

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

太田潤^{†1,a)}

概要：代謝ネットワークにおいて“経路”と呼ばれるものには *in silico* atomic tracing により基質 - 生成物の二項関係を連ねて原料分子を目的分子につないだタイプの“経路”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”とがある。前者の“経路”は原子レベルの情報を含むが目的分子構造の部分構造の生成の道筋であり目的分子構造全体の生成の道筋でない場合があり得る。後者の“経路”は目的分子構造全体の生成の道筋であるが原子レベルの情報を含まない。本稿では、後者の“経路”を elementary flux mode 型経路 (EFM 型経路) と呼ぶ。原理的に EFM 型経路では原料分子のすべての原子と目的分子のすべての原子が一对一に対応している筈である。前回の研究報告において目的分子の分子構造全体の生成経路を isotopomer tracing または EMU tracing に引き続き原子レベルで計算する方法を述べた。この方法を発展させ EFM 型経路に原子レベルの情報を与えるための基本的なアルゴリズムを開発したので報告する。

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 pathways in metabolic networks

JUN OHTA^{†1,a)}

1. はじめに

代謝工学においては有用物質を効率的に産生するための経路と原料代謝産物を知ることが重要である。また、生理学的に重要な代謝産物、生体を構成する分子、代謝中間体が食物として摂取した他の生物の生体構成成分を原料としてどのような経路で生成するかに関する知識は医学的に重要である。代謝ネットワークにおいては、これまでに、大きく分けて2種類の考え方の“経路”が研究されてきた。一方は、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を“経路”とするもの [1-6]、他方は、化学量論的に釣り合った個々の反応の集合を“経路”とするものである [7,8]。前者の“経路”の算出は *in silico* atomic tracing と呼ぶことができる。前者の“経路”は原料代謝産物の原子の1つ以上を目的代謝産物に移動させる生化学的に意味がある経路であると一般的にみなされる。原料代謝産物を目的代謝産物に結ぶ“経路”として、分岐と合流を含む、前者の“経路”の亜型を算出する方法も存在する [4]。亜型も含めて前者の“経路”は原子レベルの情報を含むが、目的分子構造の部分構造の生成の道筋であり目的分子構造全体の生成の道筋であるとは限らない。後者の“経路”においては、“経路”を構成する全反応の反応式を足し合わせたもの (同じ反応が複数回使われることもあり得る) の両辺を相殺した後の左辺に残る消費される代謝産物が原料代謝産物、右辺に残る産生される代謝産物が目的代謝産物と

みなされる。原料代謝産物、目的代謝産物ともに単一種、複数種の場合があり得る。反応式の足し合わせと両辺の相殺の後の原料代謝産物、目的代謝産物の係数が2以上の場合もあり得る。後者の“経路”の代表例が elementary flux mode である [7]。本稿では、後者の“経路”を elementary flux mode 型経路 (EFM 型経路) と呼ぶ。EFM 型経路においては、原料代謝産物がすべて揃っているときに目的分子構造全体が生成することが化学量論的に保証される。EFM 型経路により原料代謝産物の全原子が一对一対応で目的代謝産物の各原子に移動する筈であるが、EFM 型経路そのものは原子レベルの情報を含まない。

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]。研究報告 [11] では、isotopomer tracing または EMU tracing の発展型アルゴリズム (目的 isotopomer・EMU 生成の各段階に相当する connection の記録を行う) とそれに引き続く目的 isotopomer・EMU から原料 isotopomer・EMU 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。この (研究報告 [11] で述べた) 方法を発展させ 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計算するための基本的なアルゴリズムを開発したので報告する。

^{†1} 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

2. 方法

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を行う EFM 型経路が与えられると、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 $target(s)$ の情報が得られる。原料代謝産物 $source(s)$ と目的代謝産物 $target(s)$ は何れも単一の分子であるとは限らない。下記の仮想代謝産物を含む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 (combined_source_metabolite) 分子から単一の仮想目的 (combined_target_metabolite) 分子への経路に変換する。

$source\ reaction: combined_source_metabolite \rightarrow source(s)$

$target\ reaction: target(s) \rightarrow combined_target_metabolite$

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

EFM 型経路構成反応 ($source\ reaction$, $target\ reaction$ を含む) 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。EFM 型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。EFM 型経路内のすべての原子のつながりを単一の CM として表現する。 EFM 型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。

