2C2-2

# 隠れマルコフモデルによるMATLAB/Simulinkモデルのクローン検出

Clone detection for MATLAB/Simulink model by hidden Markov model

中井 淳一 *1
Junichi Nakai

*1AZAPA 株式会社
AZAPA Co., Ltd.

Model-based development has become widespread, MATLAB/Simulink models are often created as software development. As similar processes called clones increase, it becomes difficult to maintain as well as conventional software development. If there is a problem there, you need to make similar corrections to all corresponding clones. The amount of work increases, causing adverse effects such as omission of correction. Therefore, we can detect clones of model by using Hidden Markov Model and Genetic Algorithm.

## 1. はじめに

ソフトウェア開発において，コピーとペースト操作，意図的な同一処理の繰り返し記述，等により，ソースコード中に同一あるいは類似の部分が増加することが多い．これらはコードクローンと呼ばれ，ソフトウェアの保守を困難にする．例えば，コードクローンに不具合があった場合，開発者はそれと対応する全てのコードクローンに同様の修正を行う必要がある．それにより，作業量が増大し，修正漏れなどの弊害を生む．

近年，モデルベース開発が普及し，ソフトウェア開発としてMATLAB/Simulink モデルを作ることも多いが，クローンと呼ばれる類似処理が増加すると，従来のソフトウェア開発と同様に保守が困難になる．

C 言語を用いて作成されたソフトウェアの場合，プログラム行毎にテキストベースで比較すれば，概ねコードクローンを発見できる．そのためのツールも数多く存在する．しかし，simulink モデルの場合，ブロック線図で作成されたモデルがソフトウェアとなるため，テキストベースで比較することができない．そこで，simulink モデルを深さ優先の中順で，木構造解析し，1 次元のブロック列とすることで，ブロック線図の比較を容易にする．但し，それだけでは，同一のブロック線図しか探索できない．simulink モデルでは，型変換のブロックが追加されただけであったり，定数のブロックが変数のブロックに変わっただけであったりと，少しブロック構成が異なるだけのほぼ同一のブロック線図が存在する．また，ブロック構成は異なるが，同一の機能を実現している場合もある．

本研究では，そのような類似のブロック構成や同一機能で異なるブロック構成をした部分を機能的なクローンとして検出することを狙う．具体的には，simulink モデル用の隠れマルコフモデルを作成し，検出対象と同一機能のブロック線図を複数学習サンプルとして用意し，それを学習させる．検出対象のブロック線図を隠れマルコフモデルに入力すると，似ているほど高い尤度が出力される．このとき，simulink モデル内のどの部分を隠れマルコフモデルに入力するかは遺伝的アルゴリズムによるクラスタリングにより決定する．遺伝的アルゴリズムの遺伝子を simulink モデルのグルーピング情報とする．各グループの尤度が最も高くなるグルーピングを実現することで，検出対象と類似機能を持つブロック構成をグループとして検出することを狙う．

連絡先: 中井淳一，AZAPA 株式会社，nakai@robotjn.com, junichi-nakai@azapa.co.jp

## 2. DSM(Design Structure Matrix)

本研究では，制御ソフト構造，グループ分け結果を見やすくするために，DSM を用いる．DSM (Design Structure Matrix) は Steward らによって考案された [Steward 1981]，工程・組織設計のための手法であり，ノードを項目，ラインを行列として表現する手法である (図 1 参照)．基本アルゴリズムはパーティショニングとクラスタリングである．この手法で出来る事として①見える化，パーティショニングによる②順序整理（手戻り最小化），クラスタリングによる③グルーピングなどが行いやすくなる．

特に本研究では，①の見える化でソフトウェア構造の全体を把握し，隠れマルコフモデルを用いて，モデルのクローンを検出しやすくするために，③グルーピングを遺伝的アルゴリズムを用いて行っている [中井 2015]．

