

厚生労働科学研究費補助金

政策科学総合研究事業

(臨床研究等 ICT 基盤構築・人工知能実装研究事業)

多施設 SS-MIX2 標準化データベースからの臨床的表現型クラスタリングと  
その臨床エビデンス創出手法の開発研究

令和元年度 総括研究報告書

研究代表者 大江 和彦

令和2年(2020年) 5月

## 目 次

### I. 総括(分担) 研究報告

多施設SS-MIX2標準化データベースからの臨床的表現型クラスタリングと  
その臨床エビデンス創出手法の開発研究 ----- 1

東京大学医学部附属病院 教授 大江 和彦

II. 研究成果の刊行に関する一覧表 ----- 14

厚生労働科学研究費補助金  
(政策科学総合研究事業 (臨床研究等 ICT 基盤構築・人工知能実装研究事業))  
総括研究報告書

多施設 SS-MIX2 標準化データベースからの臨床的表現型クラスタリングと  
その臨床エビデンス創出手法の開発研究

研究代表者 大江 和彦 東京大学医学部附属病院・教授

【研究要旨】

目的と方法：電子カルテ由来の SS-MIX2 標準化多施設臨床データベース (DB) を使用して、1 年目：①教師なし機械学習による自動クラスタリング等の手法により、臨床的表現型において共通特性をもつ集団 (クラスタ) を多数自動生成し、②得られたクラスタの他の臨床情報特性を時系列変化を含めて類型化の手法を検討する。

2 年目：③その臨床的特性の出現確率等の統計的特性やその臨床的意味付けを分析し、④診療中の患者の電子カルテデータから上記クラスタに自動分類し、その結果にもとづいた臨床的特性を可視化することの臨床的有用性を評価する。

結果と考察：電子カルテ由来の SS-MIX2 標準化多施設臨床データベース (DB) を使用して、教師なし機械学習による自動クラスタリング等の手法で分析するための、臨床的表現型において共通特性をもつデータセットを、a) 血液系疾患 (D50-D77)、b) 免疫系疾患 (D80-D89)、c) 内分泌代謝系疾患 (E00-E87)、d) 高血圧疾患 (I10-I15)、e) 心不全 (I50)、f) 炎症性関節炎 (M05-M14)、g) 結合織障害 (M30-M36)、h) 腎糸球体・腎機能障害疾患 (N00-N19) に分けて、作成方針の確立と自動作成環境の開発を行った。試験データの抽出では、1 病院分の検体検査件数で 800 万件以上、検体検査種別で 300 以上、1 患者あたりの件数は多いもので 1700-2000 件であった。この 1 施設分の分析用データセットを教師なし機械学習のクラスタリング手法 K-Means++により 7 つの疾患グループに、さらに疾患グループごとにその ICD10 の 4 桁目 (細分類) を想定して 8 グループ程度を設定してクラスタリングを行った。K-Means++法以外にも Elbow 法等を用いたクラスタ数の決定を行いその効果を検討するとともに、クラスタごとの臨床的特性を取得する。

研究分担者		松村泰志・大阪大学	教授
中山雅晴・東北大学	教授	津本周作・島根大学	教授
近藤克幸・秋田大学	理事	中島直樹・九州大学	教授
白鳥義宗・名古屋大学	病院教授	関 倫久・東京大学	助教
木村通男・浜松医科大学	教授		

## A. 研究目的

**背景：**臨床エビデンスは、「高血圧合併2型糖尿病」のように特定の特性を有する患者集団を事前規定し、「阻害薬が有効」のようにその集団における別の臨床特性の存在を確認することで得られる。クリニカルクエスチョン(CQ)を思いつかなければ事前に集団を規定できず、存在を確認すべき臨床特性が不明で研究デザインができない。臨床の場では、患者の臨床特性で規定される集団が、どのような別の臨床特性を有するかを知りたいことが多いが、具体的なCQを思いつかないことが多く、DB駆動型のCQ自動生成、エビデンス示唆を得る手法の開発が必要である。

**研究経緯：**申請者が代表のAMED「医用知能情報システム基盤の研究開発」(2015.10～2019.3)において統計解析可能な多施設臨床DB(120万症例以上)を構築済みで、倫理審査が完了しており申請者や分担者らが利用可能となっている。また同研究では津本(分担者)らが時系列データマイニングにより時間経過情報を含めて類型化する手法を開発してきた。

