Prediction of Debacle Points for Robustness of Biological Pathways by Using Recurrent Neural Networks

Hironori Kitakaze[1,2] kitakaze@oshima-k.ac.jp
Hiroshi Matsuno[3] matsuno@sci.yamaguchi-u.ac.jp
Nobuhiko Ikeda[4] n-ikeda@tokuyama.ac.jp
Satoru Miyano[5] miyano@ims.u-tokyo.ac.jp

[1] Oshima College of Maritime Technology, 1091-1 Oshima-cho, Oshima-gun, Yamaguchi 742-2193, Japan
[2] Graduate School of Science and Engineering, Yamaguchi University, 1677-1 Yoshida, Yamaguchi-shi, Yamaguchi 753-8512, Japan
[3] Faculty of Science, Yamaguchi University, 1677-1 Yoshida, Yamaguchi-shi, Yamaguchi 753-8512, Japan
[4] Tokuyama College of Technology, 3538 Takajo, Kume, Shunan-shi, Yamaguchi 745-8585, Japan
[5] Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan

Abstract

Living organisms have ingenious control mechanisms in which many molecular interactions work for keeping their normal activities against disturbances inside and outside of them. However, at the same time, the control mechanism has debacle points at which the stability can be broken easily. This paper proposes a new method which uses recurrent neural network for predicting debacle points in an hybrid functional Petri net model of a biological pathway. Evaluation on an apoptosis signaling pathway indicates that the rates of 96.5 % of debacle points and 65.5 % of non-debacle points can be predicted by the proposed method.

Keywords: hybrid functional Petri net, Genomic Object Net, recurrent neural networks, back propagation through time

1 Introduction

Living organisms have ingenious control mechanisms in which many molecular interactions work in order to keep their normal activities against disturbances from inside and outside of living organisms. Bioinformatics researchers have constructed computational models of biological phenomena with which many results about robustness have been obtained [4]. However, there have been few researches on finding debacle points for robustness. A debacle point is the point at which homeostasis of a biological control system is broken down by some key molecules or some unusual conditions in a biological control system.

In Genomic Object Net (GON) Project [3, 6, 12], we have modeled many biological pathways including metabolic pathways, signaling pathways, and gene regulatory mechanisms with Cell Illustrator [11] which is a software for biopathway simulator developed on the basis of hybrid functional Petri net (HFPN). An HFPN is constructed with three types of elements, places, transitions, and arcs [5]. A real number in a place is controlled by the speed assigned at a transition which is attached to the place by an arc.

Debacle points can be predicted by repeating simulations, comparing contents in places from the normal HFPN model with contents in places from an incomplete HFPN model whose some arcs were removed in advance[1]. However, a huge amount of time will be necessary with this method because many simulations have to be conducted until satisfactory predictions for debacle points are obtained. In addition, since this method is rather subjective, it is difficult to make appropriate choices to pick up arcs which will affect the robustness.

This paper proposes a new method which uses recurrent neural network (RNN) [9] for predicting debacle points in an HFPN model of a biological pathway. An RNN is constructed as it keeps connections in an HFPN of a biological pathways, performing learning processes by the back propagation through time (BPTT) method so that time course data in places are kept for identifying transnormal parts and irregular time course behavior of the biological pathway by further disruption processes of molecular interactions.

This paper gives an RNN based method which can predict debacle points in biological pathways in much less time than that conventional method by comparison between two types of simulation results from normal and incomplete HFPN models. In addition, with this method, we can realize a fully automatic process for predicting debacle points by employing the two features of Cell Illustrator, the XML description of HFPN and the mechanism exporting numerical time course data in places.

Evaluation of the proposed method has performed on the HFPN model for apoptosis signaling pathway [1, 5]. With the proposed RNN based method, debacle points of this pathway could be predicted in one hour, while over eighteen hours were spent by manual procedures of the same works. For larger pathway models, the differences in these times between the proposed method and the manual procedure will become larger. The rates of 96.5 % of debacle points and 65.5 % of non-debacle points can be successfully predicted by the proposed method. Although the results of this experimentation include some false positives and false negatives, the proposed method works well in finding debacle points in a biological pathway.

2 Hybrid Functional Petri Net and Recurrent Neural Network

2.1 Hybrid Functional Petri Net

Hybrid Petri net (HPN) [2] has been introduced as an extension of the discrete Petri net [7]. In HPN, two kinds of places and transitions are used, discrete/continuous places and discrete/continuous transitions. A discrete place and a discrete transition are the same notions as used in the discrete Petri net. A continuous place holds a nonnegative real number as its content. A continuous transition fires continuously in the HPN and its firing speed is given as a function of values in the places in the model. The HFPN [5] was introduced by expanding the notion of HPN, for example, any functions can be assigned to arc/transition for controlling the speed/condition of consumption/production/firing. These expansions allow us to model various aspects in biopathways very smoothly [5, 12].