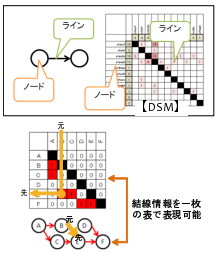



図 1: DSM の概要

## 3. アーキテクチャ

提案手法のアーキテクチャとしては，基本的には遺伝的アルゴリズムを使用している．その概要を図 2 に示す．個体プールには制御ソフトの構成要素のグループ分けを表現する個体 (図 3 参照) が適応度とともに格納されている．個体は構成要素毎の ID とそのグループ番号から構成され，ID: 1，ID: 2 のグループ番号がそれぞれ 1, 1 の場合は ID: 1, ID: 2 は同じグループ 1 に所属し， ID: 1，ID: 2 のグループ番号がそれぞ

れ 1, 2 の場合は ID: 1, ID: 2 は別々のグループ 1 とグループ 2 に所属していることを表現している．具体的には，ID 順にグループ番号を記憶し，例えば構成要素が 10 個なら，個体表現は {0,0,0,1,1,1,2,3,2,2} のようになり，順序が ID, 数字がグループ番号を示している．

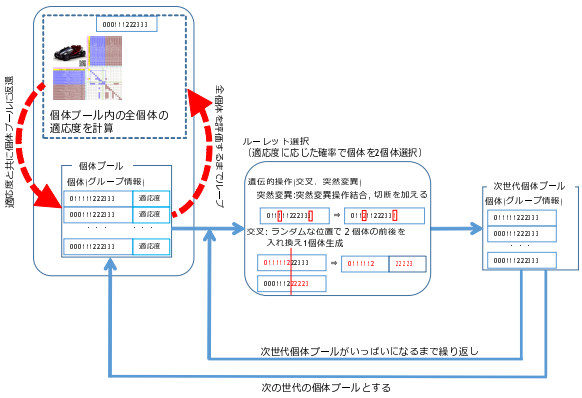



図 2: アーキテクチャ概要

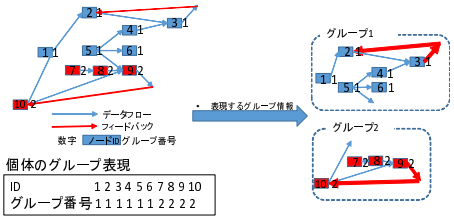



図 3: 個体のグループ表現

まず，個体プール内の全個体を評価する．その際，個体の表現しているグループ分けに対して，隠れマルコフモデルの尤度から，適応度評価関数により，その良し悪しを判定し，その適応度と共に個体プールに格納する．そして，適応度が高い個体ほど高確率となるように個体を選択し，交叉，突然変異の処理を行い，次世代個体プール内に個体プールの個体と同じ数の個体を格納する．それを次の世代の個体プールとみなす．この一連の流れを繰り返すことでグループ分けが，適応度関数が最大となるように，進化する．

遺伝的操作としては突然変異と交叉を行った．突然変異は，対象ノード，あるいは，対象のノード所属グループ全体が隣接する異なるグループ番号のノードと同じ番号に変化する結合と，1 個のグループが 2 個のグループにランダムな位置で分裂する切断，の 2 種類の操作をランダムに行った (図 4 参照)．具体的には，結合は全ノードを対象に，突然変異率 $m$ の確率で対象ノードのみに操作が行われ，さらに，突然変異の対象に選ばれたノードの中から突然変異率 $m$ の確率で，対象のノード所属グループ全体に操作が行われる．そして，切断は全ノードを対象に，突然変異率 $m \times 0.02$ の確率で操作が行われる．具体的な切断操作は，突然変異の対象ノードを起点として異なるグループ番号を付与し，所属グループ内のラインが繋がってるノードを，対象ノードに付与したグループ番号に変更しつつ辿る．そして，突然変異率 $m$ の確率で操作を止めることで，1 個のグループを 2 個のグループに分裂させている．交叉は一点交叉を行った．

