

解釈遷移からの学習を用いた遺伝子制御ネットワーク推論

Identification and Inference of Gene Regulatory Networks Using Learning from Interpretation Transition

池田 光¹ 井上 克巳^{2,1}
Hikaru Ikeda¹ Katsumi Inoue^{2,1}

¹ 東京工業大学

¹ Tokyo Institute of Technology

² 国立情報学研究所

² National Institute of Informatics

Abstract: Elucidation of gene regulatory networks is an important issue in systems biology and has been actively studied in recent years. Various inference methods have been proposed so far. However, in many cases, only binary relations between genes are considered, and there is a problem that accuracy cannot be said to be high. On the other hand, an algorithm that learns interpretable rules behind a dynamic system, called Learning From Interpretation Transition (LFIT), has been proposed. LFIT has already been applied to solve the problem of considering only binary relations in existing research. However, when the number of genes increases, the number of rules output by LFIT increases, and there is a problem that interpretability is low. Therefore, we propose a new inference method called Leave-one-out LFIT (LOO-LFIT) that applies LFIT to data excluding one gene and extracts common rules.

1 はじめに

近年、DNA マイクロアレイなどの生物学実験技術の向上により、これらの実験から得られるデータは膨大なものとなっている。これらの大量のデータから解析を行うために、従来の生物学的手法以外にも、機械学習や深層学習といった情報学的なアプローチが重要となり、バイオインフォマティクスと呼ばれる分野が誕生した。さらに、個々の遺伝子やタンパク質を対象とするバイオインフォマティクスから、生命を遺伝子やタンパク質のネットワークとして捉え、システムとして理解しようとするシステム生物学と呼ばれる分野も誕生し注目を浴びている。中でも遺伝子間の相互作用を表す遺伝子制御ネットワークは生体内に存在する様々なネットワークの根幹を成すと言われ、生体内における現象の理解に非常に重要であると考えられており、その解明は重要なテーマの1つとなっている。

遺伝子制御ネットワークの推論手法としては、従来よりブーリアンネットワークを用いた手法 [1] やベイジアンネットワークを用いた手法 [2] が提案されている。近

年では、機械学習や深層学習を用いた手法 [3] も提案されている。しかし、いずれにおいても精度が良いとは言えない状況にある。この要因の1つとして、遺伝子間の二項関係のみを考慮しているということが挙げられる。

一方、動的なシステムからその背後に潜む解釈可能なルールを学習するアルゴリズムとして、解釈遷移からの学習 (Learning From Interpretation Transition; LFIT) と呼ばれる手法 [4] が提案されている。LFIT は、動的なシステムによって観測された2つの隣接する時点間の状態遷移に注目し、各遷移ごとに矛盾しないルールを作成していくことで、動的なシステムを標準論理プログラムとして表現する。LFIT によって学習された標準論理プログラムはブーリアンネットワークに相当し、動的なシステムに含まれる要素間の関係を理解することができる。この関係性を利用することで、システムにおける問題の原因究明や将来の動作予測などが可能となる。

既存研究における二項関係のみを考えているという問題を解決するために、LFIT を適用した例はこれまでもある [5]。しかし、遺伝子数が多くなると LFIT の出力するルール数も多くなってしまい、解釈性に欠けるといふ欠点がある。本研究では、LFIT の新たな使用方法を

提案し、精度を落とすことなく、より少ないルールで表現することで解釈性の向上を目指す。

2 背景

ここでは本研究を理解する上で必要な標準論理プログラムおよび LFIT に関する背景知識について述べる。

2.1 標準論理プログラム

標準論理プログラム (Normal Logic Programming; NLP) は式 (1) の形式をとったルールの集合である。

$$A \leftarrow A_1 \wedge \dots \wedge A_m \wedge \neg A_{m+1} \wedge \dots \wedge \neg A_n \quad (1)$$

ここで、 A や A_i はアトムである。 ($1 \leq i \leq n, n \geq m \geq 0$) また、式 (1) の形式をとる任意のルール R において、 R のボディの正リテラルとして現れるアトムを $b^+(R) = \{A_1, \dots, A_m\}$ と表し、負リテラルとして現れるアトムを $b^-(R) = \{A_{m+1}, \dots, A_n\}$ と表す。

