

# 対称性のある代謝産物を通る elementary flux mode 型代謝経路の完全原子レベルマッピング

太田潤<sup>†1,a)</sup>

**概要** : 代謝ネットワークにおける“経路”として, *in silico* atomic tracing により基質-生成物の二項関係を連ねたタイプの“経路”以外に, 化学量論的に釣り合った個々の反応の集合としての“経路”を考えることができる. これまで, 反応の集合としての“経路”を elementary flux mode 型経路 (EFM 型経路) と呼び, 通常は原子レベルで考えることのない EFM 型経路に原子レベルの情報を与える (原子レベルマッピングを行う) ためのアルゴリズムを報告してきた. 本稿では, 対称性のある代謝産物を通る EFM 型経路にも正しく原子レベルマッピングを行うことができるよう, これまでのアルゴリズムにおける代謝産物の対称性の処理を厳格化したものを報告する. このアルゴリズム厳格化の過程で, これまでのアルゴリズムではその経路に可能な複数の原子レベルマッピングのうちの限られたものしか算出できないような EFM 型経路があることが判明した. このような EFM 型経路に対して可能な複数の原子レベルマッピングをすべて算出するための方法も本稿では述べる.

**キーワード** : 代謝ネットワーク, isotopomer tracing, EMU tracing, elementary flux mode

## Complete atomic-level mapping of elementary flux mode-type metabolic pathways through metabolites with symmetry

JUN OHTA<sup>†1,a)</sup>

### 1. はじめに

生理活性物質の生成は, 代謝ネットワークの主要な機能であり, 生理活性物質の生成経路に関する知識・情報は, 医学・生物学的に重要である. これまで, 代謝ネットワークにおける経路 (代謝経路) として, 原料代謝産物に含まれる原子が基質 - 生成物の二項関係の連なりにより目的代謝産物に含まれる原子に移動する道筋としての“経路” [1-6] と化学量論的に釣り合った個々の反応の集合としての“経路” [7,8] という 2 種類の考え方の“経路”が研究されてきた. 前者の“経路”の算出を *in silico* atomic tracing と呼ぶことがある. また, 後者の“経路”の代表例は, elementary flux mode [7] と extreme pathway [8] であり, 本稿ではこれらの例を含む後者の“経路”を elementary flux mode 型経路 (EFM 型経路) と呼ぶ.

特定の目的代謝産物 B の生成原料として代謝産物 A が適しているか判断する際に, 通常, *in silico* atomic tracing により A の分子を構成する原子のうちより多くをより少ない数の反応で B までトレースできる linear な経路が代謝マップ上で検討されるが, このような A から B に至る linear な経路の存在は A が B の原料であるための必要条件であって十分条件ではない. 厳密には, 反応の集合である EFM 型経路の収支式の左辺に現れる原料分子群のすべてが揃った場合にのみ右辺の目的分子群が生成すると言える. EFM 型経路における原料分子群と目的分子群は化学量論的に釣り合っており, 基本的には 1 セットの原料分子群の各原子が

1 セットの目的分子群の各原子に取り込まれる原子レベルの道筋が存在する筈である. しかし, これまで, この原子レベルの道筋を EFM 型経路において考えることは一般に行われてこなかった.

私はこれまでの研究報告で, 与えられた原料 isotopomer から生成し得る isotopomer を列挙する isotopomer tracing, 与えられた原料 elementary metabolite unit (EMU) から生成し得る EMU を列挙する EMU tracing により, 与えられた原料代謝産物から目的代謝産物の分子構造全体が合成可能であるかどうかの判定ができることを記載してきた [9,10]. さらに [11] では, isotopomer tracing または EMU tracing に引き続く目的 entity (isotopomer または EMU) から原料 entity に向けての逆行性の経路計算により, 原料分子の各原子の使用回数を制限しない場合の目的分子の分子構造全体の生成経路が原子レベルで計算できることを述べた.

[11] の方法で得られる生成経路は entity と反応を node とし目的分子の分子構造全体に対応する entity を root とする木構造である. その後, この木構造の辺を root に向けて統合 (merge) することが可能な経路を選択してその経路の merge 後の原子の動きから EFM 型代謝経路の完全原子レベルマッピングを行う (EFM 型経路における原料代謝産物の全原子の目的代謝産物の原子への移動を求める) ための基本的なアルゴリズム (基本アルゴリズム) [12] と完全原子レベルマッピングがより短い時間で行えるようにした基本アルゴリズムの改良型 (高速アルゴリズム) [13] を報告した. それらは, ペントースリン酸経路の反応により 2 分子の fructose 6-phosphate と 1 分子の glyceraldehyde 3-phosphate

<sup>†1</sup> 岡山大学 大学院医歯薬学総合研究科 (医) 生化学分野  
a) jo25@md.okayama-u.ac.jp

から3分子の ribose 5-phosphate が生成する EFM 型経路の完全原子レベルマッピングに有効であった [12,13].

基本アルゴリズムと高速アルゴリズムにおいては entity と反応を node とする木構造の辺が merge できるかどうかの判定を共通の補助関数で行う. この補助関数を [13] では ismergeable と呼んだ. この補助関数 ismergeable は対称性のある (経路上の) 代謝産物に関する処理も考慮して作成したものであったが, 今回, 新たに対称性のある代謝産物に関する処理を厳格化した補助関数, 厳格型 ismergeable を作成した. 本稿では, 厳格型 ismergeable を報告するとともに, 高速アルゴリズムにおいて対称性のある代謝産物に関する従来の処理を行う ismergeable (非厳格型 ismergeable) と厳格型 ismergeable を用いた場合の違いを, 対称性のある代謝産物を通る EFM 型経路で検討した結果を報告する.

