# Bidirectional Long Short-Term Memory Networks for Predicting the Subcellular Localization of Eukaryotic Proteins

Trias Thireou and Martin Reczko

**Abstract**—An algorithm called Bidirectional Long Short-Term Memory Networks (BLSTM) for processing sequential data is introduced. This supervised learning method trains a special recurrent neural network to use very long-range symmetric sequence context using a combination of nonlinear processing elements and linear feedback loops for storing long-range context. The algorithm is applied to the sequence-based prediction of protein localization and predicts 93.3 percent novel nonplant proteins and 88.4 percent novel plant proteins correctly, which is an improvement over feedforward and standard recurrent networks solving the same problem. The BLSTM system is available as a Web service at http://stepc.stepc.gr/~synaptic/blstm.html.

**Index Terms**—Recurrent neural networks, long short-term memory, biological sequence analysis, protein subcellular localization prediction.

## 1 Introduction

A standard problem in the processing of biological sequences is the detection and characterization of several unaligned sequence features with gaps of variable and partially very large size between them. One attractive solution is to process biosequences using recurrent neural networks (RNN) that can learn to store information using internal activation patterns [1] and have significantly fewer adaptable parameters than feedforward networks processing large subsequences. A major obstacle for the successful general application of RNNs is the "vanishing error" problem, whicht prevents the detection of relevant sequence features that occur in a distance of more than 10 sequence elements. To overcome this, Long Short-Term Memory (LSTM) networks have been developed [2]. The basic idea is to use a self-connected linear unit as a memory cell, the so-called "Constant Error Carousel," both for storing information and for propagating error information over arbitrarily long distances in the sequences. For context dependent information storage and recall, the input and output of each memory cell can be opened and closed using a multiplicative connection to gating units. It has been shown that LSTM networks can even learn context-sensitive languages, where traditional RNN fail [2]. Here, we describe the application of this algorithm for the sequence-based prediction of the localization of eukaryotic proteins.

Newly synthesized proteins are posttranslationally sorted and transported from the cytosol to different subcellular compartments by a highly optimized machinery [3]. Knowledge about the subcellular location of a protein indicates potential functions of a protein [4] and is a very valuable annotation to filter large amounts of protein sequences for which a precise functional annotation is not available. Other uses of these predictions are testing the localization of designed proteins, the large scale screening of proteomes for proteins with desired localization as targets for drug design and for protein identification in measurements of various analytical methods for proteomics. An example in protein 2D electrophoresis is the case where several spots can be encoded by the same gene with different posttranslational modifications, one of which is subcellular localization.

The most commonly used methods for predicting subcellular localization are reviewed in [5], [6], [7]. The usual distinction between these methods defines three classes: detection of targeting signals [8], [9], [10], [11], [12], detection of different amino acid compositions [13], [14], [15], and the use of evolutionary relationships between proteins targeted to the same compartment [16], [17]. Some approaches combine the use of signals and sequence composition explicitly [18], [19], [20] or implicitly [1], [21], [22]. Like these implicit methods, the LSTM neural network introduced here also reads the sequence with a window of a single amino acid and accumulates information about sequence signals and composition using activation patterns stored in feedback loops. Due to the long-range error propagation using linear feedback, this context information can be more accurate when using traditional recurrent networks [23].

• *The authors are with the Bioinformatics Programme, Biomedical Informatics Laboratory, Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH), PO Box 1385, 711 10 Heraklion, Crete, Greece. E-mail: reczko@ics.forth.gr.*

*Manuscript received 29 Sept. 2005; revised 25 Apr. 2006; accepted 1 Aug. 2006; published online 10 Jan. 2007.*
*For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0109-0905.*
*Digital Object Identifier no. 10.1109/TCBB.2007.1015.*