NLP に現れるすべてのアトムはエルブラン基底 \mathcal{B} と呼ばれる。エルブラン解釈 I は \mathcal{B} の部分集合であり、各アトムへの真理値の割り当てである。ルール R および解釈 I が与えられ、 $b^+(R) \subseteq I$ および $b^-(R) \cap I = \emptyset$ が $h(R) \in I$ を含意する時、解釈 I はルール R を満足するという。無矛盾な解釈 I が P の全てのルールを満足するとき、 I は P のモデルである。論理プログラム P と解釈 I を考える時、 T_p オペレータは $T_p : 2^{\mathcal{B}} \rightarrow 2^{\mathcal{B}}$ の対応であり、このとき T_p オペレータを式 (2) に定義する。

$$T_p(I) = \{h(R) | R \in P, b^+(R) \subseteq I, b^-(R) \cap I = \emptyset\} \quad (2)$$

時間に関して変化するシステムのダイナミクスを表現するためには時間を引数として使用する。つまり、時刻 t におけるアトム A を $A(t)$ とみなす。これにより、式 (1) のルールを動的なルールに書き換えることができる。

$$A(t+1) \leftarrow A_1(t) \wedge \dots \wedge A_m(t) \wedge \neg A_{m+1}(t) \wedge \dots \wedge \neg A_n(t) \quad (3)$$

これは、「 A_1, A_2, \dots, A_m のすべてが t 時点で真であり、 $A_{m+1}, A_{m+2}, \dots, A_n$ のすべてが t 時点で偽であるとき、 $t+1$ 時点で A は真」という意味になる。式 (3) のように記述することで、 T_p オペレータを使用して、動的システムの状態遷移をシミュレートすることができる。

2.2 LFIT

LFIT では、2 つの隣接する時点間での状態遷移に注目する。解釈 I が t 時点でのシステムの状態を反映しているとき、 $t+1$ 時点でのシステムの状態 J も解釈となる。この場合、動的システムにおける I から J への状態遷移を (I, J) と表すことができる。解釈遷移の集合 E が与えられた時、その全ての遷移 $(I, J) \in E$ について $J = T_p(I)$ を満たす論理プログラム P を学習するアルゴリズムを LFIT と呼ぶ [4]。

ルール R と状態遷移 (I, J) が与えられたとき、 $b^+(R) \subseteq I$ かつ $b^-(R) \cap I = \emptyset$ が $h(R) \in J$ を含意するときに限り、 R は (I, J) と無矛盾である。 R が集合 E に含まれるすべての状態遷移と無矛盾である場合、 R は E と無矛盾である。論理プログラム P に含まれるすべてのルール R が E と無矛盾である場合、論理プログラム P は E と無矛盾である。

論理演算を使用して LFIT アルゴリズムは複数の方法で実装されている。ボトムアップ方式は、最も一般的なルールから始まり、肯定的な例を汎化させていく方法である。つまり、汎化を用いながら、似ている状態で同じ帰結をもつ場合に規則の条件を弱めていく方法である。一方、トップダウン方式は最も一般的なルールから始まり、否定的な例を特殊化させていく方法である。つまり、条件が空の規則から始めて、反例が見つかるたびに規則を特殊化していく方法である。

3 提案手法

ここでは、LFIT をもちいた新たな遺伝子制御ネットワーク推論手法について述べる。

3.1 入力作成

DNA マイクロアレイで得られたデータである遺伝子発現プロファイルは測定間隔が不均等であり、測定値が連続値であるという特徴を持つ。一方、LFIT は時点が等間隔であり、データが離散値であることを前提としている。前者に対しては、元のデータから時点が等間隔になるように抽出を行う。後者に対しては、元のデータに対して、先行研究 [6, 7] からの 7 つの離散化手法を施し、データを離散値に変換する。ここで、 GEP を N 行 M 列の行列とし、遺伝子発現プロファイル (Gene Expression Profile; GEP) を表すとす。 N は遺伝子の数を、 M は時点数を表す。 $GEP(n, m)$ は時点 m での遺伝子 n の発現量を表し、 $GEP(n, :)$ はすべての時点

