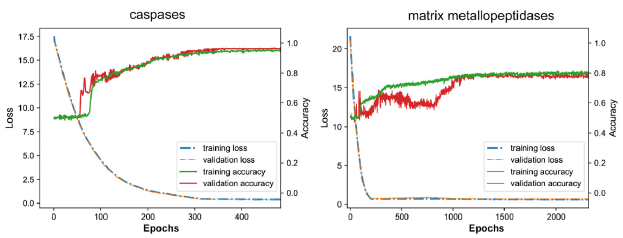

Fig. 2. The average training loss and accuracy of the family level networks for the caspase (left) and for the matrix metallopeptidase (right) cleavage site prediction

We also compare the predictive performance of the DeepCleave models trained with and without transfer learning. We summarize these results in Supplementary Tables S4 and S5. The results reveal that the DeepCleave models trained with transfer learning achieves a significantly better performance than the models trained without transfer learning for all 12 proteases. The average AUC of DeepCleave trained with transfer learning over the 12 proteases is 0.92, while the models trained without transfer learning achieve an average AUC of 0.75. In addition, DeepCleave trained with transfer learning also achieves a significantly better accuracy (0.89 versus 0.76), MCC (0.77 versus 0.52), sensitivity (0.88 versus 0.79) and precision (0.89 versus 0.76) than the models trained without the transfer learning. Taken together, these results indicate that the transfer learning strategy is effective for training accurate DeepCleave models using limited protease-specific cleavage data.

3.2 Ablation analysis for the protease-specific deep CNNs on the training dataset

As shown in Figure 1B, the DeepCleave's network uses three different kernel sizes in the second (kernel sizes = 3, 6, 9) and the third (kernel sizes = 5, 10, 15) convolutional layers. The two-dimensional attention layer is applied after the convolutional layers. We perform an ablation analysis to investigate whether the inclusion of the attention layer and the three different kernel sizes in the convolutional layers provide improvements in the predictive performance. Specifically, we compare results generated by DeepCleave (the complete architecture with the attention layer and three kernel sizes) with (i) DeepCleave without the attention layer, (ii) DeepCleave with only one kernel size (one kernel size model) in the second and the third convolutional layers (kernel sizes are 3 and 5, respectively) and (iii) DeepCleave with two kernel sizes (two kernel sizes model) in the second (kernel sizes are 3 and 6) and the third convolutional layers (kernel sizes are 5 and 10). The comparison was done based on the 5-fold cross-validation tests on the training dataset. We summarize the results in Figure 3 and Supplementary Figure S2.

The results reveal that the complete DeepCleave framework has achieved the best predictive performance for all test scenarios. On the other hand, the DeepCleave without the attention layers performed the worst, with the exception of MMP-9 where the DeepCleave version with one kernel size model performed the worst. These results demonstrate that the attention layer plays an important role in ensuring high quality of predictions produced DeepCleave. Moreover, the empirical results also justify the use of the three kernel sizes in the second and the third convolutional layers of the DeepCleave framework.

3.3 Feature representation in the DeepCleave predictor

Figure 4 gives the UMAP plots (McInnes *et al.*, 2018) that visualize feature representations that are automatically learned inside of the DeepCleave model. The UMAP plots cluster the actual feature representations into a two-dimensional space. This figure includes the mapped feature representations for the one-hot encoding, after the attention layer, and for the 2nd fully connected layer. Each dot in the figure represents a positive sample (i.e. a cleavage site for a given protease). Figure 4A and B reveals that the one-hot encoding cannot be directly used to accurately discriminate proteases. The cleavage data for the different proteases are almost randomly distributed within the UMAP plot space. However, use of the attention layer visibly improves the ability to discriminate protease cleavage data (Fig. 4C and D). The features represented further down the network at the 2nd fully connected layer generate even better results (Fig. 4E and F). The UMAP plots demonstrate that the DeepCleave framework learns informative feature representations from the one-hot encoding that is easy to extract from the input protein chains. However, even the results at the 2nd fully connected layer show overlap between some cleavage points associated with different caspases. This is not surprising since some sites are cleaved by several different proteases. In addition, we obtain similar results when using another visualization tool, t-SNE (van der Maaten and Hinton, 2008). The t-SNE plots are shown in Supplementary Figure S3.

