Vol.2015-MPS-103 No.61
Vol.2015-BIO-42 No.61
2015/6/25

低分子化合物二次元構造比較システムの改良

杉原 舜[1]　石田 貴士[2,3]　秋山 泰[2,3†]

概要：薬剤の開発において，既に薬剤として知られる化合物と類似した構造を持つ化合物を見つけることは重要な技術である．本研究では，2 つの低分子化合物の共通部分構造 (maximum common subgraph/substructure, MCS) を求めて Tanimoto 係数を用いて類似度を計算する KCOMBU という既存のシステムの改良を行った．構造比較する化合物同士の原子数が大きく異なる場合には Tanimoto 係数が小さい値となることを利用したフィルターを実装し，化合物同士の共通部分を求める計算を省略することで，ChEMBL18 データベースへの化合物探索を KCOMBU に比べて最大で約 14 倍高速に行うことに成功した．

キーワード：化合物構造比較，MCS，KCOMBU，Tanimoto 係数フィルタリング

Improvement of a Chemical Structure Comparison System

SHUN SUGIHARA[1]　TAKASHI ISHIDA[2,3]　YUTAKA AKIYAMA[2,3†]

Abstract: Finding a chemical compound similar to approved drugs or promising candidate compounds is important technique for drug discovery. In this study, we improved the chemical structure comparison system named KCOMBU which used maximum commom subgraph/substructure (MCS). We developed the Tanimoto coefficient filter which provides upper bound of Tanimoto coefficient where a threshold parameter is given. The calculations finding MCS between two compounds is omitted by the filter. As a result, we achieved up to 14 times faster calculation of similarity search for ChEMBL 18 database than KCOMBU.

Keywords: chemical structure comparison, MCS, KCOMBU, Tanimoto coefficient filter

1. はじめに

1.1 化合物の構造比較

新規薬剤の発見には，既に薬剤として認められている化合物や考慮中の候補化合物などと，大量の化合物ライブラリとの構造比較が大変重要となる[1]．既に薬剤として認められている化合物と構造的に類似する化合物は，その薬剤の標的タンパク質に対して同様に薬剤となる可能性が大きいため，創薬の初期段階におけるシード化合物の探索などに化合物の構造比較が使用されている．

化合物ライブラリは大量の化合物のデータを集約しているデータベースであり，有名な化合物ライブラリとして ChEMBL[2][3]がある．ChEMBL とは European Bioinformatics Institute が開発した医薬品及び医薬品候補化合物などの生物活性に関与する低分子化合物・タンパク質データベースである．薬物を中心とした低分子化合物が登録されており，2015 年 2 月にリリースされた ChEMBL バージョン 20 には 1,715,667 個の化合物，10,774 個の標的タンパク質，及び 1,352,737 件の生物活性情報が収録されている．これらの化合物の分子構造，標的とするタンパク質情報，アッセイ情報などが登録されそれらをリンクさせた検索が可能となっている．

薬剤開発などの応用研究ではクエリ化合物とこのような大量の化合物について類似度の検索が必要である．化合物の類似度を計算する方法として，化合物の特徴量 (fingerprint など) を計算した特徴ベクトルを用いる分子記述法と，化合物中の原子の対応付けに基づいて計算する原子マッチング法がある[4]．分子記述法は，特徴ベクトル間の内積などで類似度計算を行うため計算コストは比較的小さいが，化合物の原子構成などを類似度に反映することが難しいという問題がある一方，原子マッチング法は化合物の原子 1 つ 1 つを対応付けるため計算量が比較的大きくなるが，類似した部分構造が得られるため視覚的な理解が可

1 東京工業大学 工学部 情報工学科
（現在，東京工業大学 大学院理工学研究科 通信情報工学専攻）
Department of Computer Science, Faculty of Engineering, Tokyo Institute of Technolgy. (In present, Department of Communications and Computer Engineering, Graduate School of Engineering, Tokyo Institute of Technology)
2 東京工業大学 大学院情報理工学研究科 計算工学専攻
Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology.
3 東京工業大学 情報生命博士教育院
Education Academy of Computational Life Sciences, Tokyo Institute of Technology
† akiyama@cs.titech.ac.jp