この検討の過程で, 既報の基本アルゴリズム [12] と高速アルゴリズム [13] のままではその経路に可能な複数の原子レベルマッピングのうち限られたものしか算出できないような EFM 型経路があることが判明した. このような EFM 型経路に対して可能な複数の原子レベルマッピングをすべて算出するための方法も本稿では述べる.

## 2. 方法

基本アルゴリズムと高速アルゴリズムのうち高速アルゴリズムを用いて行った. 本研究における高速アルゴリズムの手順を述べる. その中で, 非厳格型 ismergeable, 厳格型 ismergeable, EFM 型経路に対して可能な複数の原子レベルマッピングをすべて算出するために補助関数 route2source\_core に加えた小さな改変を述べる.

### 2.1 EFM 型経路の, 単一の仮想原料分子から単一の仮想目的分子に至る EFM 型経路への変換

完全原子レベルマッピングすべき与えられた EFM 型経路から, その構成反応の集合と原料代謝産物 source(s) と目的代謝産物 target(s) の情報を得る. 原料代謝産物と目的代謝産物は何れも複数の分子である場合がある. この原料代謝産物と目的代謝産物の情報を使い, 下記の化学量論的に釣り合った2つの仮想反応 source reaction, target reaction を導入・追加し. 与えられた EFM 型経路を単一の仮想原料分子 (combined\_source\_metabolite) から単一の仮想目的分子 (combined\_target\_metabolite) への経路に変換する.

source reaction: combined\_source\_metabolite → source(s)

target reaction: target(s) → combined\_target\_metabolite

### 2.2 Connectivity matrix (CM) による EFM 型経路内の代謝産物間の原子のつながりの記述

与えられた EFM 型経路構成反応 (source reaction, target reaction を含まない) と source reaction, target reaction に含まれる代謝産物, 反応に対して, 代謝産物番号, 反応番号を付す. 代謝産物の注目する原子 (炭素等) に, 代謝産物内原子番号を付し, 代謝産物番号 m と代謝産物内原子番号 c

の組 (m, c) により各原子を表現する. 計算上 (m, c) は複素数  $m + ci$  として扱う. 経路内で各反応により各代謝産物の各原子がどの代謝産物のどの原子に移るかの情報を connectivity matrix (CM) の形式で記述する [5,6].  $m_1$  と  $m_2$  が代謝産物を,  $(m_1, c_1)$  と  $(m_2, c_2)$  が原子を,  $p_1$  が反応を表現するとするとき, CM の行  $[(m_1, c_1), (m_2, c_2), p_1]$  は, 代謝産物  $m_1$  中の原子  $(m_1, c_1)$  が反応  $p_1$  により別の代謝産物  $m_2$  中の原子  $(m_2, c_2)$  に変換されることを意味する. 多数の反応を含む経路内のすべての原子のつながりが単一の CM として表現できる. 経路に同じ反応が複数回現れる場合があり得るが, 各反応による原子のつながりは反応が複数回現れても経路全体に対する CM に1回だけ記述する. 本稿では, 与えられた EFM 型経路構成反応と source reaction による原子のつながりを表現する CM を work\_CM, 与えられた EFM 型経路構成反応と source reaction, target reaction による原子のつながりを表現する CM を work\_CM\_all と呼ぶ.

### 2.3 代謝産物番号の補足インデックス番号, 補足インデックス行列, 反応内代謝産物 ID ベクトル [9,10,11]

補足インデックス番号 subsidiary index (sub-index) により, 反応式の片側に現れる同じ代謝産物の2個以上の分子それぞれを区別する. 反応 p の反応式で代謝産物 m の係数が n の場合, すなわち反応式に代謝産物 m 分子が n 個現れる場合, n が2以上ならば n 個の代謝産物 m 分子に1から始まる通し番号の sub-index を付す. n が1ならば1個しかない代謝産物 m 分子の sub-index は1とする.

各反応において, 代謝産物中の原子にも sub-index を与え, その値は代謝産物の sub-index と同じ値とする. CM の1列目と2列目の各成分に対応する原子の sub-index が必要に応じて参照できるように CM と同じ行数を持つ2列の行列である sub-index-CM を作成し, CM の i 行 j 列の成分に対応する sub-index の値を sub-index-CM の i 行 j 列の成分として記述する. sub-index-CM を CM の補足インデックス行列と呼ぶ.

また, 代謝産物番号 m と sub-index (s\_idx) からなる行ベクトル  $[m, s\_idx]$  を反応内代謝産物 ID ベクトルと呼び, このベクトルにより反応式に現れる特定の代謝産物分子を示す. このベクトルは connection (後述) の算出に用いる.