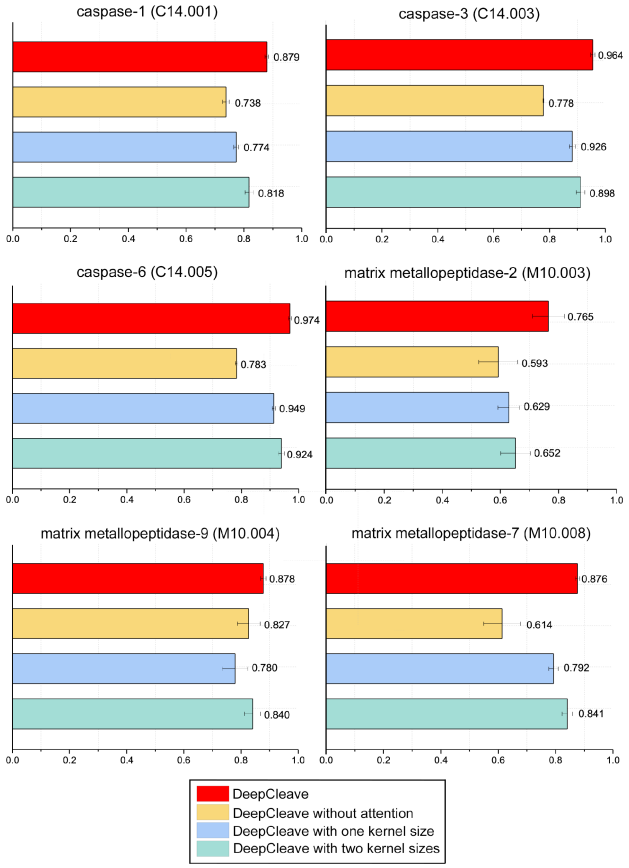

Fig. 3. Comparison of the predictive performance for the four models considered in the ablation study on the test dataset using five-fold cross-validation. The results concern six proteases: caspase-1, caspase-3, caspase-6, matrix metallopeptidase-2, matrix metallopeptidase-9 and matrix metallopeptidase-7; the identifier inside the brackets (e.g. 'C14.005') is the protease ID in MEROPS. The following models are included: DeepCleave, DeepCleave without attention layer, DeepCleave with only one kernel size in the second and the third convolutional layer, and DeepCleave with two kernel sizes in the second and the third convolutional layer

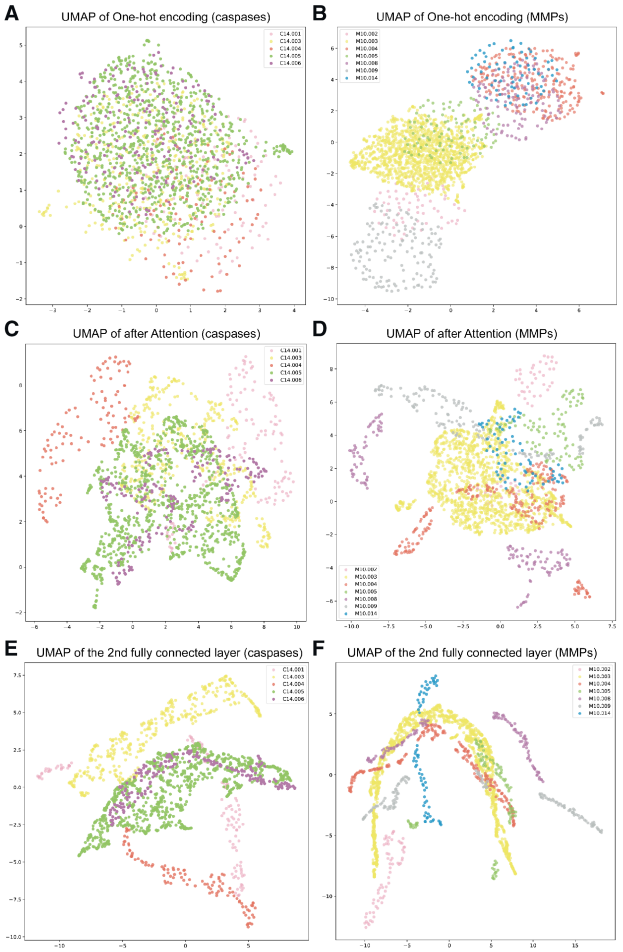

Fig. 4. UMAP plots of the input one-hot encoding (**A** and **B**), feature representation after the attention layer (**C** and **D**) and the feature representation of the 2nd fully connected layer (**E** and **F**) for the models for the caspases (on the left) and the MMPs (on the right). These results were produced using the training dataset

To sum up, our empirical results in Sections 3.1–3.3 suggest that the models trained on the protease-family cleavage data can be used to develop accurate cleavage prediction models.

