Vol.2019-BIO-60 No.6
2019/12/2

# 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

太田潤[†1,a)]

**概要**：代謝ネットワークにおける“経路”には *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”がある．前回の研究報告で，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ための基本アルゴリズムを報告した．本研究報告では基本アルゴリズムの改良型を報告する．報告する基本アルゴリズムの改良型では，同一分子複数個を生成する EFM 型経路に原子レベルの情報をより速く与えることができた．

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

# Acceleration of complete atomic-level mapping of elementary flux mode-type metabolic pathways

JUN OHTA[†1,a)]

## 1. はじめに

代謝ネットワークにおける経路に関する知識・情報は，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効率的な方策を得るために，医学においては生理学的に重要な代謝産物，生体を構成する分子，代謝中間体の生成を理解するために重要である．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，大きく分けて 2 種類の考え方の“経路”が研究されてきた．1 つは，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としての“経路”[1-6]，他の 1 つは，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”である［7,8］．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．前者の“経路”においては，“経路”を目的代謝産物から原料代謝産物に向けて逆に辿れば目的代謝産物の 1 つ以上の原子が原料代謝産物から移動してくる原子レベル情報を含む道筋を知ることができるが，それが目的代謝産物の分子構造全体の生成の道筋であることは保証されない．一方，後者の“経路”（EFM 型経路）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場合はその回数だけ加える）の両辺を相殺した後の左辺に残る消費される代謝産物を原料代謝産物，右辺に残る産生される代謝産物を目的代謝産物とみなすとき，原料代謝産物すべてが揃っていればそれ（ら）から目的代謝産物の分子構造全体が生成すると化学量論的に結論できる．このことは，EFM 型経路により原料代謝産物の全原子が一対一対応で目的代謝産物の各原子に移動する道筋が存在することを意味する．しかし，これまで，その道筋を原子レベルで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］．研究報告［11］では，isotopomer tracing または EMU tracing の発展型アルゴリズム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．研究報告［12］では，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，EFM 型代謝経路の完全原子レベルマッピングのための基本的なアルゴリズム（基本アルゴリズム）を報告した．その後，基本アルゴリズムを改良し EFM 型代謝経路の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なった．本稿では，この基本アルゴリズムの改良型を報告する．

## 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

†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

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個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 + c*i* 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．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［13］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［14］．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 isotopomer 番号と isotopomer の対応，EMU 番号と EMU の対応が一対一対応であるが，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 + t*i* 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． “$(m_1, t_1)$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， $ms_1, ms_2, \cdots, ms_n$（“$ms_1$ 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，文献［14］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．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（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 $A_1$，$A_2$，・・・，$A_n$ 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，$[a_1, b, f]$，$[a_2, b, f]$，・・・，$[a_n, b, f]$ 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．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，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］．本稿の EFM 型経路の完全原子レベルマッピングのための 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，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．

本稿の目的のためには，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．以下，本稿ではこの記法を用いる．

### 2.10 EFM 型経路の完全原子レベルマッピングのために必要な情報

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 分子である場合，

unique_target_metabolite = 配列{4, 5}

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 EFM 型経路の完全原子レベルマッピングのための補助関数

**route2source_core**

tracing で得られる配列 reached と RC，tracing に用いる配列 initial_source を使い，initial_source に含まれる原料 entity（isotopomer または EMU）から entity である target が生成する経路を算出する［11］．target の entity から原料 entity に向けて遡りながら経路を計算する，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．

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．

**count_source_atom**

配列 X_source_atom を，combined_source_metabolite の代謝産物番号 csm，上記の route2source_core で算出される X と利用される RC，RC に対応する RC_idx から算出する．配列 X_source_atom は，X に（要素として）含まれる経路で利用される combined_source_metabolite の原子の個数を combined_source_metabolite の原子位置ごとに示す行ベクトルを要素とする．X_source_atom[j]には，経路 X[j]により利用される原子の原子位置ごとの個数が格納される．

**combine_connection_tree**

connection 木構造と対応する配列 Zi・配列 Y がこの順序で与えられたとき，Zi の connection 木構造の root に Y の connection 木構造の root を子として結合した connection 木構造に対応する配列 Zj を算出する．Y としては，route2source_core で算出される配列 X の要素（＝connection 木構造と対応する配列）を想定する．