2.2 Recurrent Neural Network

Recurrent neural networks (RNNs) [9] which can be considered as generalized hierarchical networks in terms of time-space, is feed forward hierarchical networks with feedback mechanisms, enabling us to process time-space information naturally. Connections between units of RNNs are generalized asymmetrically. Hence, RNNs do not have definite concepts of layers as hierarchical networks: units of a network can be classified into input units, output units, and hidden units. Inputs units are connected to all output units and hidden units, and output units and hidden units are connected with each other.

The most simple learning method of RNN is to apply the error back propagation (BP) [8] to hierarchical networks which are extracted from non-hierarchical networks with feedback loops. This

[1] Process to remove an arc in an HFPN represent actions to disrupt the corresponding interaction in the original biological pathway.

method is called the back propagation through time (BPTT). In the BPTT method, weights of connections in an RNN will be updated based on previous inputs, outputs and weights of them which have been kept through the start time t_0 to the time t in calculating dynamics of the network. Learning of RNNs in time-space are performed with repeating such update processes on weights of connections.

3 A Method to Predict Debacle Points in a Biological Pathway

By repeating the action to remove a different arc in an HFPN model for a biological pathway, debacle points in the biological pathway can be predicted based on differences in behaviors of place contents from simulations on these two types HFPN. However, huge amount of time is required by this method because each simulation on an HFPN spends much time. In contrast, with an RNN based method proposed in this paper, we can predict these debacle points in much less time.

In addition, we can realize a fully automatic process for debacle points prediction: our method can automatically utilize parameters for prediction processes by extracting necessary data for predictions from descriptions of a biological pathway in XML file. The feature of Cell Illustrator in employing XML format for HFPN description enables this full automatic process.

The proposed method is constituted by the following three stages.

(1st) Reconstruct an RNN from an HFPN with learning.

(2nd) Remove all connections in the RNN corresponding to an arc of the HFPN and extract differences between its unsteady state of the HFPN model and its steady state of this HFPN model. Repeat this procedure in sequence for each arc of the HFPN.

(3rd) Predict debacle points by sorting the differences in ascending order.

3.1 1st Stage: RNN Construction and Its Time Expansion to BPTT

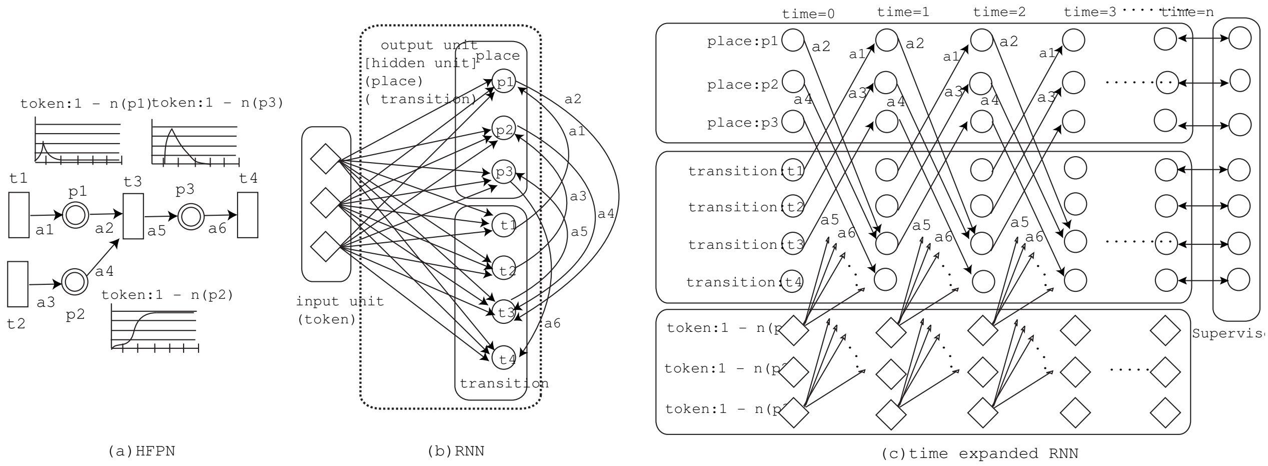

Figure 1: 1st stage: Construction of RTT and its time expansion to BPTT.

The following is a processes to construct the RNN of Figure 1 (b) from the HFPN of Figure 1(a).

- Extract all time course data of token (concentration behavior) from all places of an HFPN. These time course data are incorporated in input units of RNN.