**目的：**本研究では、電子カルテ由来のSS-MIX2標準化多施設臨床データベース(DB)を使用して、

1年目：①教師なし機械学習による自動クラスタリング等の手法により、臨床的表現型において共通特性をもつ集団(クラスタ)を多数自動生成し、②得られたクラスタの他の臨床情報特性を時系列変化を含めて類型化の手法を検討する。

2年目：③その臨床的特性の出現確率等の統計的特性やその臨床的意味付けを分析し、④診療中の患者の電子カルテデータから上

記クラスタに自動分類し、その結果にもとづいた臨床的特性を可視化することの臨床的有用性を評価する。

## B. 研究方法

1) 分析用データセットの作成環境の構築  
本研究では、このデータベースを使用し、初年度の教師なし機械学習による自動クラスタリングを実施するための分析用データセットの作成手法を確立するため、まず研究代表者の所属する1施設分のデータを用いたパイロット的なデータ分析を経て、以下の手順で分析用データセットを作成することとした。すなわち、病名データで以下のICD10コードの確定診断を有する7つの患者集団をICD10コードとともに抽出した。

- a) 血液系疾患(D50-D77)、b) 免疫系疾患(D80-D89)、c) 内分泌代謝系疾患(E00-E87)、d) 高血圧疾患(I10-I15)、e) 心不全(I50)、f) 炎症性関節炎(M05-M14)、g) 結合織障害(M30-M36)、h) 腎糸球体・腎機能障害疾患(N00-N19)。

これらを選択したのは、これらの疾患群では疾患相互および疾患内の血液検査結果のパターンだけによってもその集団特性を表現できる可能性があるのに対して、感染症、腫瘍性疾患、精神疾患、消化管炎症性疾患、外傷等はこの可能性が低いという理由による。

その上で、それぞれの患者集団における個々の疾患存在期間(診断開始日ー終了日)内において、全体で10万件以上の検査実施数がある検査項目(約120項目を対象として探索的に決定)に含まれる血液検査結果定量値を抽出した上で、同一患者ごとに連続した6ヶ月ウインドウ期間における各検

査値の平均値、最小値、最大値、分散を変数値とし、その6ヶ月ウインドウを当該患者ごとの疾患存在期間内ですらして作成し異なる患者状態とみなした。次に同じ期間における投与医薬品のATC分類コード(医薬品の国際的な効能成分分類)粒度のデータを抽出した。なお検査値については上記の定量値、医薬品については投与の有無のデータに変換した。参考までに付録1に検査結果抽出に関するPythonプログラムの主要部分を示す。

以上の分析用データセットの生成プログラムをPythonで作成し、施設を指定して自動的に分析用データセットを生成する環境が構築できた。本報告作成時点では、この手法による分析用データセットの作成は研究代表者の所属する1施設分で行った。今後、他の7施設のデータでも実施した上で、これらのデータを疾患ICD10ごとに統合する。

## 2) 教師なし機械学習のクラスタリング

上記の1施設分の分析用データセットを教師なし機械学習のクラスタリング手法であるK-Means++によりクラスタリングの試行をPython scikit-learnライブラリを用いて実施した。K-Means++は最初にクラスタ数を設定する必要があり、前記全データについて血液検査結果だけで7つの疾患グループに、さらに疾患グループごとにそのICD10の4桁目(細分類)を想定して8グループ程度を設定してクラスタリングを行った。初期値をいくつか変えて何度か実施して生成されるクラスタリングの結果と、疾患グループおよびICDの細分類とを比較した場合の一致性がどの程度見られるかについてまず検討した。

## C. 結果と考察

研究代表者の病院分での a) 血液系疾患(D50-D77)、b) 免疫系疾患(D80-D89)、c) 内分泌代謝系疾患(E00-E87)、d) 高血圧疾患(I10-I15)、e) 心不全(I50)、f) 炎症性関節炎(M05-M14)、g) 結合織障害(M30-M36)、h) 腎糸球体・腎機能障害疾患(N00-N19)、各検体検査件数は約800万件であった。また患者別の検体検査実施件数は1700-2000件あるものが見られた。