での遺伝子 n の発現データを表し、 $GEP(:, m)$ は時点 m でのすべての遺伝子の発現データを表す。また、離散化後の行列 GEP を GEP_d と表し、 $GEP(n, m)$ を離散化した値を $GEP_d(n, m)$ と表すとす。

(1) Mean

$GEP(n, m)$ が $GEP(n, :)$ の平均値以上であれば 1, そうでなければ 0 とする。

(2) Median

$GEP(n, m)$ が $GEP(n, :)$ の中央値以上であれば 1, そうでなければ 0 とする。

(3) Decrease or Increase

1 時点前との発現量の増減を計算し、増加していれば 1, 減少していれば 0 とする。

(4) Kmeans

$GEP(n, :)$ に対して、kmeans クラスタリングによって分割を行う。

(5) Cokmeans (Column Kmeans Discretization)

Cokmeans は $GEP(:, m)$ を kmeans クラスタリングによって k 分割し、時点 m での発現量を近い測定値を同じクラスに分類する。このとき、各遺伝子によって発現量の幅が異なるため、平均中心のデータになるように発現量を調整した。

(6) Bikmeans (Bidirectional Kmeans Discretization)

kmeans と Cokmeans をともに行い、2 つの離散化手法によって $GEP(n, m)$ の離散値をそれぞれ取得する。このとき、分割数は実際に使用する分割数 k より 1 多い、 $k+1$ とする。 $GEP(n, m)$ における得られた 2 つの離散値の積が x^2 以上 $(x+1)^2$ 未満である場合、 $GEP(n, m)$ の離散値を x とする。ここで、 x は $(1 \leq x \leq k)$ を満たす自然数である。

(7) SSD (Short Series Discretization)

$GEP(n, :)$ を用いて、完全グラフを作成する。このとき、頂点は各発現量を表し、エッジの重みは頂点間におけるユークリッド距離とする。重みが最も大きいエッジから削除していき、連結グラフでなくなるまで続ける。連結グラフでなくなったとき、必要とする離散化レベルの分割数であれば終了し、そうでなければ同様にエッジを削除する。これにより、必要とする個数にデータを分割する。

3.2 LFIT の利用

これまでに行われていた単純に LFIT を適用した遺伝子制御ネットワーク推論の全体の流れを図 1 に示す。

まず、遺伝子発現プロファイルに対して、離散化を行

う。次に、等間隔時点になるよう抽出を行い、抽出されたデータをもとに LFIT の入力を作成し、LFIT を適用する。このとき、非決定的な遷移を含む可能性があるため、LUST アルゴリズム [9] を用いた、LFIT の出力は、ヘッドのアトムが 1 になるルールと 0 になるルールが存在する。本研究では、遺伝子の発現する場合に注目し、ヘッドのアトムが 1 になるルールを考え、これらのいずれのルールも満たさない場合を発現していないとした。従って、LFIT の出力のうちヘッドのアトムが 1 になるルールの集合によって、入力の遺伝子発現プロファイルを表現する。

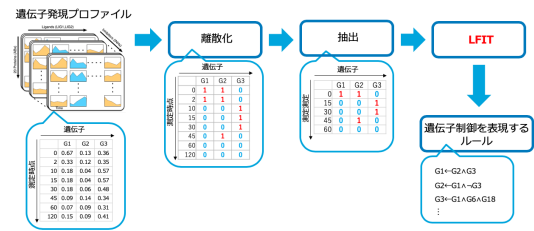


図 1 LFIT による遺伝子制御ネットワーク推論の流れ

3.3 Leave-one-out LFIT(LOO-LFIT)

LFIT を用いた新たな手法として、生物学実験に着想を得た手法を提案する。生物学実験では、重要な遺伝子を見つけるために、1 つ遺伝子を除いて実験を行い、目的を達成するかを評価する実験方法がある。これに似た方法として、Leave-one-out LFIT(LOO-LFIT) を提案する。LOO-LFIT を用いた遺伝子制御ネットワーク推論の全体の流れを図 2 に示す。