Table 1: Input signals and teaching signals.

	input signals	teaching signals
Pattern 1	all contents of places	all 0
Pattern 2	all 0	all 1

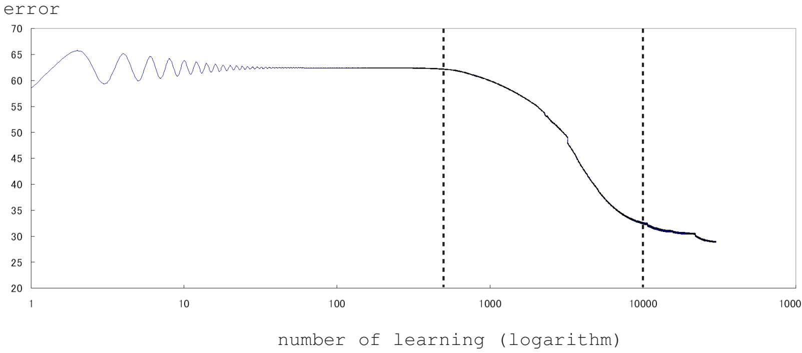

Figure 2: General transition of error differences in learning process by BPTT method.

- Places and transitions of an HFPN are defined as hidden units (including output units), where the number of units depends on the number of places and the number of transitions of an HFPN.
- A connection in an RNN is made according to each connection of arcs from places/transitions to a transitions/places in an HFPN.

Figure 1 (c) shows the RNN expanded on time (time expanded RNN) from the RNN of Figure 1(b) with which the learning by BPTT is going to be performed. The hierarchical structure of an RNN do not change when time for simulation increases, since the increase in time does not affect the basic connections of the RNN. That is, the number of hierarchies depends on the number of time course data. When the sampling period for an HFPN is too short to get result from learning, resampling process is carried out in order to avoid the increase of hierarchies. Constants are assigned to all units of the 1st hierarchy of the time expanded BPTT.

Since results of learning on only one pattern tend to be in local minimum, we used two patterns in Table 1 as input signals and teaching signals. Pattern 1 in Table 1 corresponds to a learning pattern in steady state, in which all contents of places are used as input signals. The value “0” of teaching signal in steady state reflects the situation in which the biological pathway is not affected by deletion of the arc. On the contrary, the value “1” of teaching signal reflects the situation in which the biological pathway is affected by deletion of the arc. Pattern 1 allows each value in the output layer to be a number close to zero when a state of an RNN approaches to the steady state. On the contrary, Pattern 2 allows each value in the output layer to be a number close to one when a state of an RNN is unstable. Learning processes can be performed effectively based on these two patterns.

In addition, since the number of connections in an RNN increases with increasing complexities of biological pathways, values in the output layer become larger in more complex biological pathways, making it possible to have more higher accuracy in analyzing debacle points of biological pathways. Details for this analysis will be described below.

Figure 2 shows general transition of error differences when learning processes of RNNs are performed by BPTT method. We can see that, in early period of the learning in Figure 2, behaviors of two patterns in Table 1 are converged to some level by BPTT. The next period of learning is the process to reduce differences in the output layer. If these differences decrease with increasing learning times, it can be said that reforming process of RNN is performed successfully. On the contrary, if these differences do not decrease in that period, RNN might be in a local minimum. In this case, some techniques such as simulated annealing have to be applied in order to escape from a local minimum. If the learning process is performed successfully, it reaches a steady state from which differences are not improved more. From Figure 2, we can see that learning is not improved from the 10,000 times. For efficient predictions of debacle points, the learning process has to be ended before it reaches a steady state.

3.2 2nd Stage: Error Detection by Sequential Deletions of Arcs

The next process after the completion of learning process is to detect errors in RNNs, which corresponds to sequential removal of arcs in an HFPN. Figure 3 (b) shows the RNN model whose connections are partially deleted based on the removal of arc a2 from place p1 to transition t3 in Figure 3 (a). Accordingly, the connections from *time=1* to *time=n* corresponding to the arc a2 of the HFPN are removed as shown in the dotted arc in the time expanded RNN of Figure 3 (c). That is, in accordance with removal of arc from an HFPN, the connections between any pair of units are deleted from a time expanded RNN. Note that these deletions result in errors in the output unit. If these errors are large, the deleted arc shall play an essential role in the original biological pathway, which will break robustness of the pathway. In contrast, if these errors are small, the corresponding reaction in the original biological pathway shall not affect the robustness.

By repeating the above removal processes of the connections in a time expanded RNN for each arc in an HFPN model (arcs a_1 to a_6 in Figure 3), we obtain errors in the output layer which will be used in the next stage for debacle points prediction.