検体検査の項目数はまれに検査するものを含めると300項目を超えるため、1) 末梢血血液検査、血糖関係、凝固系、2) 生化学、3) 免疫系、4) ウイルスマーカー、5) 血液ガス、などの区分に分け、区分ごとにデータセットを分割する必要があると考えられた。図1に、1) 末梢血血液検査、血糖関係、凝固系の場合の項目セットを示す。

本研究は、多施設臨床DBを教師なし機械学習による自動クラスタリング等の手法により、共通特性をもつクラスタを多数自動生成し、自動的にその集団における未知の臨床特性を得ることによって、思いつかないCQ(Clinical Question)やエビデンスの生成の鍵を得ることができる点が特色である。

大規模臨床DBから、気づかれていない臨床的表現型クラスタを自動識別し、臨床経過等の特性と確率情報を臨床エビデンスとして自動創成し、臨床現場で実際の患者にリアルタイムに近い適用する手法が開発される。これにより、たとえば標準臨床ガイドラインの策定において、1基礎疾患と2つ程度の合併疾患を有するような比較的シンプルなケースだけでなく、より多数の臨床的パラメータによる複雑な臨床的表現型の

クラスタに属するケース（いくつもの合併疾患を有し、いくつかの治療経歴を有する複雑な臨床経過をたどったケース群など）ごとに細分化した臨床ガイドラインを生成することができる可能性がある。集団の細分化をすることが、ガイドラインを適用すべき患者集団の規定を詳細化することに繋がり、個別化医療に近づくことになる。細分化された基準では、患者がどの集団に属するかの判定をコンピュータシステムに委ねなければ判定できない状況になることによりガイドライン準拠を推進する新たな ICT 手段を有することになり、標準的臨床ガイドラインの適用のあり方と普及推進に関する施策に貢献できる可能性がある。また、臨床中核拠点病院における臨床データベース駆動型の臨床研究の推進施策や、電子カルテデータの二次利用データの品質改善・管理に関する研究事業の推進にも貢献するとともに、さらに機械学習による厚労省標準 SS-MIX2 の新しい活用事例になると考えられる。

1 年目の成果は、本報告作成時点では分析用データセットの作成方針の確立と自動作成環境の開発、およびその試行による教師なし機械学習のクラスタリングの探索的実施が、研究代表者のデータベースを使用して可能となったところまでである。

今後、得られたクラスタごとの結果分析を行い、その結果によっては、分析用データセットの作成方法の修正が必要と考えられれば修正を行うとともに、あらかじめクラスタ数を設定する K-Means++法以外のクラスタリング手法として、階層的クラスタリングも合わせて実施して結果を比較するなどを実施する必要がある。

2 年目の計画としては、

1) クラスタの他の臨床情報特性の時系列変化を含めた類型化

各クラスタにおける他の臨床検査値の陽性率、疾患特異性の高い医薬品投与状況、臨床経過の類型細分化、重症度の時間的推移、既知の診断情報などを、記述統計や時系列データマイニング等の手法により、3 年間程度の期間について解析し、クラスタごとの臨床的特性を取得する

2) クラスタにおける臨床的特性の出現確率等の統計的特性やその臨床的意味付けを分析

上記統計的特性とともに多次元ベクトル情報に変換し、上記臨床的特性と、それに対応する疾患・病態を検討し、論文や診断治療基準での既知臨床エビデンスと比較し、臨床医とディスカッションを行い、今回自動的に得られた臨床エビデンスの特徴や課題を明らかにする。

3) 診療中の患者の電子カルテデータから上記クラスタに自動分類し、その結果にもとづいた臨床的特性を可視化するシステムの開発とその臨床的有用性の評価を行う。といった手順を検討する。