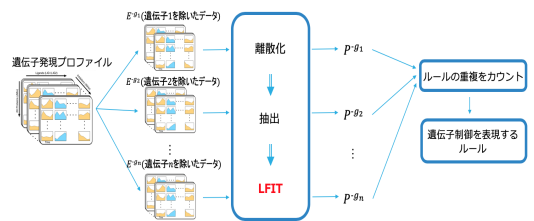


図 2 LOO-LFIT による遺伝子制御ネットワーク推論の流れ

はじめに、遺伝子発現プロファイル GEP から、遺伝子 g_i を除いたデータ GEP^{-g_i} を作成する ($1 \leq i \leq N$)。この時、 GEP^{-g_i} は $N-1$ 行 M 列の行列となる。次に、 GEP^{-g_i} に対し、離散化を行ったデータ $GEP_d^{-g_i}$ を作成する。遺伝子 g_i を除いたデータ $GEP_d^{-g_i}$ から、等間隔時点になるように抽出をおこない、LUST アルゴリズムを適用する。得られた出力は遺伝子 g_i 以外によって作られる遺伝子制御に関するルールとなる。ここで

ヘッドのアトムが 1 になるルールを抽出し、その集合を P^{-g_i} とする。得られたルールの集合 P^{-g_i} の多重集合をとり、各ルールについて重複をカウントする。最後に、重複が y 個 ($1 \leq y \leq N - 1$) 以上のルールの集合 C_y を作成する。このようにして得られた集合 C_y によって入力の変異子発現プロファイルを表現する。

4 評価実験

4.1 データセット

本研究では、遺伝子制御ネットワークの推論を行うためのデータである、Dream8 における人工データを用いた。ここで DREAM プロジェクトは、生物学におけるシステムを新しい計算モデルや手法を用いてデザインすることを目的としている。この人工データは計算機シグナル伝達モデルが組み込まれた GeneNetWeaver[8] にて、リンタンパク質の時系列の変化を生成したものであり、20 個のリンタンパク質の発現量が 0~120 の等間隔でない 11 の時点でわたって計測されている。シミュレーション環境は促進遺伝子 (2 種類) と抑制遺伝子 (3 種類) の組み合わせにより 20 個あり、それぞれの環境に対して 3 回実験を行っている。つまり、20 個の遺伝子を 11 時点でわたって計測したデータが各シミュレーション環境に対し、3 回分の 60 個あるということになる。そこで、各シミュレーション環境における各遺伝子の各時点での発現量は 3 回のシミュレーションの平均を取ることで算出した。

4.2 評価指標

本研究では、2 つの評価指標を用いた。

1 つ目は二乗平均平方根誤差 (Root Mean Squared Error; RMSE) である。RMSE は回帰モデルの誤差を評価する指標の 1 つである。RMSE は式 (4) で与えられ、0 に近づくほどよいモデルであると評価できる。

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4)$$

2 つ目はルールの数である。RMSE による精度が同程度の場合、より少ないルールで対象のネットワークを表現できるモデルの方が、解釈性が高いと考えられ、より有効なモデルであると言える。

4.3 実験

4.3.1 遺伝子制御ネットワーク構築実験

構築実験の全体の流れを図 3 に示す。

LFIT では、20 種の実験環境それぞれに対して、離散化を行い、観測時点が等間隔になるよう、0,15,30,45,60 時点のデータを抽出した。抽出したすべての状態遷移を LUST アルゴリズムの入力とした。得られたルールの集合を、20 種の実験環境の初期状態に適応し、シミュレーションを行い、入力の変異子とシミュレーションしたデータとの RMSE を計算した。

