Vol.2013-MPS-94 No.5
2013/7/22

タンデム質量分析ソフトウェアCoCoozoのマルチコアCPUとGPUを用いた高速化

小幡 康文[1,a)]　石田 貴士[2,b)]　夏目 徹[3,c)]　秋山 泰[2,d)]

概要：生命科学や創薬などの分野において，タンパク質を同定する手法の一つに，タンデム質量分析があるが，近年，機器の発展やデータベースの増大によって，質量分析において計算機による解析が律速となりつつある．そこで本研究では，この問題に対処するために，質量分析スペクトル解析ソフトウェア CoCoozo を対象にして，高速化を目的に改良を行った．本研究では，アルゴリズムの改良に加え，マルチスレッド化，GPGPU 化を行い，プレカーサ情報完備の場合の解析について，アルゴリズムの改良だけで，CPU でも従来と比べて 8.9 倍の高速化を実現した．さらに，プレカーサ情報が欠落した場合の解析においては，12 CPU コアを用いた場合で，従来に比べて 15.9 倍，それに加えて GPU を用いた場合で，従来と比べて 18.1 倍の高速化を実現した．

キーワード：質量分析，MS/MS，CoCoozo，マルチスレッド化，GPGPU

Acceleration of Tandem Mass Spectra Analysis Software CoCoozo using Multi-core CPUs and Graphics Processing Units

Yasufumi Obata[1,a)]　Takashi Ishida[2,b)]　Tohru Natsume[3,c)]　Yutaka Akiyama[2,d)]

Abstract: Tandem mass spectrometry, a method involving multiple steps of mass spectral selection, is widely used in various biological fields. In recent years, steady improvements have been made with respect to speed, and the number of protein databases available for analysis has rapidly increased. Consequently, computational analysis has become the bottleneck in tandem mass spectrometry. To overcome this problem, we attempted to improve the tandem mass spectrometry analysis software CoCoozo. To accelerate the program, we improved the algorithm and also incorporated utilization of multi-core CPU and GPGPU. As a result, when all mass spectral data files had precursor data, we achieved 8.9-fold speedups compared with the original software. In addition, in the case of no precursor data, by using a 12-core CPU and a GPU card we achieved 18.1-fold speedups compared with the original software.

Keywords: Mass Spectrometry, MS/MS, CoCoozo, Multi-threading, GPGPU

1. 導入

サンプル中に含まれるタンパク質を同定する方法として，現在多くの研究で，質量分析が用いられている [1]．この質量分析を用いた研究の例としては，癌やアルツハイマー病などの病気に特異的なタンパク質の解析 [2], [3] や，タンパク質間相互作用の研究などが挙げられる [4].

[1] 東京工業大学 工学部 情報工学科,
Department of Computer Science, Faculty 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] 産業技術総合研究所 創薬分子プロファイリング研究センター,
Molecular Profiling Research Center for Drug Discovery, National Institute of Advanced Industrial Science and Technology
[a)] obata@bi.cs.titech.ac.jp
[b)] t.ishida@bi.cs.titech.ac.jp
[c)] t-natsume@aist.go.jp
[d)] akiyama@cs.titech.ac.jp

現在，質量分析で広く用いられる方式に，タンデムマス法，もしくはタンデム質量分析（Tandem Mass Sectrometry）と呼ばれる方式がある．タンデムマス法は，通常の質量分析の計測を 2 回以上行う手法であり，その中でも，2 段階のものが現在広く用いられている．2 段階のタンデムマス法では，1 回目の計測時に分析対象のタンパク質を断片化・イオン化し，その質量電荷比（mass-to-charge ratio, m/z）とその強度（存在量，intensity）を計測する．この断片化されたペプチド鎖をプレカーサ（precursor）と呼び，2 回目の計測時に，各プレカーサは更に断片化・イオン化され，その質量電荷比と強度が計測される．また，このプレカーサから断片化されたペプチド鎖をフラグメント（fragment）と呼ぶ．

タンデムマス法によって得られる情報である質量分析スペクトルは，ソフトウェアを用いて解析され，タンパク質が同定される．これまで様々な質量分析ソフトウェアが開発されており，その中でも Mascot[5] が現在広く用いられているが，その他にも，SEQUEST[6] や SpectraST[7]，CoCoozo といったソフトウェアが開発されており，それぞれが独自のアルゴリズムを用いている．CoCoozo は，産業技術総合研究所及び東京工業大学秋山研究室で 2000 年代中頃から開発されてきた質量分析スペクトル解析システムであり，特徴として，適合率（precision）の高い解析を行うことが可能である点や，独自の誤差補正機能を搭載しているという点が挙げられる．また，CoCoozo は，過去数年間，産業技術総合研究所で実際に利用されている．