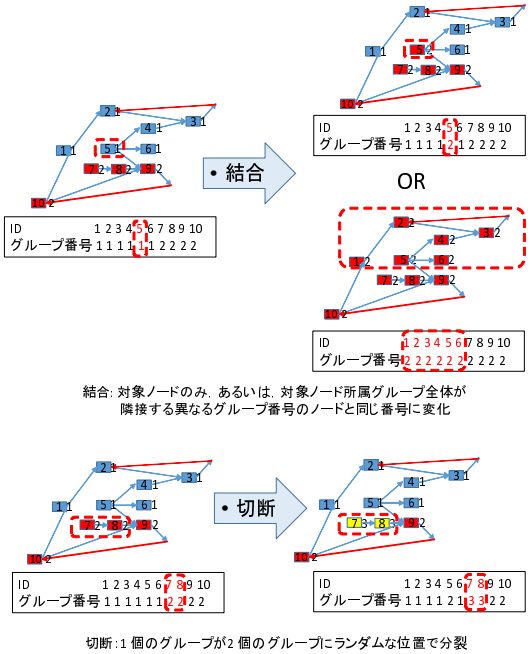


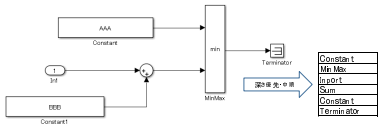

図 4: 突然変異

## 4. 隠れマルコフモデルによるモデルパターン検出

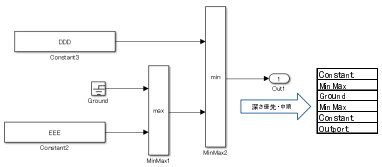
本研究では，simulink モデルのブロック構成を隠れマルコフモデルに学習させ，類似のブロック構成や同一機能で異なるブロック構成をした部分を機能的なクローンとして検出することを狙う．まず，検出対象と同一機能のブロック線図を複数学習サンプルとして用意し（図 5 参照），概ね対象の機能を検出できるよう手動で simulink モデル用の隠れマルコフモデルを作成する（図 6 参照）．このとき，確率の総和が 1 になるように正規化している．また，性能評価用のサンプルとして，上下限ガードの機能を実現する simulink モデルを用意した (図 9 参照)．

これらの学習サンプルを深さ優先の中順でブロックの列とし，隠れマルコフモデルに学習させる．学習には Baum らが提案した Baum-Welch アルゴリズムを用いた [Baum 1966][Welch 2003]．複数の学習サンプルを学習する際は，分母に足し合わせることで実現している．学習前の隠れマルコフモデルを図 6 に，学習後の隠れマルコフモデルを図 7 に示す．, 学習後については概要図を図 8 に示す．学習前は Divide, Product 等を別の状態とし，それぞれシンボル出力確率を定義していたが，学習後はそれらのシンボル出力確率が同一のシンボル出力確率となり，ほぼ区別がなくなっている．このように予め概ね対象の機能を検出できるよう手を動で設定した通りではない学習が起きるが，これは隠れマルコフモデルが学習サンプルに対して認識効率が良くなるように学習した結果であると考えられる．

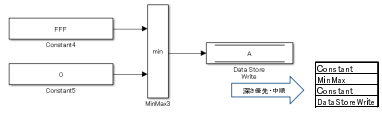
隠れマルコフモデルの尤度の計算には FORWARD アルゴリズムを用いた [Baum 1966][Welch 2003][村上 2010]．学習した隠れマルコフモデルから，検出対象の simulink モデルが出力される確率の総和を尤度として，FORWARD アルゴリズムを用いて計算すると，似ているほど高い尤度となる．このとき，simulink モデル内のどの部分を隠れマルコフモデルの入