3.4 Comparison of predictive performance on the test dataset

We compare the predictive performance of the protease-specific DeepCleave models on the independent test dataset (up to 20% similarity with the training dataset) against state-of-the-art prediction tools that have been developed for the caspase and MMP cleavage sites prediction. The considered tools include Cascleave, CAT3, ScreenCap3, SitePrediction and PROSPERous. We collect predictions from these methods using their webservers or implementations provided by the authors. We provide the corresponding ROC curves in Figure 5 and Supplementary Fig. S4. Moreover, we report MCC, ACC, sensitivity, specificity and precision for these methods for the five caspases and the seven MMPs in Supplementary Table S6.

DeepCleave achieves competitive predictive performance measured with AUC. Specifically, for the five tested caspases (caspase-1, caspase-2, caspase-3, caspase-6 and caspase-7) and four out of the seven tested MMPs (MMP-2, MMP-7, MMP-12 and membrane-type MMP-1), it secures the best AUC value. For the other three types of MMPs (MMP-3, MMP-8 and MMP-9), PROSPERous achieves the best AUC values for two, SitePrediction for one, while DeepCleave ranks either second (MMP-8 and MMP-9) or third (MMP-3). The DeepCleave's average AUC over the 12 proteases is 0.947. When compared with the second-best PROSPERous on the four caspases that both methods can predict, DeepCleave secures average AUC = 0.981 versus 0.965 for PROSPERous. For the seven MMPs, DeepCleave achieves average AUC = 0.920 versus 0.910 for PROSPERous. The two predictors with the third and fourth highest average AUCs are SitePrediction (average AUC = 0.872 over the four caspases and six MMPs that it predicts) and ScreenCap3 (average AUC = 0.869 over the five caspases it covers). DeepCleave provides the average AUCs = 0.985 (caspases) and 0.920 (MMPs) for the two corresponding sets of proteases, respectively. Similar observation can be made for the assessment of the binary predictions using MCC and accuracy. The average MCC of DeepCleave equals 0.828 over all proteases, 0.945 for the five caspases and 0.744 for the seven MMPs. These values reveal that the correlation between the cleavage sites predicted by DeepCleave and the native annotations is high. The average (over the 12 proteases) accuracy of DeepCleave is 0.914 with balanced values of specificity (0.921), sensitivity (0.908) and precision (0.916). Overall, the empirical tests demonstrate that DeepCleave provides accurate predictions of the caspase- and MMP-specific cleavage sites that outperform results generated by the currently available tools.

3.5 One-hot encoding provides favourable predictive quality

Previous studies have used a variety of input including the composition of *k*-spaced amino acid pairs (CKSAAP), BLOSUM62 matrix, position-specific scoring matrix (PSSM) and sequence conservation to predict protease-specific cleavage sites (Song *et al.*, 2018b; Wang