**ismergeable**

combined_source_metabolite の代謝産物番号 csm，connection 木構造と対応する配列 Zj，“merge 可能な経路算出のためのアルゴリズム”（6 頁）内で算出される work_RC と work_RC_idx，前述の work_CM_all，work_idxCM_all，symmetry_info を与えたとき，Zj が“merge”［12］に関して merge 可能な場合 assembled＝1 を，merge 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．

Zj は“merge 可能な経路算出のためのアルゴリズム”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 をつくることが伴う．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・不可能の判定は研究報告［12］で述べた手順で行う．

**initial_ordered_comb と next_ordered_comb**

n を自然数，m と k を n 以下の自然数，Cb を 1 から n までの自然数から m 個を選び小さい方から並べた配列とする. 2 つの異なる Cb である $Cb_1$ と $Cb_2$ に対し, $Cb_1[j] \neq Cb_2[j]$ となる j の最小値 j0 を求める．このとき $Cb_1[j0] < Cb_2[j0]$ ならば $Cb_1$ は $Cb_2$ より小, $Cb_1[j0] > Cb_2[j0]$ならば $Cb_1$ は $Cb_2$ より大と定義する．すると，すべての Cb を最大のものから始めて最小のものまで順番に並べることができる．

initial_ordered_comb は上記の n と m を与えたとき，それに対して可能な Cb のうち最大のものを算出する．

next_ordered_comb は上記の Cb の 1 つである $Cb_j$ と上記の n と k を与えて使う．k が n と等しいとき，$Cb_j$ より小さい Cb が存在すればその中で最大のものを Cb として算出する．存在しなければ Cb を空配列として算出する．k が n と異なる（n よりも小さい）とき，$Cb_j$ より小さい Cb であってその 1 番目から k 番目までの要素が $Cb_j$ の 1 番目から k 番目までの要素と完全には一致しない Cb が存在すればその中で最大のものを Cb として算出する．存在しなければ Cb を空配列として算出する．

**2.12 EFM 型経路の完全原子レベルマッピング**

前述の，EFM 型経路の完全原子レベルマッピングのために必要な情報と補助関数を用いて，“merge 可能な経路算出のためのアルゴリズム”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．merge された経路に，work_CM_all が保持する各反応の基質 - 生成物間の原子の対応関係の情報を基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が完了する．

“merge 可能な経路算出のためのアルゴリズム”では配列 A の第 j 番（j 番目）の要素を A[j]，要素数を length(A) と表す．また配列 A，B がある時，A + B により A に B の要素を順番に追加してできる配列を表現する. {a | 条件 A } により条件 A を満たすすべての a からなる配列を，{条件 A}により条件 A に記載された要素を持つ配列を表す. 本稿の{}の囲みは集合ではなく配列を示すが，特定の要素が配列に含まれるか否かは$\in$と$\notin$により表す. {}により要素を持たない空配列を示す．A[j] = a により a を配列 A の第 j 番の要素として代入することを表す．本稿は，配列の第 j 番目の要素の要素番号は j – 1 でなく j であるものとして記述している．

## 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アルゴリズムで炭素原子部分の完全原子レベルマッピングを試み，研究報告［12］の基本アルゴリズムの計算結果と比較した．計算は Matlab と GNU Octave 上で行った．

EMU tracing に引き続き経路計算を行った場合，本稿のアルゴリズムでは，4 つの merge された経路が得られた．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 4 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 2 経路に対応していた．研究報告［12］のアルゴリズムでは，24 の merge された経路が得られた．各経路を特徴づける，EMU と反応の sequence の集合は 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．これに対応して，研究報告［12］のアルゴリズムでは 1 種類の経路あたりの count が 2!×3! = 12 回であったが，本稿のアルゴリズムでは fructose 6-phosphate が 2 分子であることに起因する 2 回であったことになる．本稿のアルゴリズムで得られた 2 種類の集合に対応する経路の一方は cycle を含まない経路，他方は cycle を含む経路であり，研究報告