LOO-LFIT では、用いるデータから 1 つの遺伝子を取り除いた 20 個の新たなデータを作成し、新たな 20 個のデータに対して、それぞれ離散化を行い、観測時点が等間隔になるよう、0,15,30,45,60 時点を対象としてデータを抽出した。次に、抽出後の 20 個のデータに対し、LUST アルゴリズムを適用し、20 個の出力を得た。ここで、各出力は 1 つの遺伝子を除いた場合のルールの集合となっている。これらの出力において、あるルールに注目すると、そのルールは最大で 19 個の出力に出現する可能性がある。重複して出現するルールは重要であるという考えから、重複が y ($15 \leq y \leq 19$) 個以上であるルールを抽出してシミュレーションを行い、入力の変異子とシミュレーションしたデータとの RMSE を計算した。

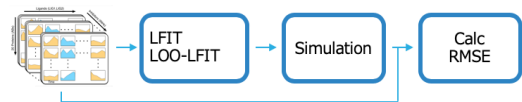


図 3 構築実験の流れ

4.3.2 遺伝子制御ネットワーク推論実験

推論実験の全体の流れを図 4 に示す。

推論実験では、20 種類ある実験環境のデータを 5 つに分割し、4 つを訓練データ、1 つをテストデータとした。4 つの訓練データに対して、構築実験と同様に実験を行い、出力されたルールによってシミュレーションしたデータと 5 分割した残りの 1 つであるテストデータとの RMSE を計算した。

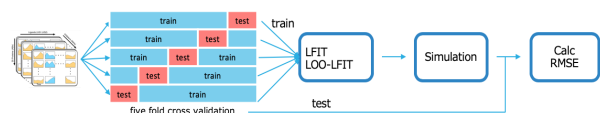


図 4 推論実験の流れ

4.4 シミュレーション

LFIT によって出力されるルール集合は” $G1(1, T) :: 3/4 :- G3(1, T-1), G9(0, T-1), G13(1, T-1)$.” のようなルールの集合となっている。これは、「T-1 時点で G3 と G13 が発現していて G9 が発現していないとき、 $\frac{3}{4}$ の確率で T 時点で G1 が発現する」という意味になる。そこで、このようなルールから、” $G1 \leftarrow G3 \wedge \neg G9$ ” というルールを作成する。現時点の状態をボディに当てはめ、ボディが真になる場合に、75% の確率で 1 を出力する確率変数を用意し、シミュレーションを行う。

単純に LFIT を適用した手法では、出力されたルール集合に含まれるすべてのルールを用いて、シミュレーションを行う。一方、LOO-LFIT では、20 個のルール集合のうち y 個 ($15 \leq y \leq 19$) に共通して出現するルールのみを抽出してシミュレーションを行う。

シミュレーションはすべて 100 回行い、それぞれにおいて RMSE を計算し、平均 RMSE を算出した。

4.5 結果

4.5.1 遺伝子制御ネットワーク構築実験

表 1 は遺伝子制御ネットワークの構築実験における、RMSE を計算した結果である。また、表 2 はルール数を比較した結果である。太文字は各離散化手法において最も精度がよかった値である。図 5 は Mean における各手法の RMSE とルール数を表した図である。

すべての離散化手法において、LFIT のほうが精度の良い結果となった。しかし、Mean や Median について考えると、精度が同程度の場合、LOO-LFIT ではより少ないルールでモデルを表現できている。

表 1 RMSE の結果

離散化手法	Mean	Median	Dec or Inc	Kmeans	Cokmeans	Bikmeans	SSD
LFIT の利用	0.000	0.000	0.000	0.374	0.414	0.251	0.583
LOO-LFIT : $y = 15$	0.000	0.000	0.025	0.430	0.419	0.254	0.584
LOO-LFIT : $y = 16$	0.064	0.064	0.051	0.439	0.422	0.316	0.583
LOO-LFIT : $y = 17$	0.563	0.563	0.051	0.472	0.448	0.467	0.583
LOO-LFIT : $y = 18$	0.652	0.652	0.510	0.472	0.468	0.583	0.517
LOO-LFIT : $y = 19$	0.656	0.656	None	0.472	0.496	0.735	0.489