### 2.4 Entity (isotopomer と EMU) の番号付け

ある代謝産物 M 分子の c 番の炭素が, 自然界に最も多い  $^{12}\text{C}$  以外の炭素同位体である場合, M 分子の c 番の炭素がその同位体により label されていると言う. M 分子の各炭素の label の状態が特定の pattern を示す M の分子種は一般に M の isotopomer である. M の炭素に関する isotopomer の label の位置は炭素数と同じ桁数をもつ2進数の1の位置で表現される [14]. 本稿では M の isotopomer の isotopomer 番号 (isotopomer に対する entity 番号) を, 上記の label の位置を示す2進数を10進数に変換した値に1を加えたもの

と定義する [9]。炭素以外の原子あるいは全原子に関する isotopomer とその isotopomer 番号による表現をすることも可能である。

EMU は分子の原子レベルの構造単位である [15]。代謝産物 M の (全く label されていない isotopomer 以外の) 1 個以上の原子が label されている isotopomer のそれぞれに対して, label された位置の原子から構成される M 分子の EMU を考えることができる。EMU に含まれている原子の位置は上述の isotopomer の場合と同様に注目している原子の数と同じ桁数をもつ 2 進数の 1 の位置で表現される。M 分子の EMU の EMU 番号 (EMU に対する entity 番号) を, 上記の M 分子上の原子の位置を示す 2 進数を 10 進数に変換した値と定義する [10]。EMU 番号は, 同じ原子位置に対応する isotopomer の isotopomer 番号より 1 だけ小さい。分子内の注目する原子 (の原子位置) に連続的に番号付けすれば任意の原子からなる任意の EMU を EMU 番号により表現できる [10]。

非対称な分子では entity が isotopomer と EMU の何れであっても entity 番号と entity の対応が一対一対応であるが, 対称性がある分子では, entity 番号の異なる entity どうしが区別できない場合がある。この場合, 該当する entity を示すために最も小さい entity 番号を用いる。本稿では entity が isotopomer と EMU の何れであっても代謝産物 M の entity を M の代謝産物番号 m と entity 番号 t の組み合わせ index である (m, t) で表現する。計算上 (m, t) は複素数  $m + ti$  として扱う。

## 2.5 Isotopomer のつながりとその isotopomer connectivity matrix (ICM) による記述 [9]

$p_1$  を反応番号,  $m_1$  と  $m_2$  を代謝産物番号とする。反応  $p_1$  により反応  $p_1$  の基質  $m_1$  の一部が反応  $p_1$  の生成物  $m_2$  の一部になるとき, 一部の  $m_1$  の原子と一部の  $m_2$  の原子の間には反応  $p_1$  による一対一の対応関係がある。この,  $m_1$  と  $m_2$  の, 対応関係にある部分全体の labeling pattern を考える。この labeling pattern が  $m_1$  と  $m_2$  の間で完全に一致するとき, その labeling pattern を与える  $m_1$  の isotopomer ( $m_1, t_1$ ) と  $m_2$  の isotopomer ( $m_2, t_2$ ) は前者から後者に向けて反応  $p_1$  によりつながっているとす。このつながりを行ベクトル  $[(m_1, t_1), (m_2, t_2), p_1]$  で表現し, 一般的にはこの行ベクトルを積み重ねて isotopomer connectivity matrix (ICM) を構成する。“ $(m_1, t_1)$  が含む label された原子すべてが  $(m_2, t_2)$  に入る” という条件を満たすつながりの行  $[(m_1, t_1), (m_2, t_2), p_1]$  のみからなる ICM を fundamental ICM と呼ぶ [12]。

## 2.6 EMU のつながりの EMU connectivity matrix (EMUCM) による記述 [10]

$p_1$  を反応番号,  $ms_1, ms_2, \dots, ms_n$  (“ $ms_1$  から  $ms_n$ ”) と mp を代謝産物番号とする。反応  $p_1$  により反応  $p_1$  の基質 “ $ms_1$  から  $ms_n$ ” を原料として反応  $p_1$  の生成物 mp が生成するとき, “ $ms_1$  から  $ms_n$ ” の原子の少なくとも一部と mp

の原子の間には反応  $p_1$  による一対一の対応関係がある。したがって mp の任意の EMU (mp, t) に対し, 反応  $p_1$  を介して EMU (mp, t) に含まれる原子と一対一対応する原子を有する代謝産物 (基質) が “ $ms_1$  から  $ms_n$ ” に必ず存在する。このような代謝産物 (基質) の 1 つを  $ms_i$  とし,  $ms_i$  の EMU であって mp の EMU (mp, t) の原子と一対一の対応関係にある原子をすべて含むが対応関係にない原子は含まない EMU を  $(ms_i, t_i)$  と表現する。このとき, EMU  $(ms_i, t_i)$  が EMU (mp, t) に向けて反応  $p_1$  によりつながっていると定義し, このつながりを, 行ベクトル  $[(ms_i, t_i), (mp, t), p_1]$  で表現する。このつながりの定義は, EMU reaction [15] において左辺の EMU が右辺の EMU につながっているとみなすことに対応する。経路内のすべての EMU のつながりを行ベクトルで表現し (=すべての反応のすべての生成物のすべての EMU について  $(ms_i, t_i)$  に相当するものすべてを求めて行ベクトルを生成し), 得られた行ベクトルを積み重ねて EMU connectivity matrix (EMUCM) を構成する。

## 2.7 sub-index-ICM と sub-index-EMUCM [9,10]

各反応の基質・生成物の isotopomer と EMU に, 基質・生成物の sub-index と同じ値の sub-index を与える。ICM と EMUCM の 1 列目と 2 列目の各成分に対応する sub-index が必要な場合に参照できるように, それぞれが ICM と EMUCM と同じ行数を持つ 2 列の行列である sub-index-ICM と sub-index-EMUCM を sub-index-CM と同様に作成し sub-index を保持させる。sub-index-ICM, sub-index-EMUCM をそれぞれ ICM, EMUCM の補足インデックス行列と呼ぶ。