近年，質量分析に関する測定技術の向上の結果，検出されるデータの量も増加し，データの質も上昇している．それに伴い，コンピュータによる解析は，質量分析においてボトルネックとなりつつある．その一方で，近年では計算機ノード自体の処理性能も飛躍的に向上し，加えて CPU のマルチコア化に伴う並列化や，GPU（Graphics Processing Units）を用いた汎用演算である，GPGPU（General-purpose computing on graphics processing units）など，様々な高速化技術が開発されている．近年では GPGPU は様々な研究分野で用いられており，天体計算 [8] や高速フーリエ変換 [9] 等で大幅な高速化を達成している．

本研究では，質量分析スペクトル解析ソフトウェア CoCoozo を対象に，アルゴリズムの改良を行い，ノード内並列化，および GPGPU 化による高速化を図った．CoCoozo は，すでに MPI を用いて，複数ノードを同時に用いて並列に動作させることが可能であるが，本研究においては，1 ノードにおける高速化を目指した．

2. CoCoozo

まず，本研究のベースとなった質量分析スペクトル解析ソフトウェア CoCoozo の詳細について述べる．そのメインプロセスのフローチャートを図 1 に示す．CoCoozo は，

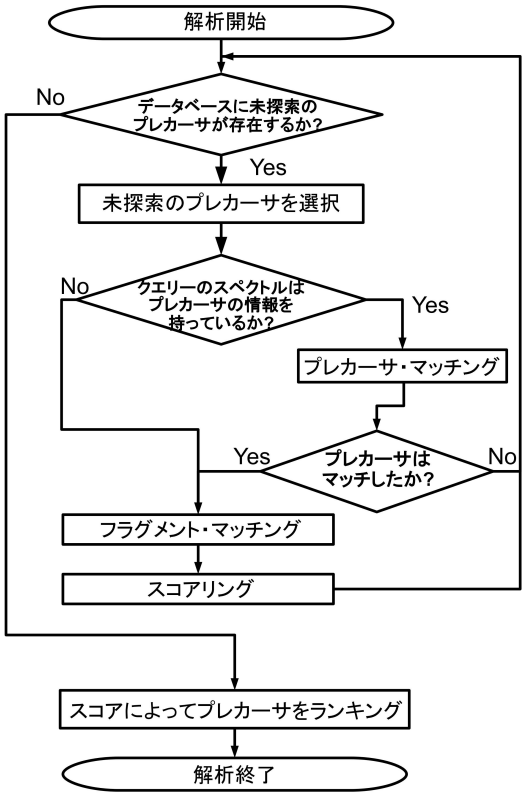

図 1 CoCoozo のメインプロセスのフローチャート（1 スペクトルデータファイル）

Fig. 1 CoCoozo Main Process Flowchart (for a mass spectral data)

このフローチャートの主な処理をデータベース側の各プレカーサに対して行っている．まず，最初にクエリーとなるスペクトルファイルにプレカーサの測定結果の情報があるかどうかを判定する．もし，スペクトルファイルにプレカーサの情報がある場合は，次に “プレカーサ・マッチング” と呼ばれる処理を行う．プレカーサ・マッチングでは，クエリーのプレカーサデータとデータベース側のプレカーサデータの比較が行われる．もし，マッチしていると判断されれば，それに続いて “フラグメント・マッチング” と呼ばれる処理が行われる．これに加えて，クエリーのスペクトルファイルにプレカーサの測定結果の情報が欠落している場合にも，プレカーサ・マッチングを行うことなく，すぐにフラグメント・マッチングが行われる．フラグメント・マッチングでは，クエリーとなるスペクトルファイルの各スペクトルデータとデータベース側のフラグメントのスペクトルデータの類似度の計算が行われる．その後，フラグメント・マッチングの結果を用いて，データベースのプレカーサに対して，スコア付けが行われる．データベース側の全てのプレカーサに対して，このマッチングとスコアリングを行った後，スコアの値を基にプレカーサをランク付けし，高いスコアであったプレカーサを解析結果として出力する．

2.1 CoCoozo のボトルネック

本研究では高速化を試みるにあたり，CoCoozo のボトルネックを明らかにするため，まず 2 つの場合について CoCoozo の各関数の消費時間を計測した．1 つ目の場合は，全てのスペクトルファイルにプレカーサの測定結果が含まれている場合（プレカーサ情報完備の場合），もう一つの場合が，全スペクトルファイル中の約 10% のスペクトルファイルがプレカーサの測定結果を欠いている場合である（プレカーサ情報欠落の場合）．この計測結果より，CoCoozo において，主に時間を消費する部分が，プレカーサとフラグメントのマッチングの部分であるということが分かった．加えて，プレカーサ情報欠落の場合，プレカーサ情報完備の場合と比べて，解析に約 13 倍の時間がかかっていることも分かった．