Vol.2015-MPS-103 No.61
Vol.2015-BIO-42 No.61
2015/6/25

能であるというメリットがある．原子マッチング法に基づいてあるクエリ化合物とデータベース中の化合物との構造を比較し，類似している化合物をランキング形式で出力する機能を持つツールがこれまでにいくつか開発されている[4][5][6][7][8][9]. これらのツールは化合物同士の Maximum Commom Subgraph/Substructure（以下 MCS）[4]を求め，その MCS によって類似度を算出するが，中でも 2011 年に開発された KCOMBU (Kemical compound COMparison using Build-Up algorithm)[9]は， Build-Up Algorithm と呼ばれる貪欲法の改良手法を用いており, MCS を高速に求めることができる．しかしながら探索対象となるデータベースの化合物の数は 100 万件を超え，KCOMBU による探索でも 1〜2 時間程度の計算時間を要する．類似化合物の探索は創薬の初期段階で用いられる基本的な手法であり，クエリ化合物を変えて複数回試行を繰り返すことも多い．気軽に化合物探索を実施するためには数分〜数十分の計算時間が限度であり，KCOMBU のさらなる高速化が求められている．

1.2 本研究の目的

本研究は，データベースに対する類似化合物の探索を高速化することを目的とする．KCOMBU の化合物同士の類似度計算に対し，実際に類似度計算を行う前にデータベース中の化合物群にフィルタリングを行うことで，計算対象の化合物を絞り込み，計算の高速化を行った．

2. 化合物類似度の計算

2.1 KCOMBU の化合物類似度計算方法

KCOMBU では，類似度を Tanimoto 係数（式(1)）により求める．V_{X}は化合物 X の原子数であり，m は 2 つの化合物の共通部分の原子数である．

$$\mathrm{Similarity(A,B)} = \frac{m}{V_{\mathrm{A}} + V_{\mathrm{B}} - m} \tag{1}$$

KCOMBU では m の算出にあたって，原子の対応付けを決める原子分離を以下のように定義している．

・原子種（C, N, O など）
・環構造の一部かどうか（C, C@）
・結合している原子の数（O1, O など）

環構造を構成する原子は原子名の後に@をつけ，環構造中では単結合や二重結合を区別しない．また，化合物は重原子のみ扱う．

2.2 Maximum Common Substructure (MCS)

2 つの化合物同士で，分類された原子が一致する部分を共通部分（common substructure）といい，その共通部分が最大のグラフとなるものを MCS と定義する．しかし共通部分構造がつながっているかどうかで，MCS に関しても異なる定義が考えられる. KCOMBU は 4 種類の異なる MCS をそれぞれ計算することが可能となっているが，本研究では最も一般的に用いられる connected MCS (C-MCS)[4]を用いた．C-MCS は共通部分構造が全てつながった MCS であり，共通部分構造の原子数 m は比較的小さくなるが，他の MCS に比べて高速に計算が可能である．

2.3 Build-Up Algorithm

KCOMBU では MCS を求める方法として Build-Up Algorithm という近似アルゴリズムを利用している．Build-Up Algorithm は貪欲法に基づいており, セレクションスコアというスコアに基づいて原子のペアを選んでいく. 1 原子対 1 原子の対応から始め，近似スコアの上位の組にのみ原子ペアを加えていくアルゴリズムである．アルゴリズムのアウトラインを図 1 に示す．選択する原子ペア数を K とし，KCOMBU では $K = 40$ としている．$K = 1$ のときは単純な貪欲法となる．