## 2.8 Isotopomer connection と EMU connection の算出

$A_j$  ( $j$  は自然数) と B を任意の entity (ここでは, 代謝産物, isotopomer, EMU 等の molecular entity 何れでもよい), F を作用 (反応等) とし,  $a_j, b, f$  をそれぞれ  $A_j, B, F$  を示す index とする。 $A_1$  から  $A_n$  のすべてが揃って初めて作用 F の下で B が生成・生起するとき, この B の生成・生起を  $A_1, A_2, \dots, A_n$  から B への F を介する n 個のつながりが一体化したものとみて connection と呼び,  $[a_1, b, f], [a_2, b, f], \dots, [a_n, b, f]$  の行ベクトル n 個を重ねた connection 行列で表現する。 $A_1$  から  $A_n$  を基質 entity, B を生成物 entity と呼ぶ。 $A_j$  と B が isotopomer である場合 connection を isotopomer connection, EMU である場合 EMU connection と呼ぶ [11]。

経路全体に対する ICM と sub-index-ICM から, 経路に存在するすべての isotopomer connection が ICM の部分行列として得られる [11]。Fundamental ICM から得られる isotopomer connection は, 基質側の isotopomer の label された原子数の合計が生成 isotopomer の label された原子数と等しい fundamental isotopomer connection [11] である。高速アルゴリズムのための isotopomer tracing には, 与えられた EFM 型経路構成反応と source reaction に対応する ICM と sub-index-ICM から算出される fundamental isotopomer

connection を用いる。

EMUCM と sub-index-EMUCM から EMU connection が算出できる [11]。高速アルゴリズムのための EMU tracing には、与えられた EFM 型経路構成反応と source reaction に対応する EMUCM と sub-index-EMUCM から算出される EMU connection を用いる。

Isotopomer connection, EMU connection を求める際には同時に connection に対応する補足インデックス行列を求める。

すべての原子が label された combined\_target\_metabolite の isotopomer を target reaction により生成する fundamental isotopomer connection の connection 行列を aICM, 対応する補足インデックス行列を aidxICM と呼ぶ。同様に, combined\_target\_metabolite 分子全体に対応する EMU を target reaction により生成する EMU connection の connection 行列を aEMUCM, 対応する補足インデックス行列を aidxEMUCM と呼ぶ。

## 2.9 Isotopomer tracing と EMU tracing

(以下で entity は isotopomer と EMU の何れかを指す。)

Isotopomer tracing ・ EMU tracing [12] では、最初に source (原料 entity の配列), connection の集合, connection の補足インデックス行列の集合を与える。そして, source 内の entity のみを利用して 1 段階の connection により生成する entity を source に要素として追加して source を更新し, その entity の生成 connection と connection の補足インデックス行列をそれぞれ配列 RC と RC\_idx に格納する。次いで更新された source 内の entity のみを利用して 1 段階の未使用 connection により生成する entity を source に追加して source を再更新し, その entity の生成 connection と connection の補足インデックス行列をそれぞれ配列 RC と RC\_idx に追加・格納する。これを source が更新されなくなるまで繰り返し, 最終的に得られる source を配列 reached とする。

高速アルゴリズムのための各 tracing 用の connection とその補足インデックス行列は, 与えられた EFM 型経路構成反応と source reaction に対する fundamental ICM と sub-index-ICM の組あるいは EMUCM と sub-index-EMUCM の組から算出されるものを用いる。用いる connection とその補足インデックス行列に target reaction に対する connection (aICM ・ aEMUCM に対応する connection) とその補足インデックス行列は加えない。

最初に与える配列 source (initial\_source) は isotopomer tracing では, label されていない仮想原料代謝産物以外の isotopomer と fundamental ICM の 1 列目に現れる仮想原料代謝産物 combined\_source\_metabolite の isotopomer とを要素とする配列, EMU tracing では, EMUCM の 1 列目に現れる仮想原料代謝産物 combined\_source\_metabolite の EMU を要素とする配列とする。

両 tracing の何れでも 3 つの配列 reached, RC, RC\_idx が得られる。reached は, 与えられた反応の connection を利用

して原料 entity から生成し得るすべての entity を含む。RC は reached の要素を直接生成する connection を, RC\_idx は RC 内の connection の補足インデックス行列を格納する。connection RC[j]により reached[j]に格納された entity が生成する。RC の要素は重複しないが, reached の要素は重複する場合がある。RC[j]の補足インデックス行列は RC\_idx[j]に格納する。ここで, “A[j]”により配列 A の j 番目の要素を表す記法を用いた。

## 2.10 高速アルゴリズムによる EFM 型経路の完全原子レベルマッピングに用いる情報

前段階の tracing で用いる initial\_source ; tracing で得られる RC ・ RC\_idx ・ reached ; combined\_source\_metabolite の代謝産物番号 csm ; combined\_source\_metabolite 中のマッピングされるべき原子の個数 csm\_an ; entity が isotopomer か EMU かに応じて aICM または aEMUCM を表す ameCM ; ameCM の補足インデックス行列 aidxmeCM ; 分子の対称性情報 symmetry\_info ; 与えられた EFM 型経路構成反応と source reaction と target reaction による原子のつながりを表現する CM である work\_CM\_all ; work\_CM\_all の補足インデックス行列 work\_idxCM\_all ; unique\_target\_metabolite ; number\_of\_unique\_target\_metabolite が必要である。末尾の 2 者は, 目的代謝産物 target(s)が代謝産物番号 4 の化合物が 3 分子と代謝産物番号 5 の化合物が 2 分子であって, 代謝産物番号 4 の化合物の分子全体に対応する entity が (4, e<sub>4</sub>), 代謝産物番号 5 の化合物のそれが(5, e<sub>5</sub>) である場合,

unique\_target\_metabolite = 配列{(4, e<sub>4</sub>), (5, e<sub>5</sub>)}

number\_of\_unique\_target\_metabolite = 配列{3, 2}

であり, 後者が前者の分子の個数を示す。

## 2.11 高速アルゴリズムによる EFM 型経路の完全原子レベルマッピングに用いる補助関数

高速アルゴリズムで使う補助関数には, route2source\_core, count\_source\_atom, combine\_connection\_tree, ismergeable, initial\_ordered\_comb と next\_ordered\_comb がある [13]。ここでは, route2source\_core, ismergeable として非厳格型 ismergeable と厳格型 ismergeable を説明する。本項目でも “A[j]”により配列 A の j 番目の要素を表す記法を用いる。