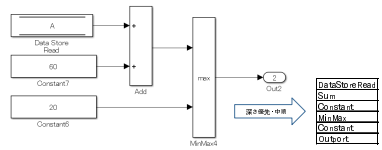

(a) 上下限ガード 1




(b) 上下限ガード 2




(c) 上下限ガード 3




(d) 上下限ガード 4

図 5: 学習サンプル simulink モデル

| | HE | Sum | Divide | Product | RO | SU | SO | MinMax |
|---|---|---|---|---|---|---|---|---|
| HE | 0.10 | 0.50 | 0.00 | 0.10 | 0.10 | 0.10 | 0.10 | 0.50 |
| Sum | 0.50 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
| Divide | 0.50 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
| Product | 0.50 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
| RO | 0.10 | 0.00 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
| SU | 0.10 | 0.10 | 0.00 | 0.10 | 0.10 | 0.10 | 0.00 | 0.00 |
| SO | 0.10 | 0.10 | 0.00 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
| MinMax | 0.10 | 0.10 | 0.00 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |

(a) 状態遷移確率表

| | Inport | Constant | DataStore Read | DataStore Write | Outport | Sum | Divide | Product | SumSub | RelationalOperator | Logic | SubsystemSize | SubSystem | Shift Arithmetic | DataType Conversion | UnitDelay | MinMax |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HE | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Sum | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Divide | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Product | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| RO | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| SO | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| MinMax | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.99999 |

(b) シンボル出力確率表

図 6: 上下限ガード学習前の隠れマルコフモデル

| | HE | Sum | Divide | Product | RO | SU | SO | MinMax |
|---|---|---|---|---|---|---|---|---|
| HE | 1.67E-08 | 2.04E-03 | 0.00E+00 | 4.58E-02 | 2.89E-02 | 5.43E-02 | 2.41E-01 | 6.20E-01 |
| Sum | 3.47E-07 | 7.35E-03 | 3.11E-03 | 3.11E-03 | 3.11E-03 | 3.11E-03 | 3.11E-03 | 9.42E-01 |
| Divide | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| Product | 6.61E-01 | 3.39E-01 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| RO | 1.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| SU | 2.80E-01 | 7.20E-01 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| SO | 2.80E-01 | 7.20E-01 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| MinMax | 6.01E-01 | 6.93E-02 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 3.33E-01 |

(a) 状態遷移確率表

| | Inport | Constant | DataStore Read | DataStore Write | Outport | Sum | Divide | Product | SumSub | RelationalOperator | Logic | SubsystemSize | SubSystem | Shift Arithmetic | DataType Conversion | UnitDelay | MinMax |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HE | 0.119608 | 0.760788 | 0.119606 | 0 | 0 | 2.88E-58 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Sum | 0 | 1 | 0 | 0 | 0 | 1.7E-22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Divide | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Product | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RO | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SO | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MinMax | 0 | 0 | 0 | 0 | 0 | 1.03E-04 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.999897 |

(b) シンボル出力確率表

図 7: 上下限ガード学習後の隠れマルコフモデル

力とするかは遺伝的アルゴリズムによるクラスタリングにより決定する．

適応度 $F_r$ は隠れマルコフモデルの尤度を用い，

$$F_r = a \times \sum_{i=0}^{N_g}(N_{sg} \times Y_i \times (-\frac{1}{1+\exp(a_d \times (g_i - (N_{gave} - 2)))}) + 1) \times (\frac{1}{1+\exp(b_d \times (g_i - (N_{gave} + 2)))}) + 1)) \quad (1)$$

で表す（大きいほど，より良いグループ分け）．$N_g$ は simulink モデルの要素数（ブロック数），添え字の $i$ は各要素番号，$g_i$ は $i$ 番目の要素のグループの要素数である． $a$ は調整パラメータであり，$a = 8$ とした．第 1 項目は基本的に各グループの尤度 $Y_i$ を合計した値となる．第 2 項目はグループの要素が，1 個，2 個など，極端に少なくなることによる，尤度の上昇を防止している．$N_{gave}$ は学習サンプルのブロック数の平均値である．この平均値から 2 を引いた値をグループ内の要素数が下回ると 0 に近づく値となる．隠れマルコフモデルは各要素の遷移確率 (0〜1 の値) を掛け合わせていくことで計算されるため，遷移する要素が少ないほど値は高く (1 に近く) なり，多いほど値は小さくなる．しかし，要素が極端に少ない場合，尤度は高くても，機能を判定しているとは言えないため，そのようなグループ構成による尤度の上昇を防ぐ必要がある．第 3 項目はグループの要素が極端に多くならないように，平均値に 2 を足した値まで緩やかに値が上昇している．$a_d$，$b_d$ はそのような場合の適応度の減少の度合いを調整するパラメータであり，$a_d = 1000$，$b_d = 0.5$ とした．