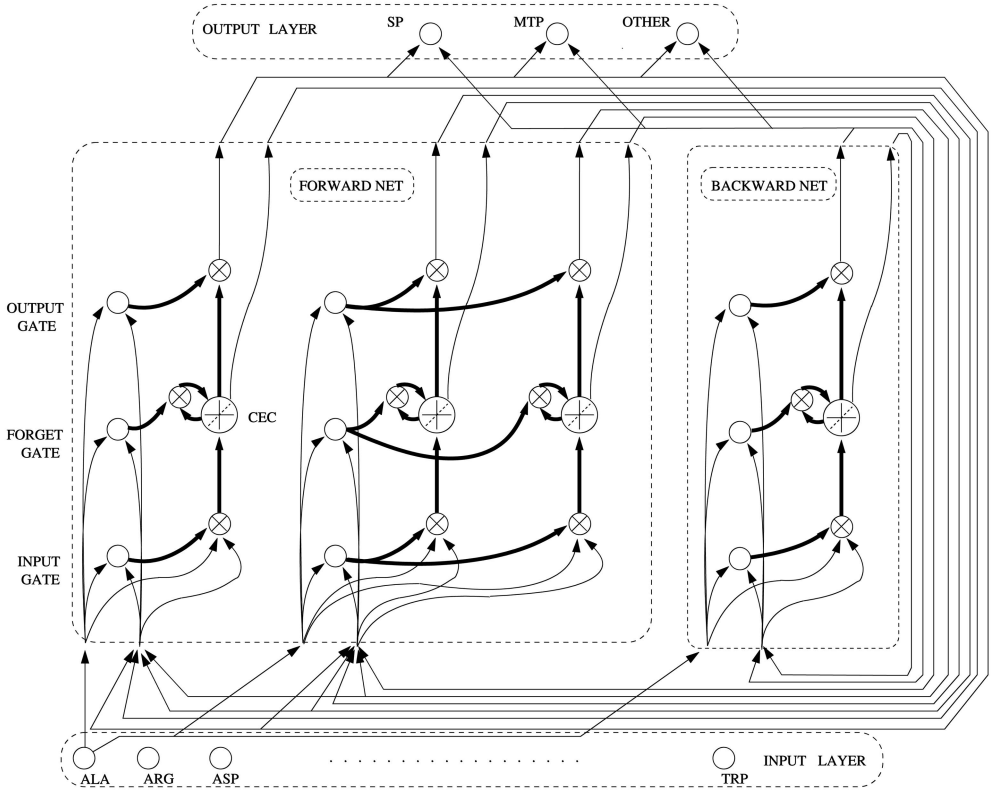


Fig. 1. The Bidirectional Long Short-Term Memory network architecture for the nonplant proteins. The forward reading net contained three memory blocks with one, two, and two memory cells, respectively, while the backward reading net had just one memory block with a single memory cell. For clarity, the third memory block in the forward reading net is not shown. The circles with crosses are multiplicative junctions where all inputs from below the junction are summed up and the sum is multiplied with the value connecting from the left of the junction. The open circles are standard neurons with a $tanh$ nonlinear transfer function. The constant-error-carousel (CEC) is shown as a unit with the identity transfer function feeding its activity into itself via the multiplicative junction controlled by the forget gate. The connections shown with bold lines have a fixed constant weight of 1.0. The architecture for the plant proteins is identical with one additional output neuron for the "cTP" class.

## 2 System and Methods

### 2.1 Data

For training, validation and testing of the networks the data available from the TargetP Web site (http://www.cbs.dtu.dk/services/TargetP) was used. The nonplant data set consisted of 371 mitochondrial targeting peptides (mTP), 715 signal peptides (SP), 1,214 nuclear, and 438 cytosolic sequences combined into the "other" class and the plant data set consisted of 141 chloroplast targeting peptides (cTP), 368 mitochondrial targeting peptides (mTP), 269 signal peptides (SP), and 162 sequences for the "other" class.

Each set of sequences $ALLSEQS_{class}$ for the three nonplant classes and the four plant classes was shuffled and then split into five equally sized disjunct subsets $CLASSSUBSET_{class,set}$ $(set = 1 \ldots 5)$. The five subsets of each class contributed to one of the five partitions $PARTITION_{set}$. A partition $PARTITION_{set}$ contained all of the subsets $CLASSSUBSET_{class,set}$ for the three or four classes.