### 2.11.1 route2source\_core

tracing で得られる配列 reached と RC, tracing に用いる原料 entity の配列 initial\_source を使い, entity である target が initial\_source に含まれる原料 entity から生成する経路を算出する [11]。経路の計算は target の entity から原料 entity に向けて遡りながら行う。以下の P0, P, X は 1 次元配列表記された経路を要素として保持・保存するための配列である。まず target を生成する connection の各 connection 番号 (経路終端に相当) を P0 に別々の要素として格納する。P0 に格納された経路を initial\_source に向けて 1 段階, 逆延長する。経路に n 個の connection が合流していれば, 1 段階で n 個の connection 番号がその経路の表記に加わる。得

られる経路は空配列 P に 1 次元配列として順次格納し、P 内の経路で端がすべて initial\_source に達したものを完成経路として配列 X に移す。この後の未完成経路のみを含む P を新たな P0 とし、次段階の逆延長を行う。この逆延長を、配列 P0 が空配列になるまで繰り返す。

原法 [11,13] では、逆延長を、原料 entity から target に至る linear な道筋上に同一 entity が 2 回以上出現しないように行っていたが、allowed\_appearance\_times に指定した回数だけ同一 entity の出現を許容するように改変した。この改変が、“原法によっては、可能な原子レベルマッピングのすべてを産出できない” EFM 型経路の可能な複数の原子レベルマッピングすべてを算出することと関係する。

経路計算により、経路を示す 1 次元配列が経路数だけ得られ X に格納される。各 1 次元配列 (= 経路) に connection を node とする木構造 (connection 木構造) が対応する。connection 木構造では、子 node である connection の生成物 entity が親 node である connection の基質 entity となっている。木の root の node である connection は、target の entity を生成物 entity とする connection である。

### 2.11.2 非厳格型 ismergeable

combined\_source\_metabolite の代謝産物番号 csm, connection 木構造と対応する配列 Zj, work\_RC (配列 RC に ameCM を要素として追加したもの), work\_RC\_idx (配列 RC\_idx に aidxmeCM を要素として追加したもの), 前述の work\_CM\_all, work\_idxCM\_all, symmetry\_info (sym\_metabolite\_no\_list, sym\_pair\_M\_cell) を与えたとき、Zj が “merge” [12] に関して merge が可能な場合 assembled = 1 を、不可能な場合 assembled = 0 を算出する。同時に、Zj で利用される combined\_source\_metabolite の原子の個数を原子位置ごとに示す行ベクトルを atom\_count として算出する。sym\_metabolite\_no\_list は対称性のある代謝産物の代謝産物番号からなる配列である。sym\_pair\_M\_cell は配列であり sym\_pair\_M\_cell[j] は sym\_metabolite\_no\_list[j] と一致する代謝産物番号の代謝産物の対称性を記述する行列 M を格納する。M の行数はその代謝産物の原子の個数 n に一致する。M の各列は 1 から n の n 個の整数を 1 つずつ使った配列である。M の任意の 2 列について各列の要素からなる配列を c<sub>1</sub>, c<sub>2</sub> とすると、n 以下の任意の自然数 m に対して、その代謝産物の原子 c<sub>1</sub>[m] は原子 c<sub>2</sub>[m] と対称である。

Zj は “merge 可能な経路算出のためのアルゴリズム” [13] の中で combined\_source\_metabolite から目的代謝産物 target(s) を経て combined\_target\_metabolite が生成する経路あるいはその一部として現れる。Zj の connection 木構造においては、各 connection 自身が entity と反応を node とする木構造を持つ。このことを基に connection 木構造全体を entity と反応を node とする木構造に変換できる。この変換後の木構造の leaf には必ず combined\_source\_metabolite の entity が現れる。combined\_source\_metabolite の entity であ

る leaf すべての合体から始めて、木構造の辺を combined\_target\_metabolite の entity である root に向けて統合していくことができる場合がある [12]。この統合のことを merge と呼ぶ。merge は EFM 型経路の完全原子レベルマッピングのための重要な段階である。merge には複数の node を合体させて新しい node をつくるのが伴う。複数の node を合体させることを node の assemble と呼ぶ。merge されると、connection 木構造で複数の connection によって表現されていた原子の移動が 1 つにまとめられる。merge 可能・不可能の判定は [12] で述べた手順で行う。combined\_source\_metabolite の entity である leaf の node を含む、ある代謝産物の entity に対応する複数の node が 1 つの node に assemble できるための条件は、① “node content の entity により指定される原子の位置がそれらの node 間で重複していないこと” と② “merge 前の木構造においてそれらの node の親 node の node content の反応がすべて同じであること” である。反応に対応する複数の node については、③ “1 つの node に assemble できると①②により判定された (一群の) entity に対応する node の親側にある反応の node は、node content (反応) が同じであり assemble できる”。