3. 提案手法

3.1 マッチングアルゴリズムとスコア初期化のアルゴリズム改良

律速となっているプレカーサ・マッチングとフラグメント・マッチングのアルゴリズムの改良を行った．ソートしたデータを用いてマッチングを行うことで，トレランスから外れた場合，すぐにマッチング処理を終了することが可能となるが，CoCoozo は翻訳後修飾に対応するための質量補正によって，ソートが困難なプログラム設計となっていた．そこで，翻訳後修飾に対応したまま，質量に応じてデータをソートすることができるように，プログラムのデータ形式から変更を行い，ソートを実現した．

また，マッチングアルゴリズムの改良に加えて，改良前のスコア初期化は冗長な処理が多かったため，それを省略して効率化を図った．

3.2 マルチスレッド化

プレカーサ情報の無いスペクトルデータファイルの解析の際のフラグメント・マッチングとスコアリングの部分をマルチスレッド化し，高速化することを目指した．フラグメント・マッチングとスコアリングの処理は連続した処理で，一連の処理で同じプレカーサから生成されるフラグメントについてのみ処理を行う．そのため，これらの処理は，ほかのプレカーサから生成されるフラグメントに対するフラグメント・マッチングとスコアリングからは独立した処理である．そこで，各スレッドが，それぞれ別のプレカーサから生成されるフラグメントについてマッチング処理とスコアリングを行うように改良する．

3.3 GPGPU による高速化

マッチング・アルゴリズムの改良後も，プレカーサ情報を持たないスペクトルデータファイルに対するフラグメント・マッチングは，処理時間の約 70 %を 占める処理であった．そのため，この処理を GPGPU を用いて高速化することを試みた．フラグメント・マッチングにおいて，データベースのあるフラグメントのデータと，クエリーとなるスペクトルデータファイルのフラグメントの 1 スペクトルとの比較処理は，他の比較処理とは独立している．このため，今回 GPGPU を比較の部分に導入した．しかし，フラグメント・マッチング全体では，判断文を多用することや前の結果に依存した処理があることなどから，全体を GPU 上で効率的に実行することは難しいと考えられる．そのため，GPU では最大幅のトレランスによるマッチングを行い，その結果を用いて CPU 側で正確なマッチングを少数回行うことで，高速化を目指した．更に，GPGPU 化に加えて，GPU での処理の後に行われる CPU での処理について，マルチスレッド化を施した．また，GPU 化には NVIDIA 社の CUDA を用いた．

4. 結果と考察

4.1 質量分析スペクトルデータと配列データベース

解析対象となるスペクトルデータファイルの総数は，1,486 ファイルである．全てのスペクトルデータファイルがプレカーサの計測結果の情報を完備しているデータセット（プレカーサ情報完備の場合）と，その内 149 ファイルから，人工的にプレカーサ情報を削除したデータセット（プレカーサ情報欠落の場合）を作成した．

データベースに含まれるデータ件数は，タンパク質が 38,415 エントリ，プレカーサが 857,298 エントリ，フラグメントが 26,489,468 エントリである．

4.2 実験環境

実験は，東京工業大学のスーパーコンピュータ TSUBAME2.0 の Thin ノードで実行した．詳しい実験環境について表 1 に示す．

表 1 実験環境
Table 1 Computing Environment

CPU	Intel Xeon 2.93 [GHz]（6 cores）x 2
Memory	54 [GB]
OS	SUSE Linux Enterprise Server 11 SP1
GPU	NVIDIA Tesla M2050
CUDA	CUDA 4.1（64bit）

時間の計測には UNIX の “`time`” コマンドを使用し，より詳しい解析を行うプロファイリングには Intel VTune Amplifier XE 2011 を使用した．

4.3 マッチングアルゴリズムとスコア初期化のアルゴリズム改良の結果

表 2 は，プレカーサ情報完備の場合の実行時間の結果で

ある．アルゴリズム改良によって，オリジナルの CoCoozo と比べて全体で約 8.9 倍の高速化を達成した．特に，プレカーサ・マッチングでは，オリジナルの CoCoozo と比べて約 65.3 倍の高速化を達成し，スコア初期化は，オリジナルから約 483.1 倍の高速化を達成している．

表 2 プレカーサ情報完備の場合の結果
Table 2 Results of Improvements in the case of complete precursor data

	時間 [sec]	速度向上比
オリジナル	**609.23**	
- プレカーサ・マッチング	(443.0)	
- フラグメント・マッチング	(27.61)	
- スコア初期化	(72.46)	
アルゴリズム改良版	**68.80**	**8.9 倍**
- プレカーサ・マッチング	(6.63)	(65.3 倍)
- フラグメント・マッチング	(11.08)	(2.5 倍)
- スコア初期化	(0.15)	(483.1 倍)