For the fivefold cross-validation, five different networks were trained using a training set that is the combination of three different partitions

$$TRA_{xval} = PARTITION_{set_a} + PARTITION_{set_b} + PARTITION_{set_c}.$$

Of the remaining two partitions, one was used as a validation set, $VAL_{xval} = PARTITION_{set_d}$, for early stopping of the learning algorithm, optimization of the network architecture, and postprocessing of the network outputs, while the other was used only for the single final test set $TES_{xval} = PARTITION_{set_e}$ for which predictive performance was measured. The choices for the different permutations were restricted such that the five tests are different and the validation sets are different.

## 3 Algorithm

### 3.1 Bidirectional Long Short-Term Memory Networks

A detailed description of the original long short-term memory (LSTM) algorithm can be found in [24]. In LSTM networks, all weights were adapted using a simplified "Real Time Recurrent Learning" (RTRL) [25] algorithm having a computational complexity of $O(1)$ per time step and per weight. For the processing of biosequences, we have modified and extended this algorithm in the following ways:

- The weight changes were accumulated over the presentation of all training patterns (batch instead of online update).
- The update procedure used the resilient backpropagation (RPROP) algorithm [26]. This method

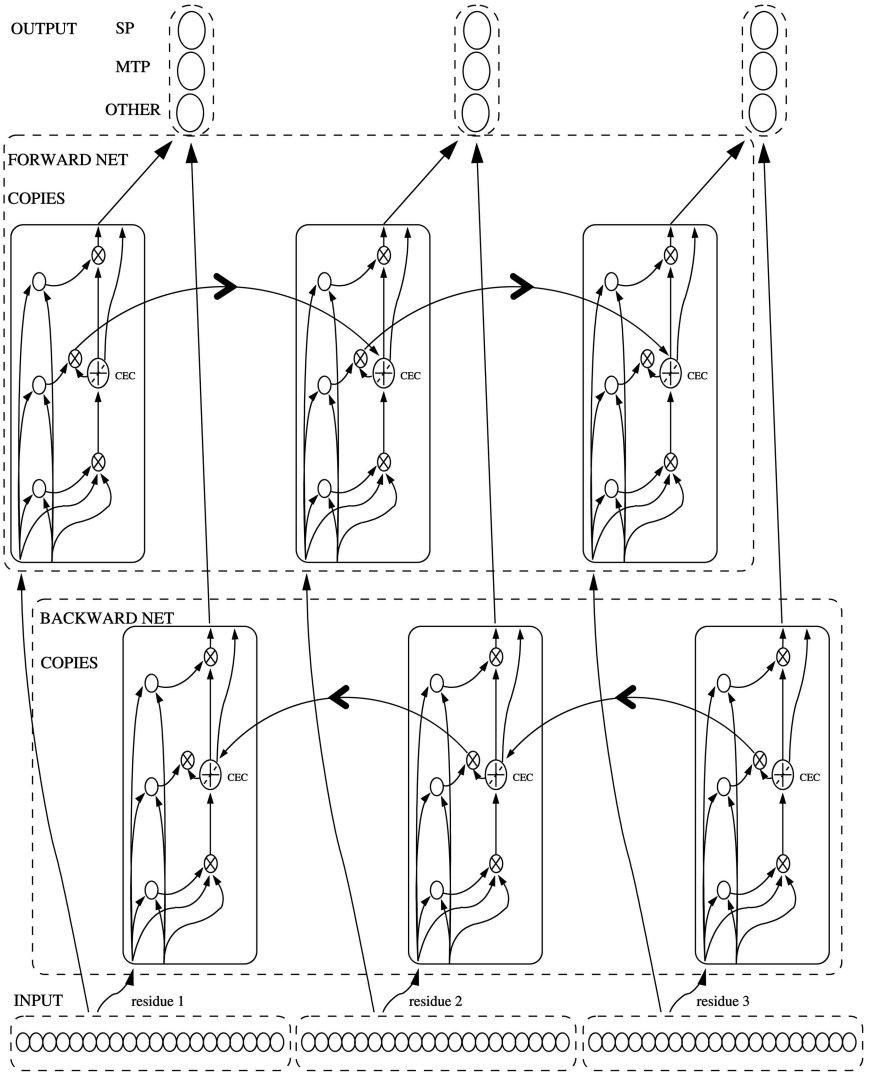


Fig. 2. A BLSTM with one memory cell in the forward and backward network is unfolded for a sequence with three elements. For clarity, the only recurrent connections shown and unfolded through time are of the constant-error-carousels (CEC).

implements an individual learning rate for each weight and is commonly used to speed up learning [21], [27].