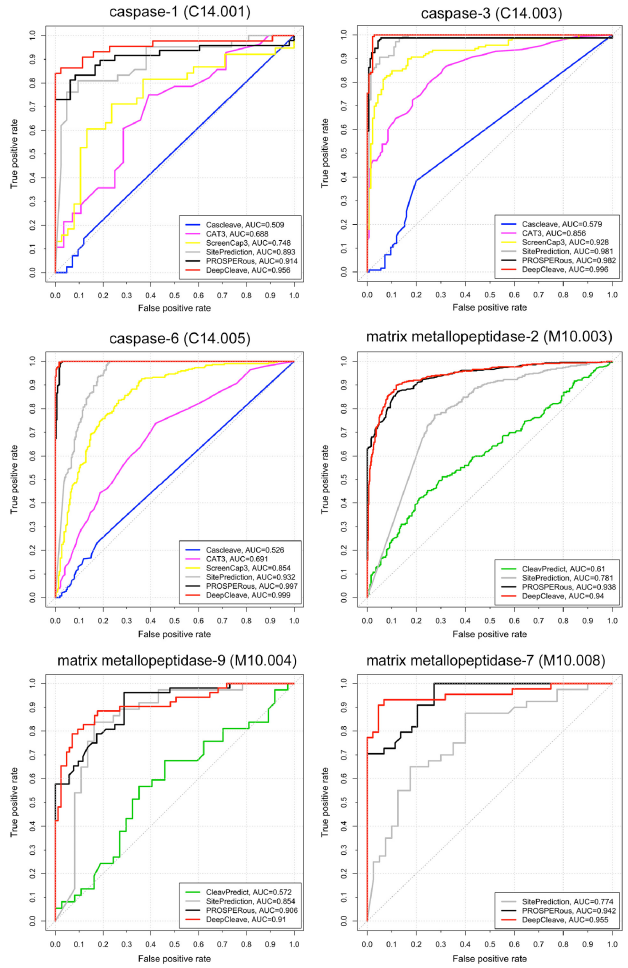

Fig. 5. ROC curves and corresponding AUC values generated by seven considered protease cleavage site predictors (DeepCleave, Cascleave, SitePrediction, CleavPredict, CAT3, ScreenCap3 and PROSPERous) for caspase-1, caspase-3, caspase-6, matrix metallopeptidase-2, matrix metallopeptidase-9 and matrix metallopeptidase-7; the identifier in the brackets (e.g. 'C14.005') is the protease ID in MEROPS

et al., 2014). Predicted structural features, such as secondary structures and solvent accessibility, are also used to predict functional sites in proteins (Zhang *et al.*, 2010, 2017). We investigate whether these inputs can be used to further improve the prediction performance of DeepCleave. We compare performance between the DeepCleave and the deep networks trained using each feature types individually (CKSAAP, BLOSUM62, putative secondary structure, putative solvent accessibility, PSSM with the PSSM-derived conservation scores) and using all inputs together. The PSSM was calculated by performing PSI-BLAST search against the UniRef90 database. The secondary structure was predicted with PSIPRED (Jones, 1999) while solvent accessibility was predicted with ASAquick (Faraggi *et al.*, 2014). We provide a detailed description of how these data were encoded into the inputs for the DeepCleave's network in Supplementary Methods section in Supplementary Materials.

We compare the predictive quality of these approaches on the training and independent test datasets in Supplementary Tables S7 and S8, respectively. Results reveal that majority of these input types, except for the putative secondary structure, can be used to predict the cleavage sites reasonably well. The average AUCs computed over the 12 proteases equal 0.605 for the putative secondary structure, 0.766 for CKSAAP, 0.798 for the putative solvent accessibility, 0.806 for the PSSM and conservation scores, 0.922 when using the BLOSUM62 matrix-derived inputs and 0.947 when using the one-hot encoding from DeepCleave. Similar trend is true when using the average MCC, with the corresponding values equal 0.181, 0.463, 0.527, 0.510, 0.732 and 0.828. These results demonstrate that the one-hot encoding utilized in DeepCleave provides the best solution for the prediction of the protease cleavage sites using this particular neural network topology, although several other types of inputs are also predictive. The deep network that uses all inputs secures relatively high average AUC = 0.865 and average MCC = 0.635. However, these results are still lower than the results secured by the one-hot encoding. Again, we believe that this stems from the architecture of the network that favours the binary inputs. One important advantage of the one-hot encoding is that it can be efficiently computed from the input protein sequence, particularly when compared to a computationally expensive calculation of some other inputs like PSSM, putative secondary structure and putative solvent accessibility. This leads to a short prediction runtime, allowing for a large-scale application of the DeepCleave method.

3.6 Webserver

The DeepCleave's webserver allows the users to perform high-throughput bioinformatics analyses of the protease specific cleavage sites. This server is freely available at http://deepcleave.erc.monash.edu/. The calculations are done on the server side, freeing the user from utilizing their own hardware. The website of the webserver also provides access to the trained DeepCleave's models. The front page was implemented using PHP and the webserver runs using Tomcat7 on the Linux system. The underlying hardware is an eight-core CPU, 500 GB hard disk and 16 GB memory, which ensures that predictions are produced efficiently.