表 2 ルール数の結果

離散化手法	Mean	Median	Dec or Inc	Kmeans	Cokmeans	Bikmeans	SSD
LFIT の利用	284,779	284,779	192,198	7,101	12,877	16,734	261
LOO-LFIT : $y = 15$	118,529	118,529	100,570	3,006	10,839	10,919	265
LOO-LFIT : $y = 16$	19,409	19,409	21,915	1,403	7,071	5,700	265
LOO-LFIT : $y = 17$	897	897	1,315	538	1,631	1,455	243
LOO-LFIT : $y = 18$	25	25	28	94	99	133	101
LOO-LFIT : $y = 19$	1	1	0	10	1	1	13

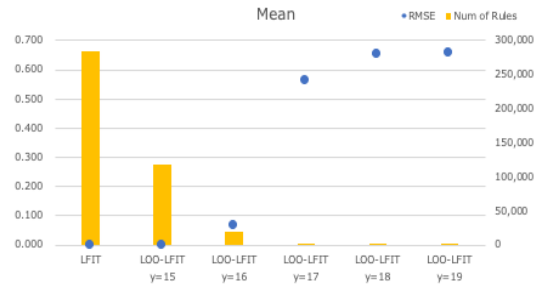


図 5 Mean による結果

4.5.2 遺伝子制御ネットワーク推論実験

表 3 は遺伝子制御ネットワークの推論実験における、RMSE を計算した結果である。また、表 4 はルール数を比較した結果である。太文字は各離散化手法において最も精度がよかった値である。図 6 は Kmeans における各手法の RMSE とルール数を表した図である。

すべての離散化手法において、LOO-LFIT のほうが精度の良い結果となった。また、ルール数に関しても、LFIT より LOO-LFIT のほうが少ないルールで表現できている。

表 3 RMSE の結果

離散化手法	Mean	Median	Dec or Inc	Kmeans	Cokmeans	Bikmeans	SSD
LFIT の利用	0.600	0.600	0.602	0.623	0.631	0.526	0.589
LOO-LFIT : $y = 15$	0.600	0.600	0.602	0.613	0.629	0.526	0.590
LOO-LFIT : $y = 16$	0.600	0.600	0.602	0.718	0.719	0.486	0.708
LOO-LFIT : $y = 17$	0.614	0.614	0.592	0.468	0.547	0.553	0.591
LOO-LFIT : $y = 18$	0.654	0.654	0.616	0.472	0.529	0.605	0.535
LOO-LFIT : $y = 19$	0.656	0.656	None	0.472	0.501	0.735	0.497

表 4 ルール数の結果

離散化手法	Mean	Median	Dec or Inc	Kmeans	Cokmeans	Bikmeans	SSD
LFIT の利用	190,309.0	190,309.0	126,591.2	5,592.4	10,038.4	12,812.4	212.2
LOO-LFIT : $y = 15$	102,895.8	102,895.8	80,156.0	2,944.2	8,835.8	9,270.8	201.4
LOO-LFIT : $y = 16$	20,808.4	20,808.4	21,740.4	1,386.0	6,205.4	5,192.8	201.4
LOO-LFIT : $y = 17$	1,310.6	1,310.6	1,728.4	494.0	1,603.4	1,430.2	186.6
LOO-LFIT : $y = 18$	29.6	29.6	45.6	92.4	120.6	145.0	91.2
LOO-LFIT : $y = 19$	1.2	1.2	0.0	9.8	0.8	1.0	12.2