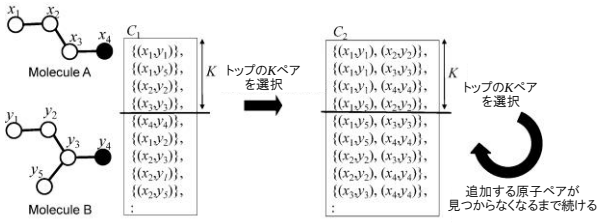

図 1　Build-Up Algorithm のアウトライン

2.4 KCOMBU で用いられているフィルター

KCOMBU では 1 つの化合物といくつかの化合物の類似度を計算し，指定した閾値以上の類似度を持つ化合物をランキング形式で出力する際にフィルターが働くようになっている．フィルターは，共通部分を構成する原子群が必ず両方の化合物の構成原子となっていなければならないことを利用して実装されている．比較を行う 2 つの化合物が共通して持つ原子の数をM_fと定義すると，M_fは式(2)のように定義される．$N_{\mathrm{X}}(e)$ は化合物 X の種類 e の原子の数を表す.

$$M_f = \sum_{e \in \text{all the atom types}} \min[N_{\mathrm{A}}(e), N_{\mathrm{B}}(e)] \tag{2}$$

さらに，M_fが必ず MCS の原子数 m 以上になるのは明らかなので，式(1)と合わせて式(3)が成り立つ．式(3)の右辺の値は MCS を求める前に計算が可能なので, 与えられた 2 つの化合物に基づいて MCS の算出を省略することができる．

$$\frac{m}{V_{\mathrm{A}} + V_{\mathrm{B}} - m} \leq \frac{M_f}{V_{\mathrm{A}} + V_{\mathrm{B}} - M_f} \tag{3}$$

このフィルターは MCS を求める計算の大部分をカットできるが，現在の KOMBU の実装では 1 つ 1 つ化合物のフ

Vol.2015-MPS-103 No.61
Vol.2015-BIO-42 No.61
2015/6/25

ファイルを開いて原子の数を種類ごとに求めているので，フィルタリング自体にも大きな計算時間がかかっている．

3. 提案手法

3.1 原子数による新たなフィルタリングの提案

前述のように KCOMBU には既に複数の化合物との類似度を求める際に働くフィルターが存在しているが，このフィルターは原子の種類ごとの原子数を必要とする．即ち，化合物のファイルを 1 つ 1 つ開いて構成原子の情報を参照する必要がある．そこで本研究では，化合物全体の原子数のみを用いたより効率的なフィルターを提案する．また更なる高速化のため，化合物ファイルを開くことなく化合物の原子数を参照することで I/O の削減を行った．

本研究で提案するフィルターは，ユーザによって与えられる類似度の閾値 P に基づいて，以下に示す Tanimoto 係数の上限を利用して計算される．

1. $V_A \leq V_B$ の場合

最大共通部分の原子数 m のとりうる値は，$0 \leq m \leq V_A$ となる．上限として $m = V_A$ を式(1)に代入して，以下を得る．

$$P \leq \frac{V_A}{V_A + V_B - V_A}$$
$$\therefore P \leq \frac{V_A}{V_B} \tag{4}$$

2. $V_A > V_B$ の場合

最大共通部分の原子数 m のとりうる値は，$0 \leq m \leq V_B$ となる．上限として $m = V_B$ を式(1)に代入して，以下を得る．

$$P \leq \frac{V_B}{V_A + V_B - V_B}$$
$$\therefore P \leq \frac{V_B}{V_A} \tag{5}$$

P はユーザから与えられる閾値であり，クエリ化合物を A とすると，P も V_A も比較する化合物ファイルを開く前に値を得ることができる．また m が取りうる最大の値を用いているので，式(4)（あるいは式(5)）を満たさない化合物 B は，類似度の閾値を超えないことが分かる．式(4)と式(5)を用いて 2 つのフィルターを設けられるが，フィルターの簡略化のため，V_Bについて式(4)と式(5)をまとめると

$$PV_A < V_B < \frac{V_A}{P} \tag{6}$$

となる．即ち計算する化合物をデータベースから選ぶ際に，式(6)を満たす原子数の化合物のみを計算すれば良いということになる．

データベース中の化合物の原子数を参照する際に，本研究では 3 つの方法を提案した．以下ではそれぞれの手法について述べる．

3.2 提案手法 1：フィルターの改良

各化合物ファイルから原子数を参照し，式(6)で提案したフィルターを用いて計算する化合物を限定する．この手法では，KCOMBU にあらかじめ実装されているフィルターと同様に I/O の処理に時間がかかるため，計算速度の改善には限界があると考えられる．