To utilize the webserver, the users should paste the sequences of the proteins of interest into the 'TEXTAREA' or upload a protein sequence file that is formatted using the FASTA format. The website provides an example of correctly formatted inputs. The webserver allows for a batch submission of up to 100 sequences at a time, which is possible due to computational efficiency of the underlying predictive model. Users can download and run the trained models of DeepCleave using their own hardware to process larger protein sets. At the submission time, users can input an e-mail address to receive notification when the submitted task is completed. This email includes links to the web page with the predictions. Detailed step-by-step instructions of how to use the DeepCleave webserver are available on the help page of the webserver.

3.7 Case studies

We illustrate the results produced by DeepCleave using two substrate proteins selected from the independent test dataset, one that is cleaved by caspases and another that is cleaved by MMPs. The first protein is the human Claspin (UniProt ID: Q9HAW4) (Chini and Chen, 2003), while the second is Heat shock 70 kDa protein 4 from mouse (UniProt ID: Q61316) (McCallister *et al.*, 2015). We visualize the predictions in Supplementary Figure S5. There are three experimentally validated cleavage sites in Claspin that are cleaved by caspases, i.e. site 25 is cleaved by caspase-3, site 82 is cleaved by caspase-6 and site 1072 is cleaved by caspase-7 (Clarke *et al.*, 2005; Julien *et al.*, 2016; Semple *et al.*, 2007). DeepCleave is able to identify all three of these sites among the highest-valued predictions generated by the corresponding three models. The Heat shock 70 kDa protein 4 has five cleavage sites processed by MMPs, i.e. sites 97, 182, 356 and 678 are cleaved by MMP-2 and site 678 is cleaved by MMP-9 (auf dem Keller *et al.*, 2010; Prudova *et al.*, 2010). The MMP-2 and MMP-9 models from the DeepCleave server generate high scores for these positions, leading to accurate identification of these cleavage sites.

3.8 Human proteome-wide prediction of the substrate cleavage sites and gene ontology enrichment analysis

We apply DeepCleave to pre-compute a human proteome-wide prediction of protease substrate cleavage sites. To this end, we collect 20 413 human proteins from the Swiss-Prot database (The UniProt Consortium, 2017), 14/02/2019). We parameterize the outputs generated by DeepCleave for these proteins to obtain putative cleavage sites that are predicted with high-confidence, i.e. we use threshold

that corresponds to the 99% specificity on the training dataset (Li *et al.*, 2015, 2016, 2018b; Song *et al.*, 2018a, b). We provide summary of the predicted cleavage substrates and sites for the five caspases and seven MMPs in Supplementary Table S9. A complete list of the predicted cleavage substrates and their cleavage sites can be freely downloaded from the DeepCleave webserver page at http://deepcleave.erc.monash.edu/.

We perform functional analysis of these putative human substrates for the five caspases and seven MMPs. We generate a set of gene ontology (GO) terms that are significantly enriched for each set of the putative substrates when compared to the human proteome. We run two-sided hypergeometric tests to quantify significance of enrichment and we divide the significantly enriched terms (p-value $\leq$ 0.05) into three categories: cellular components, biological processes and molecular functions. The top five significantly over-represented GO terms for each protease are given in Supplementary Table S10. Summarized results in Supplementary Figure S6 demonstrate that putative substrates of different proteases are associated with different GO terms. However, putative substrates targeted by the same protease family tend to be enriched in more similar GO terms than when comparing GO terms across the families. For instance, the putative substrates targeted by caspase-1, caspase-3, caspase-6 and caspase-2 are enriched in the RNA-binding function (GO: 0003723) (Matthews *et al.*, 1994), which is supported by a close relationships between RNA-binding proteins and specific caspases (Janakiraman *et al.*, 2017; Subasic *et al.*, 2016; Talwar *et al.*, 2011). Moreover, the putative substrates of all caspases are enriched in the cytosol term (GO: 0005829), which is consistent with experimental studies of subcellular localization for caspases (Juin *et al.*, 1998; Mesner *et al.*, 1999). On the other hand, the putative substrates of all MMPs, except for MMP-2, are enriched in the 'extracellular region' (GO: 0005576) and 'extracellular space' (GO: 0005615) terms, which is again consistent with the actual subcellular localization for MMPs (Christensen and Shastri, 2015; Hakulinen *et al.*, 2008; Oh *et al.*, 2001; Schmidt-Hansen *et al.*, 2004; Wiesner *et al.*, 2013). These observations support validity of the underlying DeepCleave's predictions.