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。必要な場合に CM の 1 列目と 2 列目の各成分に対応する原子の sub-index が参照できるように 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 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}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 の位置は文献 [12] のように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。本稿では M の isotopomer の isotopomer 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 番号による表現を考え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 [13]。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、同じ原子の位置に対応する isotopomer の isotopomer 番号より 1 だけ小さい値となる。分子中の注目する原子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 isotopomer 番号と isotopomer の対応、EMU 番号と EMU の対応が一对一対応であるが、対称性がある分子では、isotopomer 番号の異なる isotopomer どうし、EMU 番号の異なる EMU どうしが区別できない場合がある。この場合、該当する isotopomer あるいは EMU を示すために最も小さい isotopomer 番号あるいは EMU 番号を用いる。本稿では代謝産物 M の isotopomer を M の代謝産物番号 m と isotopomer 番号 t の組み合わせ index である (m, t) で表現する。同様に代謝産物 M の EMU を M の代謝産物番号 m と EMU 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 による一对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。これが 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る。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。EFM 型経路の完全原子レベルマッピングのためには (m_1, t_1) の含む label された原子がすべて (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 である fundamental ICM を用いる。また、target reaction による combined_target_metabolite 生成反応に関わる行については、すべての原子が label された combined_target_metabolite の isotopomer の生成に関係する行のみを ICM に加える。

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_j) に対し、反応 p_1 を介して EMU (mp, t_j) に含まれる原子と一対一対応する原子を有する “ ms_1 から ms_n ” の代謝産物が必ず存在する。このような代謝産物の 1 つを ms_i とし、 ms_i の EMU であって mp の EMU (mp, t_j) 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_{ij}) と表現する。このとき、EMU (ms_i, t_{ij}) が EMU (mp, t_j) に向けて反応 p_1 によりつながっていると本稿では定義し、このつながりを行ベクトル $[(ms_i, t_{ij}), (mp, t_j), p_1]$ で表現する。このつながりの定義は、文献 [13] の EMU reaction 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。EFM 型経路内のすべての EMU のつながりを行ベクトルで表現し、すなわち、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_{ij}) に相当するものをすべて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。target reaction による combined_target_metabolite 生成反応に関わる行については、combined_target_metabolite 分子全体に対応する EMU の生成に関係する行のみを EMUCM に加える。

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 (代謝産物, isotopomer, EMU 等の molecular entity を含む), 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]。

ICM と sub-index-ICM から、代謝産物 m を生成する反応 p により代謝産物 m の isotopomer (m, t) の注目原子の骨格全体を生成するのに必要不可欠な isotopomer の集合をすべて算出し [9]、その結果から、connection の定義を用いて、反応 p により代謝産物 m の isotopomer (m, t) を生成するすべての可能な isotopomer connection が求められる [11]。

EFM 型経路全体に対する ICM が与えられれば ICM の 3 列目から経路内の反応のリストが得られる。この反応リスト内の反応 p により生成する isotopomer のリストは 3 列目が p に等しい ICM の行からなる ICM の部分行列の 2 列目から得られる。すなわち、EFM 型経路内の反応により生成可能な isotopomer (m, t) と反応 p の組すべてを得ることが可能である。それらすべてに上記の手順を行えば、EFM 型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 である。本稿の EFM 型経路の完全原子レベルマッピングのための isotopomer tracing には fundamental isotopomer connection を用いる。

Isotopomer connection を求める場合と同じ手順で EMUCM と sub-index-EMUCM から EMU connection を算出する。EMU connection は EMUCM の部分行列である [11]。

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。

2.9 Isotopomer tracing と EMU tracing

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

Isotopomer tracing・EMU tracing 発展型 [11] を更に発展させ、生成 connection の補足インデックス行列を生成 connection と合わせ記録するようにした。すなわち、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.

最初に与える配列 source (initial_source) は isotopomer tracing では, ICM の 1 列目に現れる原料代謝産物 combined_source_metabolite の isotopomer と label されていない原料代謝産物以外の isotopomer を要素とする配列, EMU tracing では, EMUCM の 1 列目に現れる原料代謝産物 combined_source_metabolite の EMU を要素とする配列とする. 各 tracing のための connection と補足インデックス行列は, 前項の方法で得られるものを用いる.

両 tracing により 3 つの配列 reached, RC, RC_idx が得られる. reached は原料 entity からネットワークで生成し得るすべての entity からなる配列である. 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. “A[j]” により配列 A の j 番目の要素を表すとき, connection 番号 j の connection RC[j] (補足インデックス行列 RC_idx[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 の要素は重複することがあり得る.