- The network contained two subnetworks for the left and right sequence context. The outputs of these subnets are integrated in the output layer in the same way as bidirectional recurrent neural networks.

### 3.2 Batch Weight Update and RPROP

The original LSTM algorithm uses online weight update after the presentation of each input/output pair in a sequence of patterns. The bidirectional extensions to the LSTM algorithm require a weight update procedure that accumulates the weight changes over all elements of all sequences of the training set. An efficient weight update procedure of this type is the resilient backpropagation (RPROP) algorithm [26], which was incorporated into the LSTM analogous to the description in [21].

### 3.3 Context and the Bidirectional LSTM

To store both the context to the left and the right for any position in a sequence, the BLSTM architecture used two separate networks, one forward network reading the input sequence from left to right and one backward network reading the sequence from right to left (see Fig. 1). The

TABLE 1
The Parameters of the Networks for Optimal Performance on the Corresponding Validation Sets

| net | plant | | non - plant | |
|---|---|---|---|---|
| | thres-hold | deriv. win | thres-hold | deriv. win |
| 1 | 0.32 | 33 | 0.13 | 31 |
| 2 | 0.25 | 27 | 0.39 | 11 |
| 3 | 0.20 | 23 | 0.23 | 29 |
| 4 | 0.37 | 13 | 0.20 | 33 |
| 5 | 0.36 | 19 | 0.21 | 33 |

*The column $threshold$ contains the value that $avg.class.act^{class}$ has to exceed for classification into classes different from "other." The column $deriv.win$ defines the number of residues used to calculate the average derivative of the networks output.*

TABLE 2
Comparison of Feedforward (FFNN) [11], Bidirectional Recurrent (BRNN) [21], and Bidirectional Long Short-Term Memory (BLSTM) Neural Network Prediction Performances on Novel Nonplant Test Sets in Fivefold Cross-Validation

| Algorithm | True category | No. of sequences | Predicted category: mTP | SP | other | Sensitivity | Mathews correlation |
|---|---|---|---|---|---|---|---|
| BRNN (FFNN) | mTP | 371 | 290 (330) | 23 (9) | 58 (32) | 0.78 (0.89) | 0.77 (0.73) |
| | SP | 715 | 2 (13) | 668 (683) | 45 (19) | 0.93 (0.96) | 0.89 (0.92) |
| | other | 1652 | 63 (152) | 47 (49) | 1542 (1451) | 0.93 (0.88) | 0.84 (0.82) |
| | Specificity | | 0.82 (0.67) | 0.91 (0.92) | 0.94 (0.97) | | |
| BLSTM | mTP | 371 | 300 | 9 | 62 | 0.81 | 0.81 |
| | SP | 715 | 6 | 688 | 21 | 0.96 | 0.93 |
| | other | 1652 | 43 | 43 | 1566 | 0.95 | 0.87 |
| | Specificity | | 0.86 | 0.93 | 0.95 | | |

For non-plant sequences, $90.0 \pm 1.1$ of the FFNN predictions, $91.3 \pm 0.99\%$ of the BRNN predictions and $93.3 \pm 0.6\%$ of the BLSTM predictions are correct, where the standard deviations are for the five different networks and test sets used for cross-validation.

forward net accumulates any sequence context to the left of each position in the sequence and the backward net accumulates sequence context to the right of each position. In the case of processing protein sequences, the left and right context corresponds to information available from the amino and carboxy terminus of the protein, respectively. By storing the activities of both networks while reading the sequence, copies of the networks are available at each position of the sequence. After a sequence has been processed in both directions, the output activities of the separate networks were used to compute the final output activities using the weights to the output neurons. During training, any error information was propagated backward through the copies of the networks and the weight changes are accumulated. The unfolding of a BLSTM network for a hypothetical protein sequence with three residues is illustrated in Fig. 2.

### 3.4 Data Representation

All sequences were truncated after the first 70 N-terminal residues, which was found to be the length with the optimal performance on the validation set among the alternative lengths 60, 65, 70, 75, 90, and 110. A protein was processed by the BLSTM network by sequentially coding the residues in the sequence into the input layer one-by-one using the standard one-out-of-20 code. The three and four different localization classes were represented using three output neurons having normalized exponentials [28] as activation functions (the so-called "soft-max" output).