**merge 可能な経路算出のためのアルゴリズム**

```
work_RC = RC, work_RC_idx = RC_idx
work_RC[length(work_RC)+1] = ameCM
work_RC_idx[length(work_RC_idx)+1] = aidxmeCM
Z0[1] = length(work_RC)
U0[1] = csm_an 個の成分 0 からなる行ベクトル
utmi = 1
while ( utmi ≤ length(unique_target_metabolite) )
  target = unique_target_metabolite[utmi]
  X = route2source_core(initial_source, target, reached, RC)
  X_source_atom = count_source_atom(csm, X, RC,
  RC_idx)
  uniX_source_atom = {X_source_atom[j] | j∈B,
  B = { (X_source_atom[a] = X_source_atom[b]となる b∈B
  が任意の要素番号 a に対して存在)かつ(c, d ∈B かつ
  c≠d ならば X_source_atom[c]≠X_source_atom[d]) } }
  cXa = {}, Sa = {}, cXai = 1, ui = 1
  while ( ui ≤ length(uniX_source_atom) )
    if ( uniX_source_atom[ui]の成分すべてが 1 以下 )
      v = { j | uniX_source_atom[ui] = X_source_atom[j] }
      cXa[cXai] = { X[j] | j∈v }
      Sa[cXai] = uniX_source_atom[ui]
      cXai = cXai + 1
    end
    ui = ui + 1
  end
  cX = {}, Z1 = {}, U1 = {}
  n = length(cXa)
  m = number_of_unique_target_metabolite[utmi]
  Cb = intial_ordered_comb(n, m)
  while (break されるまで実行)
    sv = csm_an 個の成分 0 からなる行ベクトル
    is_atom_duplication_OK = 1
    k = 0
    i0 = 1
    while ( i0 ≤ length(Cb) )
      sv = sv + Sa[Cb[i0]]
      k = i0
      if ( 1 より大きい sv の成分が存在 )
        is_atom_duplication_OK = 0
        break
      end
      i0 = i0 + 1
    end
    if ( is_atom_duplication_OK = 1 )
      Z = {}
      z0i = 1
      while ( z0i ≤ length(Z0) )
        tv = Sa[Cb[1]] + Sa[Cb[2]] + ・・・+ Sa[Cb[m]]
             + U0[z0i]
        if ( 行ベクトル tv の成分すべてが 1 以下 )
          Z = Z + {Z0[z0i]}
        end
        z0i = z0i + 1
      end
      cX = { cXa [j] | j∈ Cb }
      cXi = 1
      while ( cXi ≤ length(cX) )
        W = {}, U = {}, Y = cX[cXi]
        zi = 1
        while ( zi ≤ length(Z) )
          yi = 1
          while ( yi ≤ length(Y) )
            Zj = combine_connection_tree(Z[zi], Y[i0])
            [assembled, atom_count] = ismergeable (csm,
            Zj, work_RC, work_RC_idx, work_CM_all,
            work_idxCM_all, symmetry_info)
            if ( assembled = 1 )
              W = W + {Zj}
              U = U + {atom_count}
            end
            yi = yi + 1
          end
          zi = zi + 1
        end
        Z = W
        cXi = cXi + 1
      end
      Z1 = Z1 + Z
      U1 = U1 + U
    end
    Cb = next_ordered_comb(Cb, n, k)
    if ( Cb = {} )
      break
    end
  end
  Z0 = Z1
  U0 = U1
  utmi = utmi + 1
end
Z = Z0
```

［12］のアルゴリズムで得られた経路と一致した．Isotopomer tracing に引き続き経路計算を行った場合にも同様の経路が算出された．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 1.2 秒程度, GNU Octave で 8 秒弱であった（Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz）．この計算時間は研究報告［12］のアルゴリズムによる計算時間の 10 分の 1 以下であり, 本稿のアルゴリズムで計算の高速化が実現された．この高速化は，目的代謝産物の 3 分子の ribose 5-phosphate に起因する経路の 3!倍の重複計算を回避したことと，combined_target_metabolite 分子全体を生成する経路をすべて算出後に merge 可能・不可能の判定を行うのではなく“merge 可能な経路算出のためのアルゴリズム”の 58 行目と 59 行目の部分で root の connection に逐次 merge 可能と判定された connection 木構造を付加することにより経路を完成させていったことによると考えられる．解糖系とペントースリン酸経路の反応による 5 分子の glucose 6-phosphate から 6 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は，計算速度の問題で研究報告［12］のアルゴリズムでは実際的でなかったが本稿のアルゴリズムにより実際の計算が可能となったことを付記する．

## 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. 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] 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. 研究報告バイオ情報学(BIO), **2019**, 2019-BIO-59(2),1-6.
[13] Schmidt, K.; Carlsen, M.; Nielsen, J.; Villadsen, J. Modeling isotopomer distributions in biochemical networks using isotopomer mapping matrices. *Biotechnol. Bioeng*. **1997**, *55*, 831–840.
[14] 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.