## E. 結論

電子カルテ由来の SS-MIX2 標準化多施設臨床データベース (DB) を使用して、教師なし機械学習による自動クラスタリング等の手法で分析するための、臨床的表現型において共通特性をもつデータセットを、a) 血液系疾患 (D50-D77)、b) 免疫系疾患 (D80-D89)、c) 内分泌代謝系疾患 (E00-E87)、d) 高血圧疾患 (I10-I15)、e) 心不全 (I50)、f) 炎症性関節炎 (M05-M14)、g) 結合織障害 (M30-M36)、

h)腎糸球体・腎機能障害疾患(N00-N19)に分けて、作成方針の確立と自動作成環境の開発を行った。試験データの抽出では、1病院分の検体検査件数で800万件以上、検体検査種別で300以上、1患者あたりの件数は多いもので1700-2000件であった。この1施設分の分析用データセットを教師なし機械学習のクラスタリング手法 K-Means++により7つの疾患グループに、さらに疾患グループごとにそのICD10の4桁目(細分類)を想定して8グループ程度を設定してクラスタリングを行った。K-Means++法以外にもElbow法等を用いたクラスタ数の決定を行いその効果を検討するとともに、クラスタごとの臨床的特性を取得する。

#### F. 健康危険情報

#### G. 研究発表

##### 1. 論文発表

1. K Yamada, M Itoh, Y Fujimura, M Kimura, K Murata, N Nakashima, M Nakayama, K Ohe, T Orii, E Sueoka, T Suzuki, H Yokoi, C Ishiguro, Y Uyama on behalf of MID - NET project group: The utilization and challenges of Japan's MID - NET® medical information database network in postmarketing drug safety assessments: A summary of pilot pharmacoepidemiological studies. *Pharmacoepidemiology and Drug Safety* 28(5),601-608, May.
2. Hayakawa M, Imai T, Kawazoe Y, Kozaki K, Ohe K. Auto-Generated Physiological Chain Data for an Ontological Framework for Pharmacology and Mechanism of Action to Determine Suspected Drugs in Cases of Dysuria. *Drug Safety*. 2019, 42
3. Kagawa R, Shinohara E, Imai T, Kawazoe Y, Ohe K. Bias of Inaccurate Disease Mentions in Electronic Health Record-based Phenotyping. *International journal of medical informatics*. 2019;124: 90-96.
4. Nakashima N, Noda M, Ueki K, Koga T, Hayashi M, Yamazaki K, Nakagami T, Ohara M, Gochi A, Matsumura Y, Kimura M, Ohe K, Kang D, Toya Y, Yamagata K, Yokote K, Ikeda S, Mitsutake N, Yamamoto R, Tanizawa Y.: Recommended configuration for personal health records by standardized data item sets for diabetes mellitus and associated chronic diseases: A report from Collaborative Initiative by six Japanese Associations. *J Diabetes Investig*. 2019 May;10(3):868-875.
5. Seki T, Tamura T, Suzuki M, SOS-KANTO 2012 Study Group. Outcome prediction of out-of-hospital cardiac arrest with presumed cardiac aetiology using an advanced machine learning technique. *Resuscitation* 141 128-135 2019
6. 大江 和彦 ビッグデータと人工知能技術による診療支援システム *Dementia Japan(1342-646X)* 34 巻 1 号 Page70-75(2020.01), 解説

7. 大江 和彦 AI とビッグデータのための  
医療情報の標準化 医療機器学  
(1882-4978)89 巻 6 号 Page545-  
551(2019.12)

2. 学会発表

1. 早川 仁, 関 倫久, 河添 悦昌, 大江  
和彦 パスウェイデータベースを利用  
したグラフ畳み込み深層学習による悪  
性腫瘍の診断分類性能の検討 医療情  
報学連合大会論文集(1347-8508)39 回  
Page352(2019.11)

2. 関 倫久, 河添 悦昌, 大江 和彦 SS-  
MIX2 標準化ストレージを用いた入院後  
の死亡退院リスク予測モデルの開発  
医療情報学連合大会論文集(1347-  
8508) 第39回 Page249(2019.11), 国  
内, 口頭