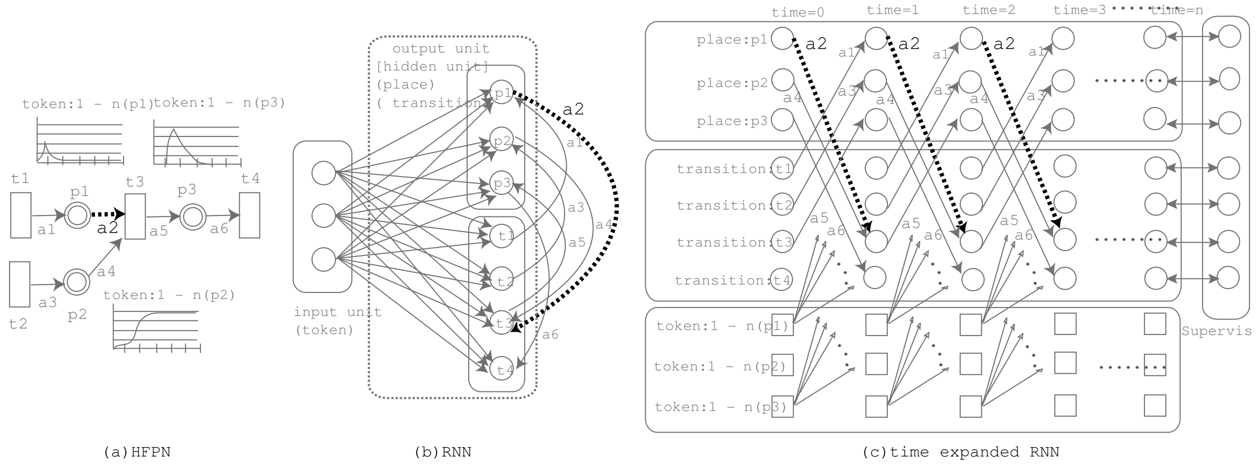

Figure 3: 2nd stage : Error detection by sequential deletion of arcs.

3.3 3rd stage: Debacle Points Prediction Based on Errors

The graph of Figure 4 shows results of errors produced from the RNN. The horizontal axis of Figure 4 represents the arc number shown in No. column of Table 3 which will be sorted in descending order of errors (see Table 4). As mentioned in the previous section, an arc number of larger error is more likely to be break down than smaller error one. Based on errors, all arc numbers are classified into two categories of robust arc numbers and fragile arc numbers.

In the following, we show a method to decide the threshold at which these two categories are

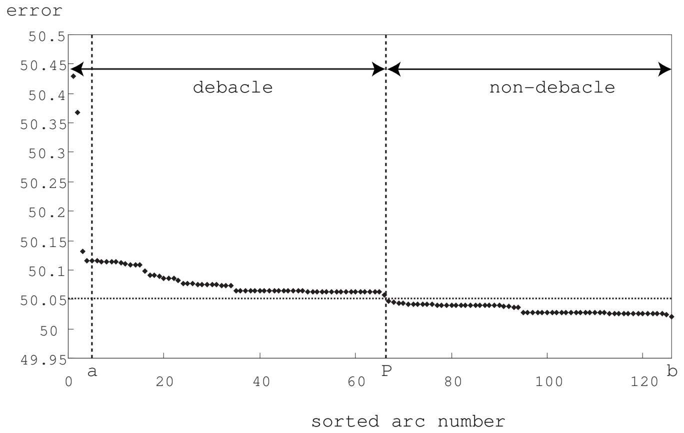

Figure 4: 3rd stage : Determining the threshold dividing debacle and non-debacle points by errors produced from the RNN.

classified. First, we calculate the difference

$$d(n) = y(n-1) - y(n),$$

where n is the sorted arc number presented in the first column of Table 4 (and also the horizontal axis of Figure 4) and $y(n)(1 \leq n)$ is the error of arc number n. That is, $d(n)$ is the difference of two neighboring arc numbers sorted by errors. The maximum $d(n)$ is chosen as the threshold, while some arc numbers of high errors are omitted from candidates of n. The reason of this omission is that the very early stages of learning are sometimes affected by initial conditions of weights of connections in an RNN which might be in a local minimum. This phenomenon is not essential, but may produce unfavorable specific output of errors for predictions. In the experimentation of Figure 4, after omitting the largest six errors of such phenomena, the range between the marked points "a" and "b" in the horizontal axis of Figure 4 has been obtained. From this range, the arc marked with "p", where the largest difference in errors between two neighboring sorted arcs occurs, was chosen as the arc of the smallest error of debacle points. This point "p" distinguishes two classes of debacle and non-debacle points.

4 Experimentation for Debacle Points Prediction on Apoptosis Signaling Pathway

4.1 Signaling Pathway of Apoptosis Induced by Fas Ligand

We conducted an experimentation in order to evaluate the efficiency of the proposed method by using signaling pathway for apoptosis. This pathway is appropriate for the first evaluation because it does not include complex set of reactions such as a feedback loop. Although several apoptosis signaling pathways have been known, we adopted the signaling pathway induced by Fas ligand for evaluation, since it is one of the most investigated signaling pathways of apoptosis. Figure 5 and Figure 6 show this signaling pathway and its HFPN model, respectively. From Figure 5, we can see that the pathway branches off in two pathways at caspase 8 and these two pathways meet again at caspase 3. The HFPN model of Figure 6 was constructed by including autocatalytic reactions in a basic HFPN model naively transformed from Figure 5. Refer to [5] for the details of this HFPN model.