The target values for the output units along the sequence had an activation of 1.0 for one of the classes mTP, cTP, or SP, if the residue in the sequence was part of the peptide indicating the corresponding localization and ranging from the N-terminal to the cleavage site of that peptide. In case the sequence belonged to the "other" class, the target of the neuron for this class was 1.0 over the whole sequence.

To assign a localization class to a test or validation sequence, the corresponding output neuron obviously had to show a sequence of high activities on the N-terminal, followed by a region of lower activities. To quantify this criterion, we first calculated an average derivative $edge_i^{class}$ of the output activities $o_i^{class}$ at each position $i = 1, \ldots, 70$ of the sequence by summing the output activities in a window of $deriv.win$ residues to the right of position $i$ and subtracting the sum of the output activities in a window of the same size to the left of that position. Formally,

$$edge_i^{class} = \frac{(\sum_{j=i}^{i+deriv.win} o_j^{class} - \sum_{j=i-deriv.win-1}^{i-1} o_j^{class})}{deriv.win.}. \quad (1)$$

For each class $class$, the position $max.edge.pos^{class}$ of the maximum of $edge_i^{class}$ was calculated,

$$max.edge.pos^{class} = \underset{i=1,\ldots,70}{argmax} \; edge_i^{class} \quad (2)$$

and the average activity $avg.class.act^{class}$ from the N-terminal to that position was the score for each class:

$$avg.class.act^{class} = \frac{\sum_{j=1}^{max.edge.pos^{class}} o_j^{class}}{max.edge.pos^{class}}. \quad (3)$$

If the maximum of $avg.class.act^{class}$ over all classes exceeded a fixed $threshold$, the sequence was classified as belonging to the class corresponding to the maximum. Otherwise, it was classified into the "other" class.

The values for $deriv.win.$, $threshold$, the parameters for the architecture and the learning algorithm, were varied systematically to give the optimal Mathews correlation (Math.corr.) on the validation set.[1]

This validation performance was monitored during the 7,000 epochs of training and the best network was saved. The point of best validation performance typically occurred after 4,000 epochs. The best results were obtained with three memory blocks having one, two, and two memory cells, respectively, in the forward network and two memory blocks with one memory cell each in the backward network. The other optimal parameters for the five networks are shown in Table 1.

1. The maximum size of the initial random weights was varied between 0.1 and 0.9 with a step size of 0.1, the learning rate between 0.01 and 0.09 with a step size of 0.01, $threshold$ between 0.01 and 0.99 with a step size of 0.01, and $deriv.win.$ between 5 and 25 with a step size of 2.

TABLE 3
Comparison of FFNN [11] and BLSTM Prediction Performance on Novel Plant Test Sets in Fivefold Cross-Validation

| True category | No. of sequences | Predicted category cTP | mTP | SP | other | Sensitivity | Math.corr. |
|---|---|---|---|---|---|---|---|
| cTP | 141 | 109 (120) | 16 (14) | 2 (2) | 14 (5) | 0.77 (0.85) | 0.74 (0.72) |
| mTP | 368 | 21 (41) | 326 (300) | 5 (9) | 18 (18) | 0.89 (0.82) | 0.84 (0.77) |
| SP | 269 | 4 (2) | 3 (7) | 254 (245) | 8 (15) | 0.94 (0.91) | 0.94 (0.90) |
| other | 162 | 8 (10) | 9 (13) | 3 (2) | 142 (137) | 0.88 (0.85) | 0.79 (0.77) |
| Specificity | | 0.78 (0.69) | 0.92 (0.90) | 0.96 (0.96) | 0.78 (0.78) | | |

In total $88.4 \pm 2.6\%$ $(85.3 \pm 3.5\%)$ correct predictions for the plant predictor, where the standard deviations refer to the spread in performance of the five different networks and test sets used for cross-validation.