H. 知的財産権の出願・登録状況

なし



表1 末梢血血液検査、血糖関係、凝固系の項目セットの例

JLAC10	項目名	JLAC10	項目名
2B14000002206201	D. Dダイマー_定量値_血漿	2A160000003460461	血液像_前骨髄球_血液塗抹標本
3D04600001920402	Hb A1c (NGSP)_全血(添加物入り)	2A160000003460456	血液像_単球_血液塗抹標本
3D01000002226101	グルコース_定量値_血漿	2A160000001966256	血液像_単球_全血(添加物入り)
4Z27100002202301	ヒト脳性Na利尿ポリペプチド_定量値_血漿	2A160000001930957	血液像-リンパ球_全血(添加物入り)
2B10000002231101	フィブリノーゲン_定量値_血漿	2A160135201930957	血液像-リンパ球_全血(添加物入り)
2B03000002231157	プロトロンビン時間_INR値_血漿	2A160000001930955	血液像-好塩基球_全血(添加物入り)
2B03000002231153	プロトロンビン時間_PT活性(%)_血漿	2A160135201930955	血液像-好塩基球_全血(添加物入り)
2B03000002231155	プロトロンビン時間_PT比_血漿	2A160000001930954	血液像-好酸球_全血(添加物入り)
2B03000002231151	プロトロンビン時間_被験血漿PT時間(秒)_血漿	2A160135201930954	血液像-好酸球_全血(添加物入り)
2B02000002231100	活性化部分トロンボプラスチン時間_血漿	2A160000001930951	血液像-好中球_全血(添加物入り)
2A16000003460477	血液像_その他_血液塗抹標本	2A160135201930951	血液像-好中球_全血(添加物入り)
2A16000003460457	血液像_リンパ球_血液塗抹標本	2A160000001930956	血液像-単球_全血(添加物入り)
2A160000001966257	血液像_リンパ球_全血(添加物入り)	2A160135201930956	血液像-単球_全血(添加物入り)
2A16000003460458	血液像_異型リンパ球_血液塗抹標本	2A160000001930106	血液像-破砕赤血球%_全血(添加物入り)
2A16000003460460	血液像_後骨髄球_血液塗抹標本	2Z010000001792001	血沈_定量値_血液(含むその他)
2A16000003460455	血液像_好塩基球_血液塗抹標本	2A990000001992057	末梢血液一般検査_MCH_全血(添加物入り)
2A160000001966255	血液像_好塩基球_全血(添加物入り)	2A990000001992058	末梢血液一般検査_MCHC_全血(添加物入り)
2A16000003460454	血液像_好酸球_血液塗抹標本	2A990000001992056	末梢血液一般検査_MCV_全血(添加物入り)
2A160000001966254	血液像_好酸球_全血(添加物入り)	2A990000001992054	末梢血液一般検査_ヘマトクリット_全血(添加物入り)
2A16000003460451	血液像_好中球_血液塗抹標本	2A990000001992053	末梢血液一般検査_ヘモグロビン_全血(添加物入り)
2A160000001966251	血液像_好中球_全血(添加物入り)	2A990000001992055	末梢血液一般検査_血小板数_全血(添加物入り)
2A16000003460453	血液像_好中球分葉核_血液塗抹標本	2A990000001992051	末梢血液一般検査_赤血球数_全血(添加物入り)
2A16000003460452	血液像_好中球桿状核_血液塗抹標本	2A990000001992052	末梢血液一般検査_白血球数_全血(添加物入り)
2A16000003460462	血液像_骨髄芽球_血液塗抹標本	2A110000001966202	網赤血球数_構成比_全血(添加物入り)
2A16000003460459	血液像_骨髄球_血液塗抹標本		

```

1 #coding: Shift_JIS
2
3 # コマンド引数で指定するICD 1 0 抽出条件にマッチするICD病名をもつ（疑いを除
- く）患者を検索し、
4 # その患者IDに関する検査結果テーブル名を取得して、その検査結果テーブル名に登
- 録されている
5 # 検査結果数がおおよそ指定件数以上である検査の場合には、その患者の検査結果レコ
- ードを抽出し
6 # CSV用に出力する
7 # 数値検査だけ出力する
8 # 検体材料として静脈血だけを対象とする
9
10
11 import os
12 import psycopg2
13 from psycopg2 import sql
14 import sys # モジュール属性 argv を取得するため
15 from datetime import datetime
16
17 # 患者IDからその患者の検査結果を含むすべての検査結果テーブル名をタプルで返す
18 def get_pt2obs(dbcon, pid):
19     with dbcon.cursor() as cursor:
20         cursor.execute("select * from patient_observation_rel where patient_id =
- (%s)" , [pid])
21         rs = cursor.fetchall()
22         cursor.close()
23         return rs;
24
25 # 検査結果テーブルタプル（患者IDとペアになっている）のすべての検査レコードを一
- 定件数以上であれば出力する
26 def out_obsConditioned(hdstr, dStart,dEnd, dbcon, pt2obs,query3,
- threshold=100000,obsFetchLimit=10, limitOutputLines=10):
-
27 #   sqlStr = "select n_live_tup from pg_stat_user_tables where relname=(%s)"
28   for obsrec in pt2obs:   # 全レコード処理
29       patient_id = obsrec[0]
30       obsTableName = obsrec[1]           # obsrec[0]: patient_id,  obsrec[1]:
-       tablename