3.3 提案手法 2：リストを用いた手法

式(6)によるフィルタリングを用いる際，事前にデータベース中の化合物の原子数を取得しておき，I/O を減らすことで高速化が可能である．提案手法 2 では，対象のデータベース中の全化合物に対して原子数を事前に計算しておき，独立したテキストファイルにリスト化しておくことで I/O の削減を行った．本研究ではデータベースとして ChEMBL を対象とした．ChEMBL に登録されている化合物は CHEMBL ID という ID が付与されており，化合物群を CHEMBL ID と原子数で表したリスト（図 2）を生成する．このリストを配列に格納し参照することで，類似度を比較する化合物のファイルを開くことなく，式(6)の範囲に含まれるかどうかを判定できる．

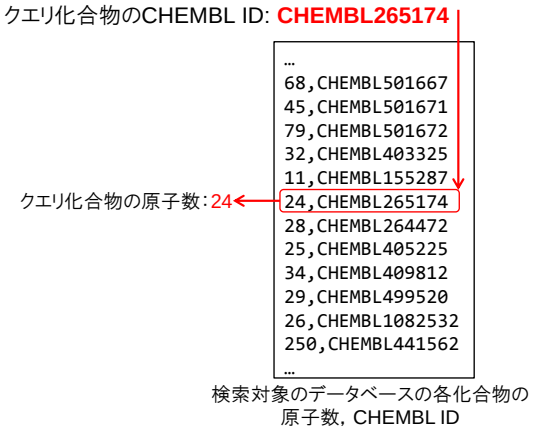

図 2　テキストファイルからの原子数取得の例

3.4 提案手法 3：ファイル名を用いた手法

提案手法 2 によって高速化が可能となるが，この方法では対象となる CHEMBL ID と原子数情報をテキストファイルのリスト上から探索する必要があるため，データベースの化合物の各ファイルにあらかじめ原子数を付与しておくことでさらなる高速化が可能と考えられる．この方法を提案手法 3 とする．この手法では化合物のファイル名を，

「原子数 + CHEMBL ID + 拡張子」

とする．例として，CHEMBL ID が CHEMBL265174 で原子数が 24 である化合物の sdf ファイルは，ファイル名が `24CHEMBL265174.sdf` となる．この手法を用いることで，

Vol.2015-MPS-103 No.61
Vol.2015-BIO-42 No.61
2015/6/25

外部のテキストファイルを参照することなく直接原子数を得ることができ，さらなる高速化が期待できる．

4. 評価実験

4.1 データセット

4.1.1 クエリ化合物

本研究では化合物のデータベースとして ChEMBL[2][3]を用いた．しかし，薬剤候補化合物の探索を想定するため，既に薬剤として利用されている化合物と原子数が乖離しないように ChEMBL からクエリ化合物を選択する目的で，本研究ではまず既に薬剤として利用されている化合物のデータベースである DRUGBANK[10][11]を用いて原子数の分布を求めた．6,874 個の化合物データを収集し，原子数分布を求めたものを図 3 に示す．図 3 より，DRUGBANK の化合物の原子数は 50 以下が大半を占めているため，本研究ではクエリ化合物として原子数 50 までの ChEMBL の化合物データを扱った．