4.4 マルチスレッド化の結果

表 3 は，プレカーサ情報欠落の場合のマルチスレッド化及び GPGPU 化による高速化の結果を表した表である．アルゴリズムの改良により改良前と比べて約 3.0 倍の高速化を達成し，更に CPU12 スレッドで，オリジナルと比較して 15.9 倍の高速化を達成した．

表 3 プレカーサ情報欠落の場合の結果
Table 3 Results of Improvements in the case of incomplete precursor data

	時間 [sec]	速度向上比
オリジナル	7752.82	
アルゴリズムの改良	2589.52	3.0 倍
マルチスレッド (12-スレッド)	488.30	15.9 倍
GPGPU	1302.57	6.0 倍
マルチスレッド (12-スレッド) & GPGPU	427.97	18.1 倍

4.5 GPGPU による高速化の結果

さらにアルゴリズム改良に加えて GPGPU を導入した場合の結果も表 3 に示した．GPGPU を導入した場合，オリジナルと比較して約 6.0 倍の高速化を実現した．特に，GPGPU を一部の処理に導入したフラグメント・マッチングの部分のみを比較すると，GPGPU 導入前のアルゴリズム改良版より約 13.8 倍の高速化を実現した．また，アルゴリズムの改良と GPGPU に加えて，CPU マルチスレッド化を実施し，CPU12 スレッドで実行した場合，オリジナルと比較して約 18.1 倍の高速化を実現した．

5. 結論

本研究では，質量分析スペクトル解析システムである CoCoozo の改良を行い，クエリであるピークファイルにプレカーサの情報が完備されている場合において，約 8.9 倍の高速化に成功した．また，ピークファイルの一部でプレカーサの情報がない場合においては，アルゴリズムの改良によって約 3 倍の高速化を達成し，マルチスレッド化によって，12 スレッドの場合，改良前から 15.9 倍の高速化に成功した．また，GPGPU を導入した場合，改良前と比べて，その他の改良とあわせて，約 18.1 倍の高速化に成功した．これらの高速化によって解析結果はほとんど変化せず，従来までとほぼ同等の結果を高速に得ることが可能となった．

謝辞
質量分析に関して様々な助言を賜りました，CoCoozo の開発グループのメンバーである，産業技術総合研究所 小池 克幸氏，草野 秀男氏，八田 知久氏にお礼申し上げます．また，CoCoozo の開発当時の主たるプログラマとして，様々な疑問に答えていただきました，高度情報科学技術研究機構の藤原 康広氏にお礼申し上げます．

参考文献

[1] W.P.Blackstock and M.P.Weir: Proteomics: quantitative and physical mapping of cellular proteins, *Trends Biotechnol.*, Vol. 17, No. 3, pp. 121–127 (1999).
[2] E.P.Diamandis: Mass spectrometry as a diagnostic and a cancer biomarker discovery tool: opportunities and potential limitations, *Mol Cell Proteomics.*, Vol. 3, No. 4, pp. 367–378 (2004).
[3] D.C.German, P.Gurnani, A.Nandi, H.R.Garner, W.Fisher, R.Diaz-Arrastia, P.O'Suilleabhain and K.P.Rosenblatt: Serum biomarkers for Alzheimer's disease: proteomic discovery, *Biomed Pharmacother.*, Vol. 61, No. 7, pp. 383–389 (2007).
[4] A.C.Gavin, K.Maeda and S.Kühner: Recent advances in charting protein-protein interaction: mass spectrometry-based approaches, *Curr Opin Biotechnol.*, Vol. 22, No. 1, pp. 42–49 (2011).
[5] D.N.Perkins, D.J.Pappin, D.M.Creasy and J.S.Cottrell: Probability-based protein identification by searching sequence databases using mass spectrometry data, *Electrophoresis.*, Vol. 20, No. 18, pp. 3551–3567 (1999).
[6] J.K.Eng, A.L.McCormack and J.R.Yates, I.: An Approach to Correlate Tandem Mass Spectral Data of Peptides with Amono Acid Sequences in a Protein Database, *J. Am. Soc. Mass Spectrom.*, Vol. 5, No. 11, pp. 976–989 (1994).
[7] H.Lam, E.W.Deutsch, J.S.Eddes, J.K.Eng, N.King, S.E.Stein and R.Aebersold: Development and validation of a spectral library searching method for peptide identification from MS/MS, *Proteomics*, Vol. 7, No. 5, pp. 655–667 (2007).
[8] Nguyen, H.: *GPU Gems 3*, Addison-Wesley Professional, Boston (2007).
[9] N.K.Govindaraju, B.Lloyed, Y.Dotsenko, B.Smith and J.Manferdelli: High Performance Discrete Fourier Transforms on Graphics Processors, *the 2008 ACM/IEEE conference on supercomputing*, pp. 1–12 (2008).