4 Conclusions

We introduce DeepCleave, the first deep learning-based approach for accurate prediction of the caspase and matrix metalloprotease substrate cleavage sites. DeepCleave employs substrate sequences as the sole input and utilizes the one-hot encoding to convert these sequences into the input for the deep network. We apply transfer learning to extend generic protease family models for the prediction of 12 specific proteases. This approach also allowed us to address the problem of a small sample sizes of the protease-specific cleavage site data.

We empirically demonstrate that the DeepCleave framework learns feature representations that accurately differentiate between different caspases and MMPs. We also show that the use of multiple kernel sizes and the attention layer lead to substantial improvements in the predictive performance of our method, and that the one-hot encoding provides favorable results when compared with several other input types. We empirically compare DeepCleave with several state-of-the-art predictors. The results reveal that DeepCleave provides accurate predictions that outperform previously proposed methods for the large majority of the considered proteases (9 out of 12). We anticipate that this deep learning framework will be useful for other similar predictive tasks, such as prediction of glycosylation and other PTM sites.

A user-friendly webserver and the source code of DeepCleave are freely available at http://deepcleave.erc.monash.edu/. This website also provides access to a pre-computed set of putative cleavage site for the entire human proteome. To sum up, DeepCleave is a computational tool for high-throughput and accurate cleavage site prediction, which has the potential to produce novel biological hypotheses.

Funding

This work was supported by grants from the Australian Research Council (ARC) (LP110200333 and DP120104460), National Health and Medical Research Council of Australia (NHMRC) (1092262, 490989), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965) and a Major Inter-Disciplinary Research (IDR) Grant Awarded by Monash University, and the Collaborative Research Program of Institute for Chemical Research, Kyoto University (2019-32). LK was supported in part by the Robert J. Mattauch Endowment funds.

Conflict of Interest: none declared.

References

Armenteros,J.A. (2019) Detecting Novel Sequence Signals in Targeting Peptides Using Deep Learning. *BioRxiv* 2019:639203, doi.org/10.1101/639203.

Auf Dem Keller,U. *et al.* (2010) A statistics-based platform for quantitative N-terminome analysis and identification of protease cleavage products. *Mol. Cell Proteomics*, **9**, 912–927.

Chen,Z. *et al.* (2018) iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, **34**, 2499–2502.

Chen,Z. *et al.* (2019) iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Brief. Bioinf.*, doi:10.1093/bib/bbz041.

Chini,C.C.S. and Chen,J. (2003) Human claspin is required for replication checkpoint control. *J. Biol. Chem.*, **278**, 30057–30062.

Christensen,J. and Shastri,V.P. (2015) Matrix-metalloproteinase-9 is cleaved and activated by Cathepsin K. *BMC Res. Notes*, **8**, 322.

Clarke,C.A. *et al.* (2005) Cleavage of claspin by caspase-7 during apoptosis inhibits the Chk1 pathway. *J. Biol. Chem.*, **280**, 35337–35345.

Elbasir,A. *et al.* (2018) DeepCrystal: a deep learning framework for sequence-based protein crystallization prediction. *Bioinformatics*, **35**, 2216–2225.

Fan,Y.X. *et al.* (2013) LabCaS: labeling calpain substrate cleavage sites from amino acid sequence using conditional random fields. *Proteins*, **81**, 622–634.

Faraggi,E. *et al.* (2014) Accurate single-sequence prediction of solvent accessible surface area using local and global features. *Proteins Struct. Funct. Bioinf.*, **82**, 3170–3176.

Fu,L. *et al.* (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**, 3150–3152.

Fu,S.C. *et al.* (2014) ScreenCap3: improving prediction of caspase-3 cleavage sites using experimentally verified noncleavage sites. *Proteomics*, **14**, 2042–2046.

Gulli,A. and Pal,S. (2017) *Deep Learning with Keras*. Packt Publishing Ltd. Birmingham, UK.