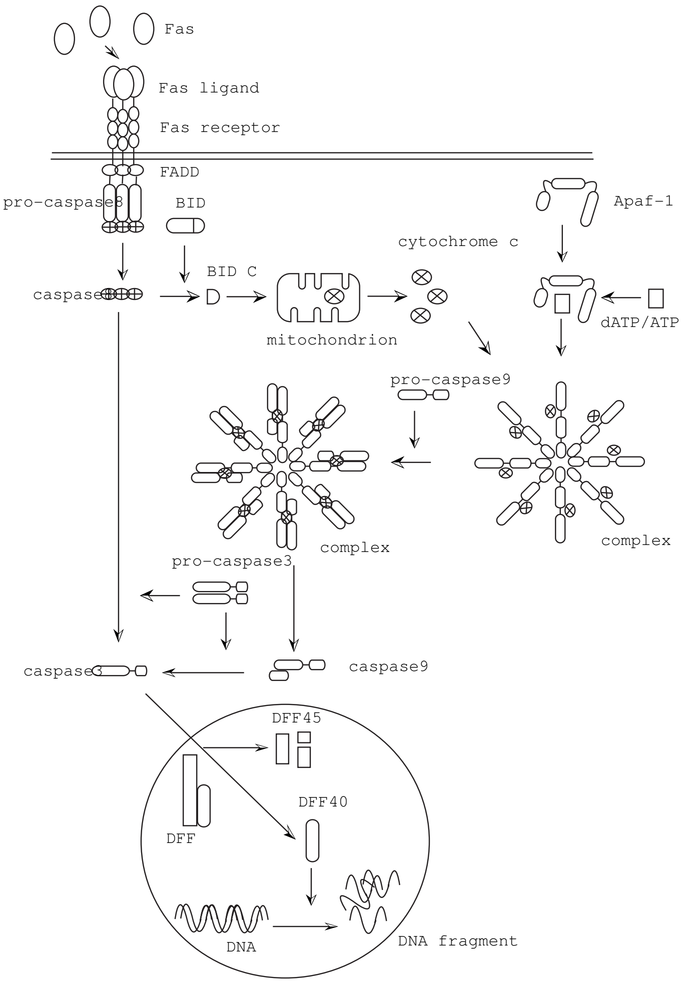

Figure 5: Signaling pathway of apoptosis.

4.2 Manual Extraction of Debacle Points

The HFPN model of signaling pathway includes 39 places, 78 transitions, and 126 arcs. Based on this HFPN model, we constructed an RNN.

Concentration behavior of each place has been sampled at 2003 points and these 2003 points were reduced to 200 points by the down sampling. 0.9 is set to the two parameters α and $\mu 0$ which determine learning constant, and random values are used as initial values of weights.

Evaluation of the proposed method has been carried out by comparing the debacle points predicted by the proposed method with the debacle points obtained by deletion of each arc in the HFPN model of the apoptosis signaling pathway. Deletions of arcs are carried out by hand, and effects of deletions are classified into five levels based on the number of damaged places, that is, these levels are remarkably different from these normal behaviors. These levels are labeled with the numbers 1 to 5 according to the number of damaged places as shown in the first and second columns of Table 2. Table 3 shows correspondences between arcs and these five levels.

In the third and the fourth columns of Table 2, the number of damaged places for each of five levels and its percentage are presented, respectively. We assumed that an arc which is classified as levels 3, 4 or 5 is the arc which gives serious damage to the biological pathway, and these levels are distinguished with bold characters in Table 3. That is, arcs of bold characters represent debacle points for the apoptosis pathway. There are 42 arcs of debacle points and 84 debacle points of non-debacle