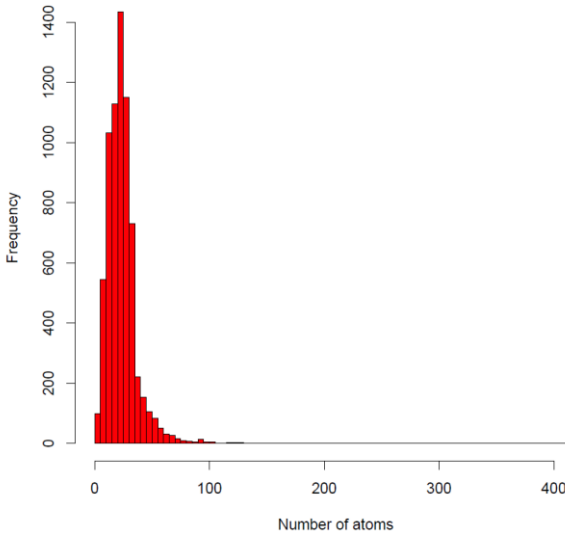

図 3　DRUGBANK に含まれる化合物の原子数

4.1.2 データベース化合物

探索対象のデータベースとして ChEMBL version 18 (2014 年 4 月更新版) を用いた．ChEMBL 18 には 1,352,681 個の化合物が登録されており，1 つの sdf ファイルで提供されている．本研究では，この sdf ファイルを化合物単位で分割し，それぞれのファイル名をその化合物の CHEMBL ID とした．ChEMBL 18 の原子数の分布を図 4 に示す．原子数は 100 以下のものが大半を占めており，最大値は原子数 26 で化合物は 72,170 個存在している．

4.2 計算機環境

本研究で用いた計算機環境を表 1 に示す．

表 1　利用したノードのスペック

CPU	Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz (8 コア)×2 ソケット
メモリ	64GB

4.3 評価方法

ChEMBL 18 の化合物の sdf ファイルの中から，原子数 10，20，30，40，50 の化合物をランダムで 9 つ選び，それぞれに関して従来手法（KCOMBU[9]），提案手法 1（フィルターの改良のみ），提案手法 2（外部原子数情報ファイルの利用による I/O 削減），提案手法 3（原子数情報のファイル名への追加による I/O 削減）を用いて ChEMBL 18 の全化合物に対する類似化合物探索を行った．本研究では類似度計算の閾値 P を 0.7 とし，MCS の取り方は C-MCS として実験を行った．

4.4 結果と考察

実験結果を図 5 に示す．全ての手法で実行結果は同一であることを確認している．

本研究の提案したフィルタリング手法によって，従来手法に比べて提案手法 1 は平均 1.41 倍の高速化を達成した．このことから，従来手法はフィルタリング自体にも計算時間がかかってしまっていることが伺える．また，提案手法 2 および提案手法 3 による I/O 削減によって，従来手法に比べて平均で 4.21 倍，最大で 14.3 倍 (CHEMBL1775001，提案手法 3) の速度向上が得られた．提案手法 2 と提案手法 3 の比較では，提案手法 3 の方が平均 1.43 倍高速であった．

また，どの原子数においても提案手法によって速度が向上しているが，特に原子数 10 の CHEMBL1775001 はその変化が顕著である．これはクエリの原子数が小さいため，提案手法 2 や提案手法 3 によってほとんどの化合物ファイルを開く必要がなくなったためである．原子数 10 で閾値 P を 0.7 に設定した場合，$P\times10$ と $10/P$ より，原子数 6 以下および原子数 15 以上の化合物は全てファイルを開かずにフィルターアウトすることができる．原子数 6 以下および原子数 15 以上の化合物数は 1,316,056 個あり，全体の 97%に相当する．

さらに，同じ原子数でも計算時間が大きく異なる場合があることが確認できる．即ち，原子数が同じでも原子の種類によって計算時間が変わることがあるということである．原子数 30 の CHEMBL1093061 と CHEMBL1210081 については CHEMBL1093061 の方が高速に計算できているが，計算結果を確認すると CHEMBL1093061 の類似化合物は自身を含め 2 個しかなかった．このことから，珍しい原子構成を有する化合物は MCS を求める計算時間は小さくなると言える．