Hakulinen,J. *et al.* (2008) Secretion of active membrane type 1 matrix metalloproteinase (MMP-14) into extracellular space in microvesicular exosomes. *J. Cell. Biochem.*, **105**, 1211–1218.

Hilt,W. and Wolf,D.H. (1995) Proteasomes. Complex proteases lead to a new understanding of cellular regulation through proteolysis. *Naturwissenschaften*, **82**, 257–268.

Hurtado,D.M. *et al.* (2018) Deep transfer learning in the assessment of the quality of protein models. *arXiv preprint arXiv: 1804.06281*.

Janakiraman,H. *et al.* (2017) Repression of caspase-3 and RNA-binding protein HuR cleavage by cyclooxygenase-2 promotes drug resistance in oral squamous cell carcinoma. *Oncogene*, **36**, 3137–3148.

Jones,D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, **292**, 195–202.

Juin,P. *et al.* (1998) Induction of a caspase-3-like activity by calcium in normal cytosolic extracts triggers nuclear apoptosis in a cell-free system. *J. Biol. Chem.*, **273**, 17559–17564.

Julien,O. *et al.* (2016) Quantitative MS-based enzymology of caspases reveals distinct protein substrate specificities, hierarchies, and cellular roles. *Proc. Natl. Acad. Sci. USA*, **113**, E2001–2010.

Kingma,D.P. *et al.* (2014) A method for stochastic optimization. *arXiv preprint arXiv: 1412.6980*.

Krizhevsky,A. *et al.* (2017) ImageNet classification with deep convolutional neural networks. *Commun. ACM*, **60**, 84–90.

Kumar,S. *et al.* (2015) CleavPredict: a platform for reasoning about matrix metalloproteinases proteolytic events. *PLoS One*, **10**, e0127877.

LeCun,Y. *et al.* (2010) Convolutional networks and applications in vision. In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE. p. 253–256.

Li,F. *et al.* (2015) GlycoMine: a machine learning-based approach for predicting N-, C- and O-linked glycosylation in the human proteome. *Bioinformatics*, **31**, 1411–1419.

Li,F. *et al*. (2016) GlycoMine(struct): a new bioinformatics tool for highly accurate mapping of the human N-linked and O-linked glycoproteomes by incorporating structural features. *Sci. Rep*., **6**, 34595.

Li,F. *et al*. (2018a) Twenty years of bioinformatics research for protease-specific substrate and cleavage site prediction: a comprehensive revisit and benchmarking of existing methods. *Brief. Bioinf.*, doi: 10.1093/bib/bby077.

Li,F. *et al*. (2018b) Quokka: a comprehensive tool for rapid and accurate prediction of kinase family-specific phosphorylation sites in the human proteome. *Bioinformatics*, **34**, 4223–4231.

López-Otín,C. and Overall,C.M. (2002) Protease degradomics: a new challenge for proteomics. *Nat. Rev. Mol. Cell Biol*., **3**, 509–519.

Luo,F. *et al*. (2019) DeepPhos: prediction of protein phosphorylation sites with deep learning. *Bioinformatics*, **35**, 2766.

Matthews,D.A. *et al*. (1994) Structure of human rhinovirus 3C protease reveals a trypsin-like polypeptide fold, RNA-binding site, and means for cleaving precursor polyprotein. *Cell*, **77**, 761–771.

McCallister,C. *et al*. (2015) Functional diversification and specialization of cytosolic 70-kDa heat shock proteins. *Sci. Rep*., **5**, 9363.

McInnes,L. *et al*. (2018) Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv: 1802.03426*.

Mesner,P.W.,Jr *et al*. (1999) Characterization of caspase processing and activation in HL-60 cell cytosol under cell-free conditions. Nucleotide requirement and inhibitor profile. *J. Biol. Chem*., **274**, 22635–22645.

Oh,J. *et al*. (2001) The membrane-anchored MMP inhibitor RECK is a key regulator of extracellular matrix integrity and angiogenesis. *Cell*, **107**, 789–800.

Piippo,M. *et al*. (2010) Pripper: prediction of caspase cleavage sites from whole proteomes. *BMC Bioinformatics*, **11**, 320.

Prudova,A. *et al*. (2010) Multiplex N-terminome analysis of MMP-2 and MMP-9 substrate degradomes by iTRAQ-TAILS quantitative proteomics. *Mol. Cell Proteomics*, **9**, 894–911.