2.10 木構造としての原子レベル完全生成経路の算出

Isotopomer tracing, EMU tracing に引き続き, tracing で得られた配列 reached と RC, tracing に用いた配列 initial_source を使い, target から initial_source に含まれる原料 entity に向けて遡りながら経路を計算する [11]. このときの target は仮想の combined_target_metabolite 分子全体に相当する entity (isotopomer または EMU) とする. 以下の P0, P, X は経路 (= 配列) 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) を P0 の要素 (経路 = 配列) とする. P0 内の経路を initial_source に向けて 1 段階, 逆延長する. 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. 残った P 内の未完成経路を空配列 P0 に移して新たな P0 をつくり次段階の逆延長を行う. これを, 配列 P0 が空配列になるまで繰り返す. 原料 entity から target に至る linear な道筋上に同一 entity が重複しないように計算する [11].

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る. 各 1 次元配列 (= 経路) に connection を node とする木構造 (子 node である connection の生成物 entity が親 node である connection の基質 entity となる) が対応する. 木の

root の connection は仮想 combined_target_metabolite 分子全体に相当する entity を生成物 entity とする connection である. 木の leaf の connection は, EMU tracing 後の経路計算の場合は combined_source_metabolite の EMU を基質 entity とし, isotopomer tracing 後の経路計算の場合は combined_source_metabolite の isotopomer と label されていない他の代謝産物の isotopomer の何れかを基質 entity とする. combined_source_metabolite の entity を leaf 以外の node の connection が基質 entity とする場合もあり得る.

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 を node とする木構造全体を entity と反応を node とする木構造に変換し, 各 node に node number, node type, node content, sub-index1, sub-index2 からなる行ベクトル node vector を与える. node number は 1 から始めて全ノードに付ける番号, node type は node が entity か反応かを区別する番号 (entity の場合 1, 反応の場合 2), node content はその node の entity または反応 (entity を示す複素数または反応を示す反応番号) である. sub-index1, sub-index2 は node が entity (node content = NE とする) の場合, 変換前の connection の木構造で, その node (node content = NE) を受け取る connection のその node に相当する基質 entity (=NE) の sub-index を sub-index1, その node (node content = NE) を与える connection のその node に相当する生成物 entity (=NE) の sub-index を sub-index2 とする. root の sub-index1 と leaf の sub-index2 は 0 とする. node が反応である場合の sub-index1, sub-index2 も 0 とする. すべての node vector を重ね合わせて各行が node vector である行列 node list を作る. node 間のつながり情報は 1 列目を子, 2 列目を親とする隣接リスト行列 CM_s に保持する. combined_target_metabolite の entity (木構造の root) の各原子を, 木構造のつながりに沿って leaf である combined_source_metabolite の entity まで逆向き trace する. CM に保持された各反応の基質と生成物の間の原子の対応関係の情報を逆向き trace に用いる. この過程で, node content が entity である node のそれぞれについて, その node の entity の逆向き trace された原子の位置番号 position とその原子に対応する combined_target_metabolite の原子の位置番号 position_root の組 (position, position_root) の集合を得る. これを保持し, 次項の assemble に用いる.

2.11 木構造としての原子レベル完全生成経路の merge による EFM 型経路の完全原子レベルマッピング

combined_source_metabolite の entity であるすべての leaf から始めて, 木構造の辺を combined_target_metabolite の entity である root に向けて順次 merge (統合) していき, combined_source_metabolite から combined_target_metabolite に至る EFM 型経路の完全原子レベルマッピングを行う.

“merge”の説明のために下記の 3 node, 2 辺からなる部分木構造 A, A', B, C, C' を定義する (→ は子から親に向か

う). M1, M2, M3 は代謝産物, p, q は反応である.

A : M1 の EMU “001” $\rightarrow$ p $\rightarrow$ M2 の EMU “01”

A' : M1 の EMU “010” $\rightarrow$ p $\rightarrow$ M2 の EMU “01”

B : M1 の EMU “010” $\rightarrow$ p $\rightarrow$ M2 の EMU “10”

C : M1 の EMU “100” $\rightarrow$ p $\rightarrow$ M3 の EMU “1”

C' : M1 の EMU “100” $\rightarrow$ q $\rightarrow$ M3 の EMU “1”

このとき A, B, C は下記の D に “merge できると” する.