[12] の 5 頁の Step 11 に node の entity の代謝産物に対称性がある場合の上記①に関する処理について、“node の entity の代謝産物に対称性がある場合、複数の異なる bit 表示 (=2 進数表示) の (bit 表示で異なる原子位置に 1 がある) entity が化学的に同等である場合があるので、assemble できないと判定するためには、すべての bit 表示の原子位置での assemble を試みる必要がある。”と記載している。非厳格型 ismergeable では、対称性がある代謝産物の node が assemble できるかどうかの判定をこの基準に基づき Step 5 で行う。以下、非厳格型 ismergeable の手順を示す。計算の過程では assembled と atom\_count に加えて CM\_s (assemble 後の node のつながりを記述する), node\_list (assemble 後の node の情報を記述する) [12], node\_relation (assemble により生成した node が assemble 前のどの node に由来するかを示す), cAM (node の entity の各原子が combined\_target\_metabolite のどの原子へトレースされるかを示す) も算出される。

**Step 1.** Zj, work\_RC, work\_RC\_idx から行列 CM\_s\_before と行列 node\_list\_before を算出する。Zj の connection 木構造全体を entity と反応を node とする木構造に変換した木構造において、CM\_s\_before はその node 間のつながりを、node\_list\_before は node 情報を示す。CM\_s\_before と node\_list\_before はそれぞれ CM\_s と node\_list の形式 [12] に従う。CM\_s と node\_list の形式 [12] では、CM\_s は 1 列目を子の、2 列目を親の node number とする隣接リスト行列であり、node\_list の各行は 1 つの node に対する node number (=行番号), node type (node が entity のとき 1, 反応のとき 2), node content (entity を示す複素数または反応

を示す反応番号), sub-index1 と sub-index2 (node が反応の場合は両方 0, node が entity の場合は, その node を受け取る connection におけるその node に相当する基質 entity の sub-index が sub-index1, その node を与える connection のその node に相当する生成物 entity の sub-index が sub-index2) の情報を格納する.

**Step 2.** CM\_s\_before, node\_list\_before, work\_CM\_all, work\_idxCM\_all, sym\_metabolite\_no\_list, sym\_pair\_M\_cell から cAM を算出する. cAM は配列であり, cAM[j] は node number j の node に関して node が反応の場合は空行列を, node が entity の場合は 2 列の行列 AM を格納する. AM の 1 列目は combined\_target\_metabolite から逆トレースできる node の entity が所属する代謝産物内の原子の番号であり, 各行の 2 列目はその行の 1 列目の原子まで逆トレースできる combined\_target\_metabolite 内の原子の番号を示す.

**Step 3 (merge\_route\_tree).** csm, CM\_s\_before, node\_list\_before から CM\_s, node\_list\_tentative, number\_of\_nodes\_initial と node\_relation (assemble により生成した node が assemble 前のどの node に由来するかを示す) を算出する. number\_of\_nodes\_initial は node\_list\_before の行数として算出する. CM\_s と node\_list にそれぞれ CM\_s\_before と node\_list\_before を代入して [12] の 5 から 6 頁の Step 1 から 14 のうちの Step 11 以外を順番に行うと, CM\_s (assemble 後の node のつながりを記述する) が CM\_s として, node\_list\_tentative (assemble 後の node の情報を記述する) が node\_list として算出される. node\_list\_tentative においては, assemble により生じた node の node type は assemble 前と同じ値に, node content は assemble 前の node content の実部の値に設定される. また, assemble により CM\_s から除かれた node の node type は元の値に-1 を乗じた値に設定される. node\_relation は配列であり node\_relation[j] は number\_of\_nodes\_initial を基準に j がそれより大きい場合は node number j の node の assemble 前の node の node number すべてからなる配列, j がそれ以下の場合は j とする. merge できないと判定された場合には assembled=0 とし, この段階で停止・終了する.

**Step 4.** csm, node\_list\_before, work\_CM\_all, work\_idxCM\_all, sym\_metabolite\_no\_list, sym\_pair\_M\_cell から atom\_count を算出する. atom\_count に 1 より大きな成分がある場合は, assembled=0 とし, この段階で停止・終了する.

**Step 5.** node\_relation, number\_of\_nodes\_initial, node\_list\_tentative, cAM, sym\_metabolite\_no\_list, sym\_pair\_M\_cell, work\_CM\_all, work\_idxCM\_all から node\_relation と要素数が同じ配列 cAM\_combined を算出する. node number j の node が反応の場合 cAM\_combined[j] は空行列を格納する. node number j が entity の場合 cAM\_combined[j] は number\_of\_nodes\_initial を基準に, j がそれ以下であれば, cAM[j] の AM を格納し, j がそれより

大きければ node\_relation[j] に含まれる node の assemble により可能な (原子位置が重複しない) AM のうち最初に見つかったものを格納する. node number j が entity であって, cAM\_combined[j] に格納されるべき AM が存在しないような j が存在しない場合は, merge 可能であると判断して assembled=1 を算出し, 得られた cAM\_combined を assemble 後の cAM とするが, そのような j が存在する場合は, assembled=0 とし, この段階で停止・終了する.

**Step 6.** node\_list\_tentative の assemble 後の node の node content (虚部が 0) に cAM の情報に基づき虚部を与えることにより実際の entity を示す node content とし, node content に虚部を与えた後の node\_list\_tentative を node\_list とする. cAM は次項の原子レベルマッピングにも使用する.