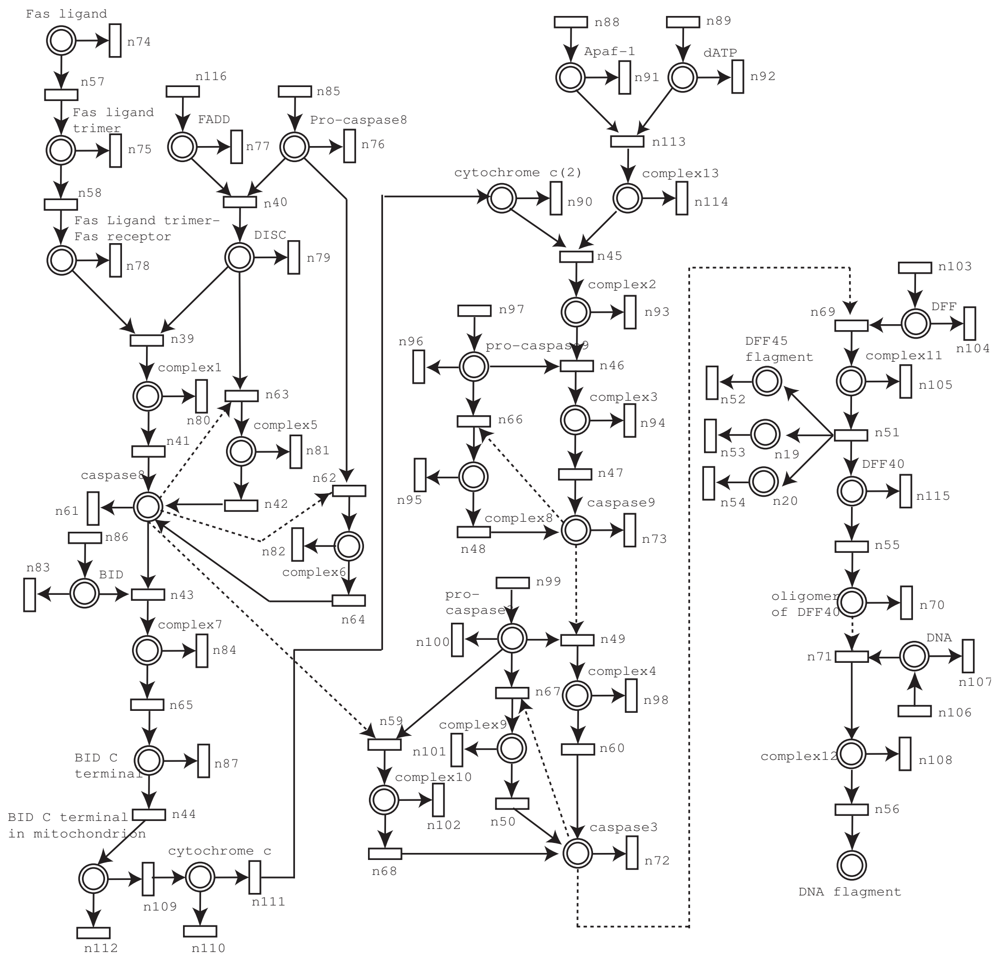

Figure 6: HFPN model of apoptosis signaling pathway.

points in Table 3.

4.3 Results of Predictions

Table 4 is obtained from Table 3 by sorting arcs with respect to errors produced by the time expanded RNN. Learning processes are performed 3000 times on this RNN. The line drawn around at the center of the table represents the threshold 55.24, which is shown in Figure 4, obtained from the result of predictions by the time expanded RNN. The arcs above (below) this line are arcs which are predicted as debacle (non-debacle) points in the apoptosis pathway.

From Table 4, we can see that 40 arcs (57 arcs) are predicted as debacle (non-debacle) points among 42 arcs (87 arcs). However, 27 arcs (2 arcs) are falsely predicted as debacle points (non-debacle points) by BPTT, while these arcs were classified as level 1 or 2 (level 3, 4, or 5) in Table 2 by manual extraction of debacle points. In summary, the proposed method with the RNN exhibits high rate of efficiency of 96.5 % (55/57) in predicting debacle points. However, for the prediction of non-debacle points, the proposed method have gave only the rate of 65.5 % (57/87). We are now working on improving the proposed method in this paper for upgrading the quality of prediction as well as for reducing the numbers of false positives and false negatives shown in Table 4.

Table 2: Debacle levels based on the number of damaged places and the number of arcs of these levels.

No. of damaged places	levels	No. of arcs	ratio
0 ∼ 2	1	84	66.67 %
3 ∼ 5	2	17	13.49 %
6 ∼ 8	3	14	11.11 %
9 ∼ 11	4	4	3.17 %
above 12	5	7	5.56 %

5 Conclusion

This paper proposed a method to use RNNs for predicting debacle points in a biological pathway which will give serious damages to the the biological pathway. With BPTT method, the RNN learns behaviors of places in an HFPN model of the biological pathway. With the RNN which has finished the learning, we realized the automatic prediction system for the debacle points, to which an XML file of an HFPN model of a biological pathway and time series data of all places of the HFPN are given, and from which a table as Table 5 presenting debacle points is produced. This automatic system allows us to predict the debacle point in a number of hours, while more than a day has been spent by a method as in Section 4.2 with repeating a simulation for each deletion of all arcs in the HFPN.

The evaluation on the example of apoptosis shows the availability of the proposed method to a practical use in debacle points prediction. However, this paper only considers the availability from the aspects of computational efficiency. The availability has to be also discussed from biological aspects, that is, biological meaning has to be discussed for each debacle point predicted from RNN as well as for each behavior resulted from the removal of an arc. Hence, our efforts will be dedicated in applying our method to more varieties of biological pathways including metabolic pathways and gene regulation networks with considering biological meanings of debacle points suggested from the RNN.