Rawlings,N.D. *et al*. (2018) The MEROPS database of proteolytic enzymes, their substrates and inhibitors in 2017 and a comparison with peptidases in the PANTHER database. *Nucleic Acids Res*., **46**, D624–D632.

Sainath,T.N. *et al*. (2013) Deep convolutional neural networks for LVCSR. In: Int Conf Acoust SPEE, pp. 8614–8618.

Schmidt-Hansen,B. *et al*. (2004) Extracellular S100A4(mts1) stimulates invasive growth of mouse endothelial cells and modulates MMP-13 matrix metalloproteinase activity. *Oncogene*, **23**, 5487–5495.

Semple,J.I. *et al*. (2007) Cleavage and degradation of Claspin during apoptosis by caspases and the proteasome. *Cell Death Differ*., **14**, 1433–1442.

Snoek,J. *et al*. (2012) Practical Bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Process. Syst*., **25**, 2960–2968.

Song,J. *et al*. (2010) Cascleave: towards more accurate prediction of caspase substrate cleavage sites. *Bioinformatics*, **26**, 752–760.

Song,J. *et al*. (2012) PROSPER: an integrated feature-based tool for predicting protease substrate cleavage sites. *PLoS One*, **7**, e50300.

Song,J. *et al*. (2018a) PROSPERous: high-throughput prediction of substrate cleavage sites for 90 proteases with improved accuracy. *Bioinformatics*, **34**, 684–687.

Song,J. *et al*. (2018b) iProt-Sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. *Brief. Bioinf*., **20**, 638–658.

Subasic,D. *et al*. (2016) Post-transcriptional control of executioner caspases by RNA-binding proteins. *Genes Dev*., **30**, 2213–2225.

Talwar,S. *et al*. (2011) Caspase-mediated cleavage of RNA-binding protein HuR regulates c-Myc protein expression after hypoxic stress. *J. Biol. Chem*., **286**, 32333–32343.

Team,T.T.D. *et al*. (2016) Theano: a Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv: 1605.02688*.

The UniProt Consortium (2017) UniProt: the universal protein knowledgebase. *Nucleic Acids Res*., **46**, 2699.

Umarov,R. *et al*. (2019) Promoter analysis and prediction in the human genome using sequence-based deep learning models. *Bioinformatics*, **35**, 2730.

van der Maaten,L. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res*., **9**, 2579–2605.

Wallace,B.C. *et al*. (2011) Class imbalance, redux. In: *2011 IEEE 11th international conference on data mining*. IEEE. pp. 754–763.

Wang,D. *et al*. (2017) MusiteDeep: a deep-learning framework for general and kinase-specific phosphorylation site prediction. *Bioinformatics*, **33**, 3909–3916.

Wang,D. *et al*. (2018) Capsule network for protein post-translational modification site prediction. *Bioinformatics*, **35**, 2386–2394.

Wang,M. *et al*. (2014) Cascleave 2.0, a new approach for predicting caspase and granzyme cleavage targets. *Bioinformatics*, **30**, 71–80.

Wiesner,C. *et al*. (2013) A specific subset of RabGTPases controls cell surface exposure of MT1-MMP, extracellular matrix degradation and three-dimensional invasion of macrophages. *J. Cell Sci*., **126**, 2820–2833.

Yao,Y. *et al*. (2007) On early stopping in gradient descent learning. *Constr. Approx*., **26**, 289–315.

Yosinski,J. *et al*. (2014) How transferable are features in deep neural networks? *Ad. Neural Inf. Process. Syst*., **27**, 3320–3328.

Zhang,F. *et al*. (2019) DeepFunc: a deep learning framework for accurate prediction of protein functions from protein sequences and interactions. *Proteomics*, **19**, 1900019.

Zhang,J. *et al*. (2017) Comprehensive review and empirical analysis of hallmarks of DNA-, RNA- and protein-binding residues in protein chains. *Brief. Bioinform*, doi.org/10.1093/bib/bbx168.

Zhang,T. *et al*. (2010) Analysis and prediction of RNA-binding residues using sequence, evolutionary conservation, and predicted secondary structure and solvent accessibility. *Curr. Protein Pept. Sci*., **11**, 609–628.