Vol.2015-MPS-103 No.61
Vol.2015-BIO-42 No.61
2015/6/25

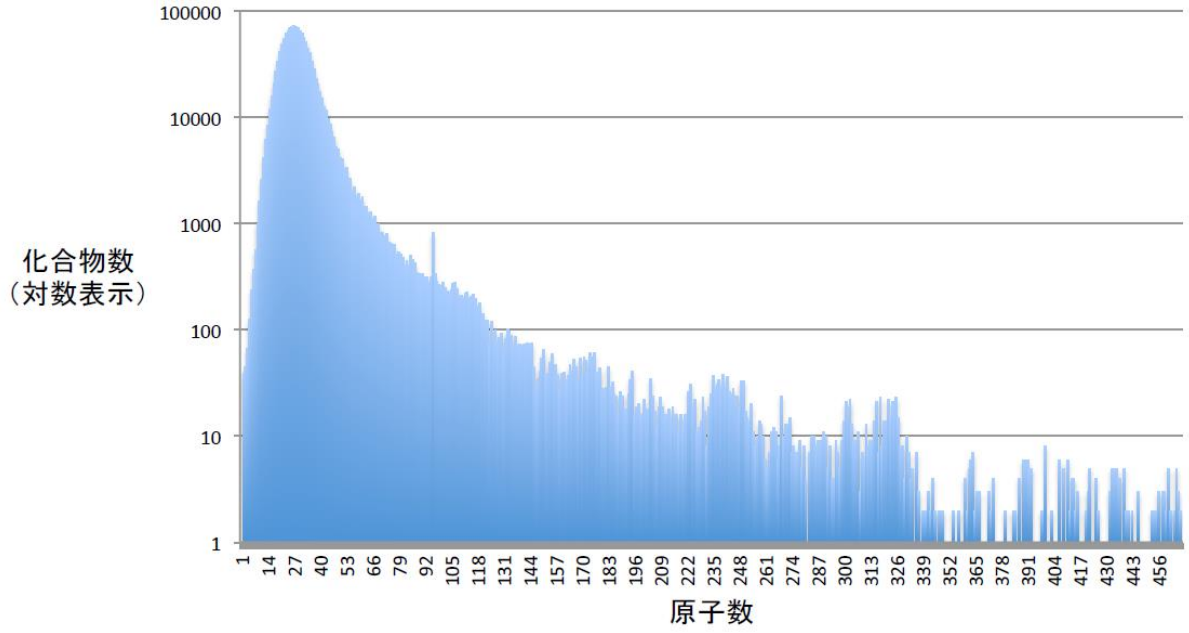

図 4　ChEMBL 18 の原子数ごとの化合物数分布

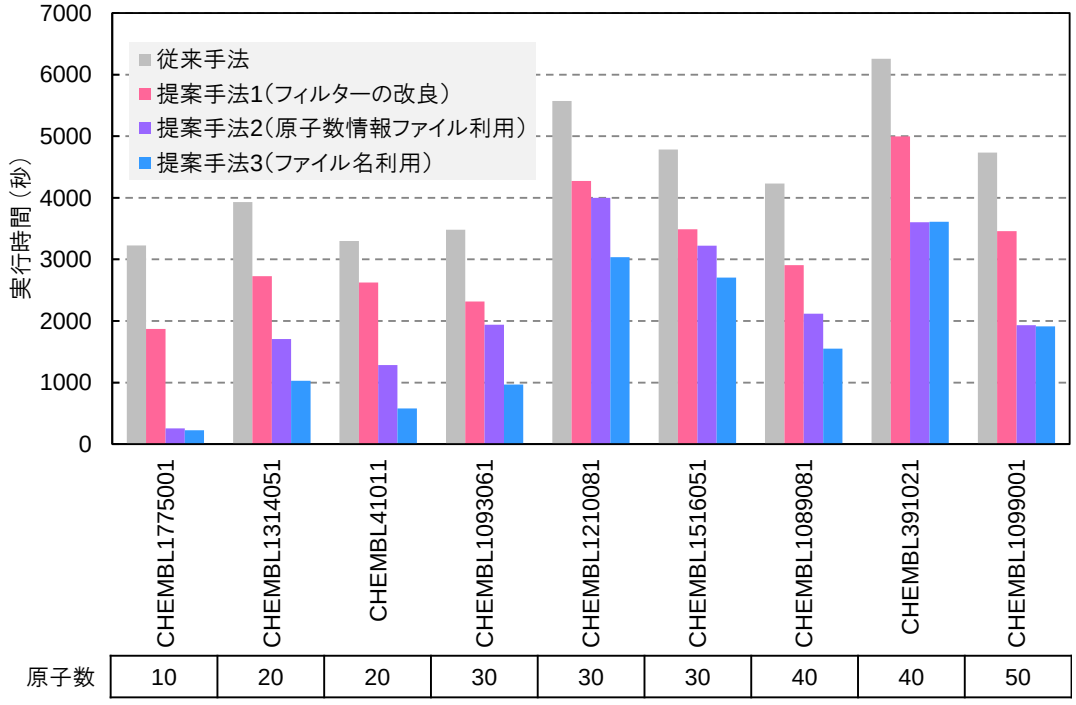

図 5　ChEMBL 18 データベースの全化合物に対する類似化合物探索の計算時間の比較.

5. 結論

本研究はデータベースに対する類似化合物の探索を高速化することを目的とし，化合物同士の類似度計算のフィルタリングによって計算対象の化合物の絞り込みを行う手法を提案した．データベースの化合物ファイルのファイル名にあらかじめ原子数を付与しておく前処理によって，化合物類似度の検索を最大で 14 倍高速化することに成功した. 今後の課題として，KCOMBU 以外の化合物類似度計算ツールとして[12][13]に対する実装が挙げられる.

Vol.2015-MPS-103 No.61
Vol.2015-BIO-42 No.61
2015/6/25

謝辞 本研究を遂行するにあたり，フィルタリング手法の開発や本稿の執筆に関する助言を頂いた東京工業大学 大学院情報理工学研究科 計算工学専攻 大上雅史博士に深く感謝する．本研究の一部は日本学術振興会 科研費 基盤研究(A) (24240044) の支援を受けて行われた．

参考文献

[1] Sheridan RP, Kearsley SK. Why do we need so many chemical similarity search methods? *Drug Discov Today*, 7(17):903–911, 2002.

[2] Bento AP, Gaulton A, Hersey A, Bellis LJ, Chambers J, Davies M, Krüger FA, Light Y, Mak L, McGlinchey S, Nowotka M, Papadatos G, Santos R, Overington JP. The ChEMBL bioactivity database: an update. *Nucleic Acids Res*, 42: D1083–1090, 2014.

[3] ChEMBL, https://www.ebi.ac.uk/chembl/

[4] Raymond JW, Willett P. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J Comput Aided Mol Des*, 16(7):521–533, 2002.

[5] Barnard JM. Substructure searching methods: old and new. *J Chem Inf Comput Sci*, 33, 532–538, 1992.