References

[1] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P., *Molecular Biology of The Cell*, Garland Science, 2002.

[2] Alla, H. and David, R., Continuous and hybrid Petri nets, *Journal of Circuits, Systems, and Computers*, 8(1):159–188, 1998.

[3] Doi, A., Nagasaki, M., Fujita, S., Matsuno, H., and Miyano, S., Genomic Object Net:II. Modeling biopathways by hybrid functional Petri net with extension, *Appl. Bioinformatics*, 2:185–188, 2004.

[4] Kitano, H., *Foundations of Systems Biology*, MIT Press, 2001.

[5] Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., and Miyano, S., Biopathways representation and simulation on hybrid functional Petri Net, *In Silico Biology*, 3(3):389–404, 2003.

[6] Nagasaki, M., Doi, A., Matsuno, H., and Miyano, S., Genomic Object Net:I. A platform for modeling and simulating biopathways, *Appl. Bioinformatics*, 2:181–184, 2004.

[7] Reising, W., *Petri Nets*, Springer-Verlag, 1985.

[8] Rumelhart, D.E., Hinton, G.E., and Williams, R.J., Learning representation by back-propagation errors, *Nature*, 323:533–536, 1986.

[9] Williams, R. and Zipser, D., A learning algorithm for continually running fully recurrent neural networks, *Neural Computation*, 1:270–280, 1989.

[10] `http://genome.ib.sci.yamaguchi-u.ac.jp/~gon/`

[11] http://www.gene-networks.com/english/index.html

[12] http://www.genomicobject.net/

Table 3: Damage levels of arcs by manual deletion.

from : to	R
FADD : n77	1
FADD : n40	1
complex13 : n114	1
complex13 : n45	1
cytochrome c : n111	2
cytochrome c : n110	1
BID C terminal in mitochondrion : n112	1
BID C terminal in mitochondrion : n109	1
complex12 : n108	1
complex12 : n56	1
complex11 : n105	1
complex11 : n51	3
complex10 : n102	1
complex10 : n68	3
complex9 : n101	1
complex9 : n50	4
complex8 : n95	1
complex8 : n48	3
complex7 : n84	1
complex7 : n65	2
complex6 : n82	1
complex6 : n64	4
complex5 : n81	1
complex5 : n42	4
caspase3 : n72	1
caspase3 : n69	1
caspase3 : n67	1
complex4 : n98	1
complex4 : n60	2
Fas ligand trimer : n75	1
Fas ligand trimer : n58	3
Fas ligand : n74	1
Fas ligand : n57	3
DNA : n107	1
DNA : n71	1
oligomer of DFF40 : n71	1
oligomer of DFF40 : n70	1
n20 : n54	1
n19 : n53	1
DFF45 fragment : n52	1
DFF40 : n115	1
DFF40 : n55	1
DFF : n104	1
DFF : n69	1
pro-caspase3 : n100	1
pro-caspase3 : n67	1
pro-caspase3 : n59	1
pro-caspase3 : n49	1
caspase9 : n73	1
caspase9 : n66	1
caspase9 : n49	1
complex3 : n94	1
complex3 : n47	2
pro-caspase9 : n96	1
pro-caspase9 : n66	1
pro-caspase9 : n46	1
complex2 : n93	1
complex2 : n46	1
dATP : n113	1
dATP : n92	1
Apaf-1 : n113	1
Apaf-1 : n91	1
cytochrome c(2) : n90	1
cytochrome c(2) : n45	1
BID C terminal : n87	1
BID C terminal : n44	2
BID : n83	1
BID : n43	2
caspase8 : n63	1
caspase8 : n62	1
caspase8 : n61	2
caspase8 : n59	1
caspase8 : n43	1
complex1 : n80	1
complex1 : n41	3
DISC : n79	1
DISC : n63	1
DISC : n39	1
pro-caspase8 : n76	1
pro-caspase8 : n62	1
pro-caspase8 : n40	1
Fas ligand trimer-Fas receptor : n78	1
Fas ligand trimer-Fas receptor : n39	1
n116 : FADD	2
n113 : complex13	3
n111 : cytochrome c(2)	4
n109 : cytochrome c	5
n106 : DNA	1
n103 : DFF	2
n99 : pro-caspase3	2
n97 : pro-caspase9	2
n89 : dATP	3
n88 : Apaf-1	3
n86 : BID	3
n85 : pro-caspase8	2
n71 : complex12	1
n69 : complex11	3
n68 : caspase3	2
n67 : complex9	1
n66 : complex8	1
n65 : BID C terminal	5
n64 : caspase8	1
n63 : complex5	1
n62 : complex6	1
n60 : caspase3	2
n59 : complex10	1
n58 : Fas ligand trimer-Fas receptor	5
n57 : Fas ligand trimer	5
n56 : DNA fragment	1
n55 : oligomer of DFF40	2
n51 : n20	1
n51 : n19	1
n51 : DFF45 fragment	1
n51 : DFF40	2
n50 : caspase3	1
n49 : complex4	2
n48 : caspase9	1
n47 : caspase9	3
n46 : complex3	3
n45 : complex2	3
n44 : BID C terminal in mitochondrion	5
n43 : complex7	5
n42 : caspase8	1
n41 : caspase8	5
n40 : DISC	1
n39 : complex1	1

Table 4: Results of prediction.