*The FFNN results are given in brackets.*

## 4 Results and Discussion

The crossvalidated classification results on the nonplant test sets, $TES_1 \ldots TES_5$, for all five independent networks are compared in Table 2 with the performance of a feedforward network (TargetP) [11] and a bidirectional recurrent network (BRNN) [21].[2] These comparisons with BRNN, obtained on the same data sets, show that there is a contribution of detecting and using long range dependencies using the BLSTM algorithm. The Mathews correlation averaged over the three classes avoids the influence of unevenly distributed classes and also shows an improvement from 0.83 for BRNN to 0.87 for BLSTM. To assess the influence of the different ways in which the average activations are calculated for the BRNN and the BLSTM, we also evaluated the performance of the BLSTM networks using the average activation up to a fixed position, as was done for the BRNN networks. The results on the nonplant set show 93.1 $\pm$ 0.3 percent correct classifications in this case and still show an improvement over BRNN having $91.3 \pm 0.99$ percent correct classifications. The results on the plant test sets are compared in Table 3 with the performance of a feedforward network (TargetP), again demonstrating the improvement.

To assess the "receiver-operating-characteristic" (ROC) of the classifiers, sequences with a value for $max.edge^{class}$ below a threshold for the classes SP, mTP, and cTP can be classified into the "other" class. In Figs. 3 and 4, the tradeoff between sensitivity and specificity is shown averaged over the five test sets for both plant and nonplant proteins.[3] The labeled cut-points for the output activity thresholds are useful to interpret the network outputs for desired levels of sensitivity or specificity. The increasing difficulty for predicting the classes SP, mTP, and cTP is also clearly visible in the ROC curves.

The number of parameters adjusted by the BLSTM learning algorithm is around 300. In other feedforward neural network approaches to this problem, there are around 5,000 adjustable parameters. According to the principle of minimum description length, better generalization can be expected. The improved performance of the BLSTM networks applied to the same data and under the same conditions as a BRNN [21] indicates the advantage of considering and using long range dependencies for this problem. As the detection of long-range context information is required in many bioinformatics problems, there are many

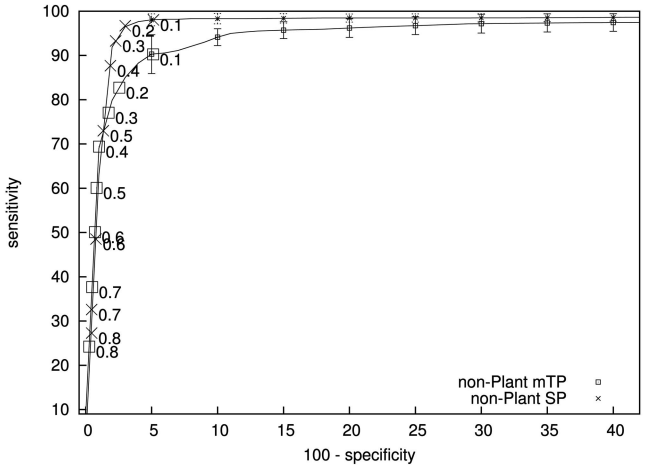


Fig. 3. Average ROC curves for nonplant proteins from the test sets. The errorbars indicate the standard deviations of the sensitivity over the five test sets. The additional lables indicate the threshold cut-points.

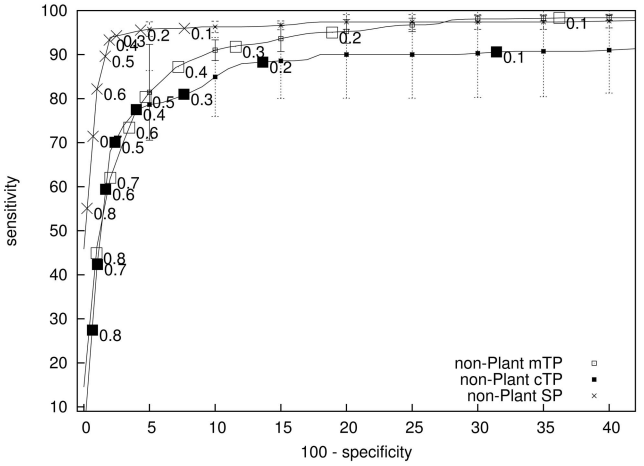


Fig. 4. Average ROC curves for plant proteins from the test sets. See the explanation in Fig. 3.