```

```

31     if obsTableName.endswith("81") or obsTableName.endswith("82") or
-     obsTableName.endswith("84") or obsTableName.endswith("85"):
32         out_obs(hdstr, dStart,dEnd, dbcon, obsTableName, patient_id,
-         query3,obsFetchLimit, limitOutputLines)
33     return
34
35 # 指定された1つの検査結果テーブルのすべての検査レコードを出力する
36 def out_obs(hstr, dStart,dEnd, dbcon, obsTableName, pid,
- query3,fetchLimit,limitOutputLines):
37     localCode = (obsTableName)[12:]
38     outlines =0
39     with dbcon.cursor('queyObs') as cursoObsTable:
40         cursoObsTable.execute(sql.SQL(query3).format(sql.Identifier(obsTableName))
-         , {'pid':pid,'startdis':dStart, 'enddis':dEnd })
41     while True:
42         rows = cursoObsTable.fetchmany(fetchLimit)
43         if len(rows) <= 0:
44             break;
45         for rec in rows:
46             print(hstr , end="")
47             print(localCode , end="")
48 #             sys.stderr.write ( localCode)
49
50             for i in range(len(rec)):
51                 print(", ", end="")
52                 print(rec[i], end="")
53 #                 sys.stderr.write ( ", ")
54 #                 sys.stderr.write ( str(rec[i]))
55             print()
56 #             sys.stderr.write ( "%n")
57         if len(rows) < fetchLimit:
58             break;
59         outlines +=len(rows)
60         if outlines >=limitOutputLines:
61             break;
62     cursoObsTable.close()
63     return
64

```

```

65 # プログラム本体メインルーチン
66
67 file_name1 = 'sqlICD.sql'
68 file_name2 = 'sqlDis.sql'
69 file_name3 = 'selectObs.sql'
70
71
72 with open(file_name1, 'r') as f1: # ICDサマリテーブル検索用
73     query1 = f1.read()
74 f1.close()
75
76 with open(file_name2, 'r') as f2: # 個別の病名テーブル（交換コード別テーブル）検索
77     query2 = f2.read()
78 f2.close()
79
80 with open(file_name3, 'r') as f3: # 個別の検査結果テーブル（検査ローカルコード別テ
81     query3 = f3.read()
82 f3.close()
83
84 argvs = sys.argv # コマンドライン引数を格納したリストの取得
85 argc = len(argvs) # 引数の個数
86
87
88 # psycopg2 の execute ではバインドパラメータを %(変数名)s で指定する必要がある
89
90 url = 'postgresql://ssmix2summary_101_2:dS2qbXef@192.41.180.250:5432/
91     - smmix2summary_101_2db'
92
93 limitOutputLines1 = 100000
94 limitOutputLines2 = 100000 # for each disCode
95 limitOutputLines3 = 100000 # for each obsCode
96 threshold = 100000
97 icdFetchLimit = 1000
98 disFetchLimit = 1000
99 obsFetchLimit = 1000
100 datetime19000101 = datetime(1900,1,1)