No.	from : to	R
1	n56 : DNA fragment	1
2	complex11 : n105	1
3	n69 : complex11	3
4	n59 : complex10	1
5	n71 : complex12	1
6	n49 : complex4	2
7	n43 : complex7	5
8	complex12 : n108	1
9	n67 : complex9	1
10	n66 : complex8	1
11	n46 : complex3	3
12	n39 : complex1	1
13	n45 : complex2	3
14	n63 : complex5	1
15	n113 : complex13	3
16	n40 : DISC	1
17	n19 : n53	1
18	n20 : n54	1
19	DFF45 fragment : n52	1
20	n60 : caspase3	2
21	n50 : caspase3	1
22	n68 : caspase3	2
23	n44 : BID C terminal in mitochondrion	5
24	n65 : BID C terminal	5
25	n48 : caspase9	1
26	n47 : caspase9	3
27	n111 : cytochrome c(2)	4
28	n57 : Fas ligand trimer	5
29	n42 : caspase8	1
30	n41 : caspase8	5
31	n64 : caspase8	1
32	n109 : cytochrome c	5
33	n58 : Fas ligand trimer-Fas receptor	5
34	n55 : oligomer of DFF40	2
35	n51 : n19	1
36	n51 : DFF45 fragment	1
37	n51 : n20	1
38	n51 : DFF40	2
39	complex12 : n56	1
40	n116 : FADD	2
41	n106 : DNA	1
42	n103 : DFF	2
43	n99 : pro-caspase3	2
44	n97 : pro-caspase9	2
45	n89 : dATP	3
46	n88 : Apaf-1	3
47	n86 : BID	3
48	n85 : pro-caspase8	2
49	complex11 : n51	3
50	complex9 : n50	4
51	complex10 : n68	3
52	complex4 : n60	2
53	BID C terminal : n44	2
54	BID C terminal in mitochondrion : n109	1
55	cytochrome c : n111	2
56	complex3 : n47	2
57	complex8 : n48	3
58	complex7 : n65	2
59	Fas ligand : n57	3
60	complex6 : n64	4
61	complex1 : n41	3
62	complex5 : n42	4
63	Fas ligand trimer : n58	3
64	DFF40 : n55	1
65	n62 : complex6	1
66	BID C terminal in mitochondrion : n112	1
67	pro-caspase8 : n62	1
68	caspase8 : n62	1
69	FADD : n40	1
70	pro-caspase8 : n40	1
71	Apaf-1 : n113	1
72	complex13 : n45	1
73	caspase8 : n63	1
74	cytochrome c(2) : n45	1
75	DISC : n39	1
76	dATP : n113	1
77	pro-caspase9 : n46	1
78	DISC : tiny n63	1
79	Fas ligand trimer-Fas receptor : n39	1
80	caspase9 : n66	1
81	complex2 : n46	1
82	pro-caspase3 : n67	1
83	caspase8 : n43	1
84	caspase9 : n49	1
85	pro-caspase9 : n66	1
86	caspase3 : n67	1
87	caspase8 : n59	1
88	BID : n43	2
89	pro-caspase3 : n49	1
90	pro-caspase3 : n59	1
91	DNA : n71	1
92	oligomer of DFF40 : n71	1
93	caspase3 : n69	1
94	DFF : n69	1
95	DFF : n104	1
96	FADD : n77	1
97	caspase3 : n72	1
98	cytochrome c(2) : n90	1
99	complex2 : n93	1
100	dATP : n92	1
101	DISC : n79	1
102	Apaf-1 : n91	1
103	oligomer of DFF40 : n70	1
104	DNA : n107	1
105	Fas ligand trimer-Fas receptor : n78	1
106	BID : n83	1
107	pro-caspase8 : n76	1
108	pro-caspase9 : n96	1
109	complex13 : n114	1
110	caspase9 : n73	1
111	pro-caspase3 : n100	1
112	caspase8 : n61	2
113	Fas ligand : n74	1
114	complex7 : n84	1
115	cytochrome c : n110	1
116	BID C terminal : n87	1
117	complex3 : n94	1
118	complex6 : n82	1
119	complex1 : n80	1
120	complex8 : n95	1
121	complex5 : n81	1
122	complex10 : n102	1
123	complex9 : n101	1
124	complex4 : n98	1
125	Fas ligand trimer : n75	1
126	DFF40 : n115	1