D: M1 の EMU “111” $\rightarrow$ p $\rightarrow$ M2 の EMU “11”+M3 の EMU “1”

一方 A', B, C は, A' と B の M1 の EMU の原子が重複しているので “merge できない” とする. また, M1 の EMU “111” を同時に異なる反応 p と q に使うことはできないので, M1 を反応 p で利用する A, B と反応 q で利用する C' は “merge できない” とする. すなわち merge の結果, 1 entity (isotopomer または EMU) が複数の反応につながるような merge はできない. p による M2 と M3 の生成を含む A, B, C の D への merge ができるとすることからわかるように, merge の結果, 1 反応が複数の entity につながるような merge は可能である. 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. 以上から, 複数の node (node content が同じ代謝産物に属する entity である) が assemble できるための条件は, 下記①②である.

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

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

EFM 型経路の完全原子レベルマッピングのための木構造の leaf からの merge ができるためには, 条件①を満たすよう combined_source_metabolite の entity であるすべての leaf node が 1 つの node に assemble できなければいけない.

そこで, 前項の方法で算出される経路において, 利用されている combined_source_metabolite の entity (entity と反応を node とする木構造の leaf として存在する) をすべて列挙し, それらが指定する combined_source_metabolite の原子を原子位置ごとにすべて count し, count が 2 以上の原子位置がない経路のみ, 次の手順で条件①②を満たすよう merge することを試みる. 以下, “A[j]” により配列 A の j 番目の要素を表す.

Step 1. 経路の木構造情報を持つ node list および CM_s を与える. d, nv を空配列, i0 = 1, di = 1 とする. CM_s の成分 (node number) の最大値に 1 を加えた値を ni とする.

Step 2. node content 項が combined_source_metabolite の entity でありかつ node type 項が 1 である node list の行の行番号の配列 u を得る. 行番号が u に含まれる node list の行の各 node number 項をすべて要素とする配列を d[di] に格納する.

Step 3. node list の u[1] 行目の複製である node vector の node number 項を ni に, node content 項をその実部に置き換えた行ベクトルを nv[di] に格納し, ni = ni + 1, di = di + 1 とする.

Step 4. d[i0] 中の node number を持つ node の親 node の node number の配列 na を node list と CM_s から得る.

Step 5. n_type を node number が na[1] である node の node type とし, その値を node list から得る.

Step 6. node list の, node number 項値が na に含まれている行からなる部分行列を作る. ただし, na の任意の要素 na[j] について, na[j] が部分行列の node number 列の j 番目に現れるようにする. 得られた部分行列から node content, sub-index1, sub-index2 の列部分の部分行列として NCI を得る. NCI (node content 列に複素数を含む) の全成分の実部からなる行列として real_NCI を得る. real_NCI の重複行を unique 化することにより行列 unique_real_NCI を得る.

Step 7. unique_real_NCI の行数が 2 以上でありかつ n_type が 2 に等しいという条件に合致すれば merge できないと判定し計算を終了する. 合致しなければ **Step 8** に進む.

Step 8. unique_real_NCI の各行 (以下で W とする) に対して次の処理を行う.

W と一致する real_NCI の行の行番号の配列 v を得る. v の要素数が 2 以上の場合のみ, d[di] に, “配列 na の要素であってその要素番号が v に含まれているもの” からなる配列を格納し, nv[di] に, node number 項が ni, node type 項が n_type, node content 項・sub-index1 項・sub-index2 項の 3 項が W の node content 項, sub-index1 項, sub-index2 項と同じ node vector 形式の行ベクトルを格納する.

その後に ni = ni + 1, di = di + 1 とする.

Step 9. unique_real_NCI の全行に対して **Step 8** の処理が終わった後に i0 = i0 + 1 とする.

Step 10. i0 が d の要素数以下である場合は **Step 4** に戻り処理を続ける. i0 が d の要素数よりも大きい場合は **Step 11** に進む.

Step 11. d の要素数以下の自然数であるすべての j0 に対して以下の処理を行う.

nv[j0] の node type 項が 1 である場合 d[j0] に含まれる node number の node すべての assemble を試みる. assemble できない場合は merge できないと判定し計算を終了する. 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. assemble できる場合は, nv[j0] の node vector の node content 項を assemble によって得られた entity で置き換える.