2. As used in the cited publications, we define sensitivity as ("correct positives")/("correct positives" + "false negatives") and specificity as ("correct positives")/("correct positives" + "false positives").

3. Note that the alternative definition of specificity as ("correct negatives")/("correct negatives" + "false positives") is commonly used for ROC curves and also used here. This definition ensures the monotonicity of the ROC curve.

potential applications for the LSTM and BLSTM algorithms. We are currently investigating the use of BLSTM networks for the sequence-based prediction of protein and RNA secondary structure, surface exposure, disulfide bonding connectivity, dihedral angles, and contact maps.

## ACKNOWLEDGMENTS

T. Thireou acknowledges support by the State Fellowships Foundation of Greece (IKY) for postdoctoral research in Bioinformatics. The authors also thank the anonymous reviewers for their comments and suggestions.

## REFERENCES

[1] M. Reczko, E. Staub, P. Fiziev, and A. Hatzigeorgiou, "Finding Signal Peptides in Human Protein Sequences Using Recurrent Neural Networks," *Lecture Notes in Computer Science,* R. Guigo and D. Gusfield, eds., vol. 2452, pp. 60-67, 2002.

[2] F. Gers and J. Schmidhuber, "LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages," *IEEE Trans. Neural Networks,* vol. 12, no. 6, pp. 1333-1340, 2001.

[3] G. Schatz and B. Dobberstein, "Common Principles of Protein Translocation across Membranes," *Science,* vol. 271, no. 5255, pp. 1519-1526, 1996.

[4] B. Eisenhaber and P. Bork, "Wanted: Subcellular Localization of Proteins Based on Sequence," *Trends Cell Biology,* vol. 9, pp. 169-170, 1998.

[5] O. Emanuelsson and G. von Heijne, "Predicting of Organellar Targeting Signals," *Biochimica et Biophysica Acta,* vol. 1541, pp. 114-119, 2001.

[6] K. Nakai, "Review: Prediction of in vivo Fates of Proteins in the Era of Genomics and Proteomics," *J. Structural Biology,* vol. 134, pp. 103-116, 2001.

[7] K. Nakai, "Protein Sorting Signals and Prediction of Subcellular Localization," *Advances in Protein Chemistry,* vol. 54, pp. 277-344, 2000.

[8] H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne, "Identification of Prokaryotic and Eukaryotic Signal Peptides and Prediction of Their Cleavage Sites," *Protein Eng.,* vol. 10, no. 1, pp. 1-6, 1997.

[9] H. Nielsen, S. Brunak, and G. von Heijne, "Machine Learning Approaches for the Prediction of Signal Peptides and Other Protein Sorting Signals," *Protein Eng.,* vol. 12, no. 1, pp. 3-9, 1999.

[10] M.G. Claros and P. Vincens, "Computational Method to Predict Mitochondrially Imported Proteins and Their Targeting Sequences," *European J. Biochemistry,* vol. 241, pp. 779-786, 1996.

[11] O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne, "Predicting Subcellular Localization of Proteins Based on Their N-Terminal Amino Acid Sequence," *J. Molecular Biology,* vol. 300, pp. 1005-1016, 2000.

[12] B. Jagla and J. Schuchhardt, "Adaptive Encoding Neural Networks for the Recognition of Human Signal Peptide Cleavage Sites," *Bioinformatics,* vol. 16, pp. 245-250, 2000.

[13] A. Reinhardt and T. Hubbard, "Using Neural Networks for Prediction of the Subcellular Location of Proteins," *Nucleic Acids Research,* vol. 26, no. 9, pp. 2230-2236, 1998.

[14] K.C. Chou, "Using Subsite Coupling to Predict Signal Peptides," *Protein Eng.,* vol. 14, pp. 75-79, 2001.

[15] S. Hua and Z. Sun, "Support Vector Machine Approach for Protein Subcellular Localization Prediction," *Bioinformatics,* vol. 17, no. 8, pp. 721-728, 2001.

[16] E.M. Marcotte, I. Xenarios, A.M. van der Bliek, and D. Eisenberg, "Localizing Proteins in the Cell from Their Phylogenetic Profiles," *Proc. Nat'l Academy of Sciences USA,* vol. 97, no. 22, pp. 12115-12120, 2000.