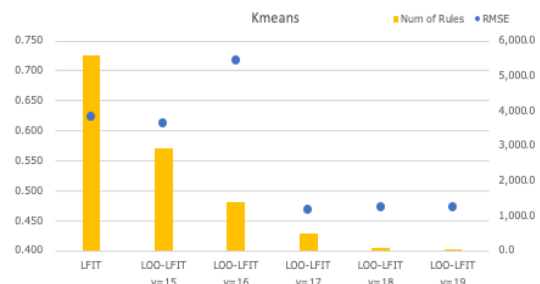


図 6 Kmeans による結果

4.6 考察

本研究における目的は、従来の LFIT の使用法における解釈性の低さを解決するために、精度を落とすことなくより少ないルールで表現することであった。結果をみると、遺伝子制御ネットワークの構築においては、LFIT の方が多くの場合精度が良いという結果になった。しかし、同程度の精度の場合、LOO-LFIT のほうがより少ないルールで遺伝子制御ネットワークを表現できた。また、遺伝子制御ネットワークの推論においては、ほとんどの離散化手法において、LOO-LFIT のほうが精度がよくかつ少ないルールで遺伝子制御ネットワークを表現できている。これらのことから、目的は達成できていると言える。

また、LFIT で得られたルールと LOO-LFIT で得られたルールについて、それぞれで学習したルールを離散化手法ごとに調べたところ、LOO-LFIT で学習したルールは LFIT で学習したルールの完全部分集合となっていた。このことから、LOO-LFIT では LFIT で学習したルールの中でも遺伝子制御ネットワークを表す上で、重要なルールの抽出に成功していると言え、この LFIT の新たな枠組みが有効であると言える。

5 おわりに

本研究では、LFIT の新たな使用法である LOO-LFIT という手法を提案した。これは、実際の生物学的実験に着想を得た手法であり、データセットから 1 つの遺伝子を除いたデータを作り、それぞれに対して LFIT アルゴリズムを適用し、得られた出力から共通して出現するルールを抽出する方法である。人工の遺伝子発現データに対して適用し、単純に LFIT を用いた手法より LOO-LFIT の方が、精度を落とさず少ないルールで遺伝子制御ネットワークを表現できることに成功した。

この枠組みは、これまでの LFIT の出力するルール数が膨大となり解釈性に欠けるという欠点を補うことができると期待され、生物学分野における適応のみならず、ダイナミクス学習が適用できるロボティクスなど他分野にも応用できると期待される。

今後の課題としては、離散化を多値で行うこと、ルールの遷移確率だけでなく出現頻度を考慮に入れること、1 時点前のマルコフ性だけでなく k 時点前も考慮に入れることなどが挙げられる。

参考文献

- [1] Tatsuya Akutsu, Satoru Miyano, Satoru Kuhara: Identification Of Genetic Networks from a Small Number of Gene Expression Patterns Under the Boolean Network Model, *Pacific Symposium on Biocomputing*, Vol. 4, pp. 18–29 (1999)
- [2] Alakwaa Fadhil, Solouma Nahed, Kadah Yasser: Construction of gene regulatory networks using biclustering and bayesian networks, *Theoretical Biology and Medical Modelling*, Vol. 8, No. 39 (2011)
- [3] Abhinandan Khan, Sudip Mandal, Rajat Kumar Pal, et al.: Construction of Gene Regulatory Networks Using Recurrent Neural Networks and Swarm Intelligence, *Scientifica*, Vol. 2016 (2016)
- [4] Katsumi Inoue, Tony Ribeiro, Chiaki Sakama: Learning from interpretation transition, *Machine Learning*, Vol. 94, No/ 1, pp. 51–79 (2014)
- [5] Tony Ribeiro, Morgan Magni, Katsumi Inoue, Chiaki Sakama: Learning Multi-Valued Biological Models with Delayed Influence from Time-Series Observations, *IEEE 14th International Conference on Machine Learning and Applications*, pp. 25–31 (2015)
- [6] Dimitrova Elena, Vera-Licona Paola, Mcgee John, et al.: Discretization of time series data, *Journal of computational biology*, Vol. 17, pp. 853–68 (2010)
- [7] Yong Li, Lili Liu, Xi Bai, et al.: Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks, *BMC Bioinformatics*, Vol. 11, No. 520 (2010)
- [8] Schaffter Thomas, Marbach Daniel, Floreano Dario: GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods, *Bioinformatics*, Vol. 27, pp. 2263–2270 (2011)
- [9] David Martinez, Tony Ribeiro, Katsumi Inoue, et al.: Learning Probabilistic Action Models from Interpretation Transitions, *ICLP* (2015)