$N_{sg}$ は

$$N_{sg} = \prod_{j=0}^{N_{gnear}} 1.1 \times \prod_{j=0}^{N_{nonear}} 0 \quad (2)$$

で算出する．$N_{gnear}$ はグループ内の各要素に対して，結線で繋がる要素が同じグループであるかをカウントした値である．$N_{nonear}$ はグループ内の各要素に対して，結線で繋がっている要素に同じグループがない要素数である．添え字の $j$ は各要素番号である．これらにより，グループの要素同士が結線で繋がっているほど高い値となる．また，結線が繋がっていない要素同士がグループになった場合，$Nsg$ が 0 となり，適応度が大幅に減少することで，そのようなグループの形成を抑止している．

遺伝的アルゴリズムの遺伝子を simulink モデルのグルーピング情報とする．各グループの尤度が最も高くなるグルーピングを実現することで，検出対象と類似機能を持つブロック構成をグループとして検出することを狙う．

## 5. シミュレーション実験結果

提案手法をサンプルモデルの上下限ガードを含む simulink モデル (図 9 参照) に適用し，その性能を評価した．この性能評価用サンプルを深さ優先の中順でブロックの列とし，各グループ毎に FORWARD アルゴリズムにより，学習した隠れマルコフモデルの尤度を算出する．

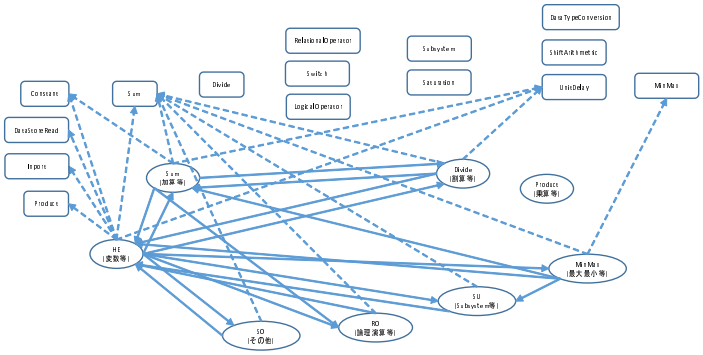



図 8: 学習後の隠れマルコフモデル概要図

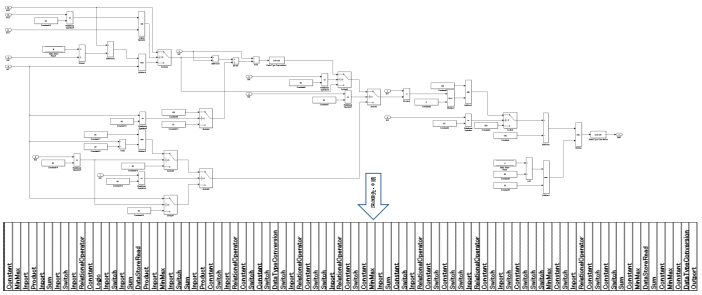


図 9: 上下限ガードを含む性能評価用 simulink モデル

個体プール，次世代個体プール共に格納しうる最大個数は 30 とした．突然変異率は 0.1 とし，交叉率は 0 とした．但し，突然変異操作の切断のみ確率を突然変異率× 0.02 とした．計算は，1 個体の評価を 1step とすると，200,000steps(6,666.66 世代) まで行った．