Step 12. d の要素数以下の自然数であるすべての k に対して以下の処理を行う.

d[k] に含まれる node number の何れかに一致する CM_s の成分 (node number) すべてを nv[k] の node number 項の node number で置き換える.

上記の処理が終わった後に CM_s の重複行を unique 化する.

Step 13. Step 12 で CM_s から除かれた node に対応する node list の node type の値を元の値に-1 を乗じた値で置き換える.

Step 14. nv のすべての要素を node list に node list の新たな行として追加する.

3. 計算実験の結果と考察

糖質代謝モデルネットワーク [6] のペントースリン酸経路に含まれる 2 種類の transketolase 反応・transaldolase 反応・ribulose-5-phosphate 3-epimerase 反応の計 4 反応 (文献 [6] のプロセス番号 26, 28, 29, 30) の逆反応と ribose-5-phosphate ketoisomerase 反応 (文献 [6] のプロセス番号 27) の組み合わせ (計 5 反応、延べ 7 反応) により化学量論的に 2 分子の fructose 6-phosphate と 1 分子の glyceraldehyde 3-phosphate から 3 分子の ribose 5-phosphate を生成する反応収支式ができる. これを 5 反応 (延べ 7 反応) からなる EFM 型経路とみて本稿の方法で炭素原子部分の完全原子レベルマッピングを試みた. 計算は Matlab と GNU Octave 上で行った.

炭素原子部分の EMU tracing に引き続き木構造としての原子レベル完全生合成経路を算出したところ仮想 combined_target_metabolite 分子全体の炭素原子部分に対応する EMU を root とし combined_source_metabolite の EMU を leaf に持つ 4096 の経路 (=木構造) が得られた. この中で combined_source_metabolite の EMU に相当する node の assemble が可能な 96 の経路 (=木構造) を選択し, merge を試みたところ, 24 の merge された経路が得られた. 得られた経路は原子位置が定まった EMU が反応を介してつながっているネットワーク構造である. ネットワーク内の EMU の原子位置の情報と CM に保持された各反応の基質と生成物の間の原子の対応関係の情報を基に 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ことができた. 各経路を特徴づける, combined_source_metabolite から combined_target_metabolite に至る, EMU と反応の sequence の集合を求めたところ. 集合は 24 経路に対して 2 種類しかなく, 2 種類の集合のそれぞれが 12 経路に対応していた. combined_source_metabolite から生成する fructose 6-phosphate は 2 分子である. また, 3 分子の ribose 5-phosphate から combined_target_metabolite が生成する. そのため本稿の手順に従えば 1 種類の経路を $2! \times 3! = 12$ 回 count することになる. 24 経路を特徴づける集合が 2 種類であったことはこのことと一致する. merge できた 24 経路すべてにおいて反応の使用回数は前述した元の EFM 型経路の “5 反応 (延べ 7 反応)” と使用回数の内訳も含めて一致した. Isotopomer tracing に引き続き経路計算を行った場合も経路数と反応の使用回数について全く同じ結果が得られた. 計算時間は EMU tracing と isotopomer tracing のどちらに引き続いて行う場合も, Matlab で約 14 秒, GNU Octave

で約 150 秒であった (Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz).

2 分子の fructose 6-phosphate と 1 分子の glyceraldehyde 3-phosphate から 3 分子の ribose 5-phosphate を生成する経路が原子レベルで見ると 2 種類存在することが明らかになった. 本稿ではその詳細を述べないが, 一方は cycle を含まない経路, 他方は cycle を含む経路であった. EFM 型経路における完全原子レベルマッピングは一通りであるとは限らないと言える. 前者の cycle を含まない経路については, 代謝産物レベルの経路として教科書に記載されている. 後者の cycle を含む経路についてはこれまでの報告を調査中である. 本稿で試みた EFM 型経路の完全原子レベルマッピングにより代謝経路の今まで気づかれなかった隠された性質が明らかになる可能性がある.

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. *IPSI 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. 研究報告バイオ情報学(BIO), **2018**, 2018-BIO-54(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. 研究報告バイオ情報学(BIO), **2018**, 2018-BIO-55(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合成経路の算出. 研究報告バイオ情報学(BIO), **2019**, 2019-BIO-58(59),1–6.
- [12] Schmidt, K.; Carlsen, M.; Nielsen, J.; Villadsen, J. Modeling isotopomer distributions in biochemical networks using isotopomer mapping matrices. *Biotechnol. Bioeng.* **1997**, *55*, 831–840.
- [13] 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.