[6] Hagadone TR. Molecular substructure similarity searching: efficient retrieval in two-dimensional structure databases. *J Chem Inf Comput Sci*, 32, 515–521, 1992.

[7] Hattori M, Okuno Y, Goto S, Kanehisa M. Development of a chemical structure comparison for integrated analysis of chemical and genetic information in the metabolic pathway. *J Am Chem Soc*, 125: 11853–11865, 2003.

[8] Berglund AE, Head R. D.PZIM: A method for similarity searching using atom environments and 2D alignment. *J Chem Inf Model*, 50: 1790–1795, 2010.

[9] Kawabata T. Build-Up Algorithm for Atomic Correspondence between Chemical Structures. *J Chem Inf Model*, 51: 1775–1787, 2011.

[10] Law V, Knox C, Djoumbou Y, Jewison T, Guo AC, Liu Y, Maciejewski A, Arndt D, Wilson M, Neveu V, Tang A, Gabriel G, Ly C, Adamjee S, Dame ZT, Han B, Zhou Y, Wishart DS. DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Res*, 42: D1091-1097, 2014.

[11] DRUGBANK Open Data Drug & Drug Target Database, http://www.drugbank.ca

[12] Englert P, Kovács P. Efficient Heuristics for Maximum Common Substructure Search. *J Chem Inf Model*, Article ASAP, 2015.

[13] EPAM. Indigo Toolkit, http://lifescience.opensource.epam.com/indigo/