まず，上下限ガードのサンプルを学習させた隠れマルコフモデルを用いて，グルーピングしたサンプルモデルの結果を MATLAB/Simulink で図 10 に，その DSM 表現を図 11 に示す．尤度，適応度のどちらかが 0 でないグループはその数値を表記している．多くのグループが作成さているが，この中の適応度が 0 でないグループが検出されたモデルのクローン部分となる．②，③，④は概ね狙い通りに上下限ガード部分を検出できている．①は上下限の計算は行っているものの，上下限ガードは行っていないため，尤度は 1.92E-11 と低くなっている．⑤は尤度は要素数が 1 個と極端に少ない為，尤度は 1.0 と高い値となっているが，適応度は 0 となっており，狙い通り極端に少ない要素による高尤度のグループを抑制できていることが分かる．

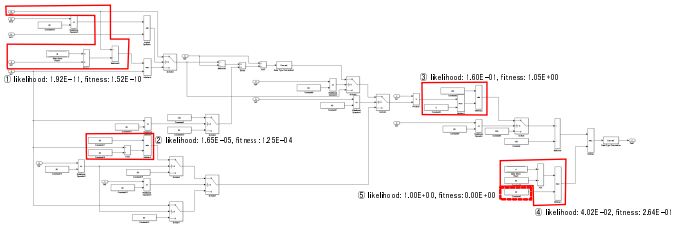



図 10: 提案手法によるグループ分け（simulink モデル）

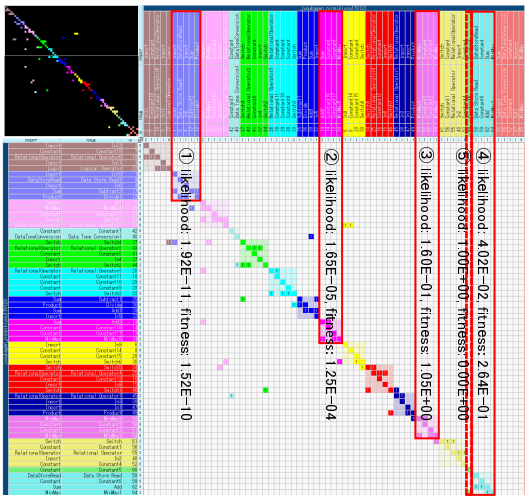



図 11: 提案手法によるグループ分け（DSM）

## 6. おわりに

近年，モデルベース開発が普及し，ソフトウェア開発として MATLAB/Simulink モデルを作ることも多いが，クローンと呼ばれる類似処理が増加すると，従来のソフトウェア開発と同様に保守が困難になる．

そこで，そのような類似のブロック構成や同一機能で異なるブロック構成をした部分を機能的なクローンとして検出することを狙った．その際，simulink モデルを深さ優先の中順で，木構造解析し，1 次元のブロック列とすることで，ブロック線図の比較を容易にした．具体的には，simulink モデル用の隠れマルコフモデルを作成し，検出対象と同一機能のブロック線図を複数学習サンプルとして用意し，それを学習させた．

さらに，simulink モデル内のどの部分を隠れマルコフモデルに入力するかを遺伝的アルゴリズムによるクラスタリングにより決定し，検出対象と類似機能を持つブロック構成をグループとして検出することができた．

## 参考文献

[Baum 1966] L. E. Baum, T. Petrie, “Statistical inference for probabilistic functtions of finite state Markov chains” *The Annals of Mathematical Statistics*, 37, No.6, pp.1554-1563, (1966).

[中井 2015] 中井淳一, 筑紫晴久, “手戻りを考慮した MATLAB/Simulink モデルのためのクラスタリング手法” 人工知能学会論文誌. 30, No. 6, pp.791-801 (2015).

[村上 2010] 村上仁一, “Baum-Welch アルゴリズムの動作と応用例” 電子情報通信学会. 基礎・境界ソサイエティ. 4, No. 1, pp.48-56 (2010).

[Steward 1981] D. V. Steward, “The Design Structure System: A Method for Managing the Design of Complex Systems” *IEEE Transactions on Engineering Management* S, 28, No. 3, pp.71-74 (1981).

[Welch 2003] L. R. Welch, ”Hidden Markov models and the Baum-Welch algorithm.” *IEEE Information Theory Society Newsletter* 53, No. 4, pp.10-13 (2003).