### 2.11.3 厳格型 ismergeable

csm, Zj, work\_RC, work\_RC\_idx, work\_CM\_all, work\_idxCM\_all, symmetry\_info (sym\_metabolite\_no\_list, sym\_pair\_M\_cell, meCM) を与えると assembled, atom\_count を算出する. 計算の過程では CM\_s, node\_list, node\_relation, cAM も算出される. 5 列の行列である meCM は, entity が isotopomer か EMU かに応じて 1 列目から 3 列目が ICM または EMUCM に, 4 列目と 5 列目が (1 列目から 3 列目に対する) 補足インデックス行列 idxICM または idxEMUCM に一致するような 5 列の行列から, 対称性がある代謝産物の entity が 1 列と 2 列の少なくとも一方に現れる行すべてを抜き出し積み重ねて得られる行列である. 以下に, 厳格型 ismergeable の手順を示す.

**Step 1 から 4.** 非厳格型 ismergeable の Step 1 から 4 と全く同じ処理を行う.

**Step 5.** node\_relation, number\_of\_nodes\_initial, node\_list\_tentative, cAM, sym\_metabolite\_no\_list, sym\_pair\_M\_cell, work\_CM\_all, work\_idxCM\_all から node\_relation と要素数が同じ配列 cAM\_combined を算出する. cAM\_combined[j] に格納される cAM\_combined の要素は node number j の node に対応する. node number j の node が反応の場合 cAM\_combined[j] は空行列を格納する. node number j が entity の場合 cAM\_combined[j] は, number\_of\_nodes\_initial を基準に j がそれ以下であれば, cAM[j] の AM をただ 1 つの要素とする配列を格納し, j がそれよりも大きければ node\_relation[j] に含まれる node の assemble により可能な (原子位置が重複しない) すべての AM を要素として含む配列を格納する. node number j が entity であって, cAM\_combined[j] の配列に格納されるべき AM が存在しないような j が存在する場合は, assembled=0 となり, この段階で停止・終了する.

**Step 6.** node type=1 のすべての (entity の) node それぞれに, その node に対応する cAM\_combined の要素である (AM の) 配列から 1 つの AM を選び割り当て node と AM の対応の組み合わせ (AM の組み合わせ) を作る. 得られる組

み合わせの中に、可能な（経路での原子の動きに対応する）ものがあるか調べる。もしあれば、cAMを、組み合わせに含まれるAMを要素とする新たなcAMで置き換える。

AMの組み合わせが実際に可能なのは、組み合わせ中のAMにより指定されるentity (entity1)のnodeがCM<sub>s</sub>に指定される通りに反応のnodeを介して（組み合わせ中のAMにより指定される）別のentity (entity2)のnodeとつながっている場合に限られる。entity1-反応-entity2のつながりは、entity1とentity2の代謝産物の両方が対称性のないものである場合には可能である（存在し得る）ことが明らかなので、entity1とentity2の2代謝産物の少なくとも一方に対称性のあるentity1-反応-entity2のつながりについて、それぞれ可能である（存在し得る）かどうかを、少なくとも一方が対称性のある代謝産物のentityである2entity間の反応を介するつながりの情報をすべて含むmeCMを参照して調べる。meCMの4列目はentity1のnodeのsub-index1に対応し、5列目はentity2のnodeのsub-index2に対応する。

AMの特定の組み合わせに含まれるentity1-反応-entity2のつながりであってentity1とentity2の少なくとも一方が対称性のある代謝産物のentityであるものすべてが可能である（存在し得る）ことが確認できればmerge可能であると判断してassembled=1を算出し、そのAMの組み合わせをassemble後のcAMとする。“AMのその組み合わせに含まれるentity1-反応-entity2のつながりであってentity1とentity2の少なくとも一方が対称性のある代謝産物のentityであるものすべてが可能である（存在し得る）ことが確認できる”という条件を満たすAMの組み合わせが存在しなければmergeできないと判断してassembled=0とし、この段階で停止・終了する。

**Step 7.** node\_list\_tentativeのassemble後のnodeのnode content（虚部が0）にcAMの情報に基づき虚部を与えることにより実際のentityを示すnode contentとし、node contentに虚部を与えた後のnode\_list\_tentativeをnode\_listとする。cAMは次項の原子レベルマッピングにも使用する。

### 2.12 EFM型経路の完全原子レベルマッピング

2.10に挙げた情報および2.11に記した補助関数を用いて、“merge可能な経路算出のためのアルゴリズム”[13]に従い計算する。算出されるZはcombined\_source\_metaboliteからcombined\_target\_metaboliteに至るmerge可能な経路を要素とする配列であり、Zの要素である経路それぞれから、経路に対応する2種類の木構造（connection木構造およびentityと反応をnodeとする木構造）を経てmergeされた経路を得る。ismergeableで得られるcAMに格納されたAMが保持する各entityの原子とcombined\_target\_metaboliteの原子の間の対応関係の情報を基に、mergeされた経路に原子レベルの情報を付与してcombined\_source\_metaboliteからcombined\_target\_metaboliteへのEFM型経路の完全原子レベルマッピングを行う。

## 3. 計算実験の結果と考察

クエン酸回路の反応により1分子のacetyl-CoAから2分子の二酸化炭素が生成するEFM型経路の完全原子レベルマッピングを非厳格型ismergeableまたは厳格型ismergeableを用いるallowed\_appearance\_times=1の高速アルゴリズムで行うと、非厳格型ismergeableを用いた場合は4種類の、厳格型ismergeableを用いた場合は2種類の原子レベルマッピングが得られた。このマッピングの種類の数の違いはクエン酸回路上の対称性を持つ代謝産物succinateとfumarateの処理にあると考えて精査すると、非厳格型ismergeableを用いて得られた4種類のマッピングのうち2種類が、fumarate分子内の隣り合った2原子がmalate分子内の離れた2原子へ移動する不合理なマッピングであり、厳格型ismergeableを用いたマッピングでは、これら2種類の不合理な原子レベルマッピングが除外されていた。厳格型ismergeableを用いることにより、対称性のある代謝産物を通る経路が正しく原子レベルマッピングできたと考えられた。