[17] R. Mott, J. Schultz, P. Bork, and C.P. Ponting, "Predicting Protein Cellular Localization Using a Domain Projection Method," *Genome Reserach,* vol. 12, pp. 1168-1174, 2002.

[18] H. Bannai, Y. Tamada, O. Maruyama, K. Nakai, and S. Miyano, "Extensive Feature Detection of n-Terminal Protein Sorting Signals," *Bioinformatics,* vol. 18, no. 2, pp. 298-305, 2002.

[19] A. Drawid and M. Gerstein, "A Bayesian System Integrating Expression Data with Sequence Patterns for Localizing Proteins: Comprehensive Application to the Yeast Genome," *J. Molecular Biology,* vol. 301, pp. 1059-1075, 2000.

[20] M. Bhasin and G. Raghava, "ESLpred: SVM-Based Method for Subcellular Localization of Eukaryotic Proteins Using Dipeptide Composition and PSI-BLAST," *Nucleic Acids Research,* vol. 32, pp. W414-W419, 2004.

[21] M. Reczko and A. Hatzigeorgiou, "Prediction of Subcellular Localization of Eukaryotic Proteins Using Sequence Signals and Composition," *PROTEOMICS,* vol. 4, no. 6, pp. 1591-1596, 2004.

[22] J. Hawkins and M. Boden, "The Applicability of Recurrent Neural Networks for Biological Sequence Analysis," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 2, no. 3, pp. 243-253, July-Sept. 2005.

[23] F. Gers et al., "Learning Precise Timing with LSTM Recurrent Networks," *J. Machine Learning Research,* vol. 3, pp. 115-143, 2002.

[24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[25] A.J. Robinson and F. Fallside, "The Utility Driven Dynamic Error Propagation Network," Technical Report CUED/F-INFENG/TR.1, Eng. Dept., Cambridge Univ., 1987.

[26] M. Riedmiller and H. Braun, "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm," *Proc. IEEE Int'l Conf. Neural Networks (ICNN '93),* H. Ruspini, ed., pp. 586-591, 1993.

[27] M. Schuster and K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Trans. Signal Processing,* vol. 45, pp. 2673-2681, 1997.

[28] P. Baldi, S. Brunak, Y. Chauvin, C.A.F. Andersen, and H. Nielsen, "Assessing the Accuracy of Prediction Algorithms for Classification: An Overview," *Bioinformatics,* vol. 16, pp. 412-424, 2000.

**Trias Thireou** received the PhD degree in biomedical engineering from the National Technical University of Athens (NTUA), Greece, in collaboration with the German Cancer Research Center in Heidelberg, in 2002, concerning the iterative image reconstruction and analysis of dynamic Positron Emission Tomography studies. She was a research assistant at the Institute of Communication and Computer Systems (ICCS-NTUA) until 2003, participating in numerous European and Greek Research Projects. She is currently a postdoctoral fellow in bioinformatics at the Institute of Computer Science (ICS) and the Institute of Molecular Biology and Biotechnology (IMBB) of the Foundation for Research and Technology-Hellas (FORTH). Her research interests include applications of machine learning methods and data mining in bioinformatics, computational protein design, Monte Carlo simulations, and biomedical image analysis.

**Martin Reczko** received the PhD degree in computer science from the University of Stuttgart in 1995 conducting a collaborative project organized by him with the German Cancer Research Center in Heidelberg concerning the development of artificial neural networks for modeling systems in molecular biology. Since then, he has been the managing director of Synaptic Ltd., a company he cofounded which focuses on the development and application of artificial neural networks and evolutionary methods in computational molecular biology. From 1998 to 2001, he held a postdoctoral position within a European Training and Mobility of Researchers (TMR) network project concerned with advanced signal processing for magnetic resonance imaging at the Democritus University of Thrace, Xanthi. Since 2002, he has been a principal researcher in bioinformatics activity at the Institute of Computer Science (ICS) and the Institute of Molecular Biology and Biotechnology (IMBB) of the Foundation for Research and Technology-Hellas (FORTH). His research interest is the development of novel computational methods for analyzing complex biological problems. On this subject, he has published several journal papers and has contributed to national and European R&D projects.