次に、可能な原子レベルマッピングが厳格型ismergeableを用いる高速アルゴリズムで得られる2種類以外にないか確認する目的で、クエン酸回路の反応の逆反応により2分子の二酸化炭素から1分子のacetyl-CoAが生成する仮想のEFM型経路の原子レベルマッピングを、厳格型ismergeableを用いるallowed\_appearance\_times=1の高速アルゴリズムで算出し、それをacetyl-CoAから二酸化炭素に至る方向に読み替えることにより、1分子のacetyl-CoAから2分子の二酸化炭素が生成するEFM型経路の完全原子レベルマッピングを求めた。得られたマッピングは4種類（原料代謝産物を二酸化炭素2分子としたことに対応して、各種類のマッピングが2つずつ）であり、厳格型ismergeableを用いるallowed\_appearance\_times=1の高速アルゴリズムで1分子のacetyl-CoAから2分子の二酸化炭素が生成するEFM型経路を直接、完全原子レベルマッピングした場合の2種類と一致しなかった。そこで、厳格型ismergeableを用いるallowed\_appearance\_times=2の高速アルゴリズムで1分子のacetyl-CoAから2分子の二酸化炭素が生成するEFM型経路の直接の完全原子レベルマッピングを行ったところ、4種類のマッピングが得られ、何れもmerge後のentityに関してacetyl-CoAのentityから二酸化炭素のentityに至る経路上にentityの重複はなかった。EFM型経路内に原子レベルのcycleが存在する（ある原子から出発してEFM型経路内の原子レベルのつながりを辿ってその原子自身に到達できる）EFM型経路においてはallowed\_appearance\_times=1の条件で特定の原子レベルマッピングが算出されないということが起こり得るが、allowed\_appearance\_timesの値を調整することによりすべての原子レベルマッピングが算出できると考えられる。

## 参考文献

- [1] Arita, M. *In silico* atomic tracing by substrate-product relationship in *Escherichia coli* intermediary metabolism. *Genome Res.* **2003**, *13*, 2455–2466. DOI: 10.1101/gr.1212003
- [2] Latendresse, M.; Krummenacker, M.; Karp, P.D. Optimal metabolic route search based on atom mappings. *Bioinformatics* **2014**, *30*, 2043–2050. DOI: 10.1093/bioinformatics/btu150
- [3] Heath, A.P.; Bennett, G.N.; Kavraki, L.E. Finding metabolic pathways using atom tracking. *Bioinformatics* **2010**, *26*, 1548–1555. DOI: 10.1093/bioinformatics/btq223
- [4] Pitkänen, E.; Jouhten, P.; Rousu, J. Inferring branching pathways in genome-scale metabolic networks. *BMC Syst. Biol.* **2009**, *3*, 1–22. DOI: 10.1186/1752-0509-3-103
- [5] Ohta, J. Connectivity matrix method for analyses of biological networks and its application to atom-level analysis of a model network of carbohydrate metabolism. *IEE Proc. Syst. Biol.* **2006**, *153*, 372–374. DOI: 10.1049/ip-syb:20060018
- [6] Ohta, J. Single-atom tracing in a model network of carbohydrate metabolism and pathway selection. *IPSJ Transactions on Bioinformatics* **2018**, *11*, 1–13.
- [7] Schuster, S.; Hilgetag, C. On elementary flux modes in biochemical reaction systems at steady state. *J. Biol. Syst.* **1994**, *2*, 165–182.
- [8] Schilling, C. H.; Palsson, B.O. The underlying pathway structure of biochemical reaction networks. *Proc. Nat. Acad. Sci. USA* **1998**, *95*, 4193–4198.
- [9] 太田 潤 Isotopomer tracing: 与えられた代謝ネットワークにより原料 isotopomer から生成し得る isotopomer の列挙. 情報処理学会研究報告, **2018**, Vol. 2018-BIO-54, No. 48, 1–6.
- [10] 太田 潤 EMU tracing: 与えられた代謝ネットワークにより原料 elementary metabolite unit (EMU) から生成し得る代謝産物 EMU の列挙. 情報処理学会研究報告, **2018**, Vol.2018-BIO-55, No. 6, 1–7.
- [11] 太田 潤 代謝ネットワークにおける原子レベル完全生成経路の算出. 情報処理学会研究報告, **2019**, Vol. 2019-BIO-58, No. 59, 1–6.
- [12] 太田 潤 代謝ネットワークにおける elementary flux mode 型経路の完全原子レベルマッピング. 情報処理学会研究報告, **2019**, Vol. 2019-BIO-59, No. 2, 1–6.
- [13] 太田 潤 Elementary flux mode 型代謝経路の完全原子レベルマッピングの高速化. 情報処理学会研究報告, **2019**, Vol.2019-BIO-60, No. 6, 1–7.
- [14] Schmidt, K.; Carlsen, M.; Nielsen, J.; Villadsen, J. Modeling isotopomer distributions in biochemical networks using isotopomer mapping matrices. *Biotechnol. Bioeng.* **1997**, *55*, 831–840.
- [15] Antoniewicz, M.R.; Kelleher, J.K.; Stephanopoulos, G. Elementary metabolite units (EMU): A novel framework for modeling isotopic distributions. *Metab. Eng.* **2007**, *9*, 68–86.