```

```

100 datetime21001231 = datetime(2100,12,31)
101 try:
102     conn = psycopg2.connect(url)
103     sys.stderr.write("successfully connected"+"¥n")
104     tablename = "icd10_summary"
105     if argc <2 :
106         condvalue = 'E21%' # 対象とするICD10パターン
107     else :
108         icd10Pattern=argvs[1]
109         condvalue = icd10Pattern
110     sys.stderr.write("ICD Pattern"+condvalue+"¥n")
111
112     with conn.cursor('queryEach') as cursorDisSummary:
113         cursorDisSummary.execute(sql.SQL(query1).format(sql.Identifier(tablename)),
114             -   {'condval':condvalue} )
115         lines = 0
116         while True:
117             rowsDisSummary = cursorDisSummary.fetchmany(icdFetchLimit)
118             sys.stderr.write("enter loop¥n")
119             sys.stderr.write("rowsDisSummary="+str(len(rowsDisSummary))+ "¥n")
120             if len(rowsDisSummary) <= 0:
121                 break;
122             for recDisSummary in rowsDisSummary: # 全レコード処理
123                 lines +=1
124                 sys.stderr.write (str(lines))
125                 for i in range(len(recDisSummary)): # 全カラム処理
126                     sys.stderr.write(",")
127                     sys.stderr.write(str(recDisSummary[i]))
128                 sys.stderr.write ( "¥n")
129             #
130             if recDisSummary[2] != "disease_none" :
131                 with conn.cursor('queryDisease') as cursorDisTable:
132                     cursorDisTable.execute(sql.SQL(query2).format(sql.Identifier(
133                         -   recDisSummary[2])) ) # recDisSummary[2]は病名交換用
134                     -   コード別の病名テーブル名
135                     dislines=0;
136                     while True:
137                         disrows = cursorDisTable.fetchmany(disFetchLimit)

```

```

135     if len(disrows) <= 0:
136         break;
137     sys.stderr.write("disrows="+str(len(disrows))+ "\n")
138     for disrec in disrows:    # 全レコード処理
139         dislines +=1
140         header =""
141     #     print(dislines, end="")
142     #     sys.stderr.write(str(dislines))
143         header +=str(dislines)
144     for j in range(len(disrec)):
145         #         # 全カラム処理
146         #         print(",", end="")
147         #         print(disrec[j], end="")
148         #         sys.stderr.write(",")
149         #         sys.stderr.write(str(disrec[j]))
150         #         header +="," + str(disrec[j])
151     #     print()
152     #     sys.stderr.write ( "\n")Fstr
153         header +=","
154
155         patientid = disrec[2]
156         # disrec[2]はpatient_id
157         # patientidから関連する検査結果テーブルをすべてチェック
158         disStart = disrec[8]
159         if disrec[9] is None or disrec[7] ==
160             datetime19000101:
161             disEnd = datetime21001231
162         else:
163             disEnd = disrec[9]
164         rows = get_pt2obs(conn, patientid)
165         out_obsConditioned(header,
166             disStart,disEnd,conn, rows,query3,
167             threshold,obsFetchLimit,limitOutputLines3)
168     if len(disrows) < disFetchLimit:
169         sys.stderr.write("break:
170             len(disrows):"+str(len(disrows))+ " <
171             disFetchLimit\n")

```

```

165         break;
166         if dislines >= limitOutputLines2 :
167             break;
168             cursorDisTable.close()
169         if len(rowsDisSummary) < icdFetchLimit:
170             sys.stderr.write("break: len(rowsDisSummary): "+
-             str(len(rowsDisSummary))+"<icdFetchLimit¥n")
171             break;
172         if lines >= limitOutputLines1 :
173             break;
174         cursorDisSummary.close()
175     except Exception as e:
176         print(e)
177     finally:
178         conn.close()
179
180

```

研究成果の刊行に関する一覧表

書籍

著者氏名	論文タイトル名	書籍全体の編集者名	書籍名	出版社名	出版地	出版年	ページ
該当なし							

雑誌

発表者氏名	論文タイトル名	発表誌名	巻号	ページ	出版年
K Yamada, M Ito, Y Fujimura, M Kimura, K Mura, N Nakashima, M Nakayama, K Ohe, T Orii, E Sueoka, T Suzuki, H Yokoi, Cryo Ishiguro, Y Uyama on behalf of MID-NET project group	The utilization and challenges of Japan's MID-NET® medical information database network in postmarketing drug safety assessments: A summary of pilot pharmacoeconomic studies	Pharmacoepidemiology and Drug Safety	28(5)	601-608	2019
Hayakawa M, Imai T, Kawazoe Y, Kozaki K, Ohe K.	Auto-Generated Physiological Chain Data for an Ontological Framework for Pharmacology and Mechanism of Action to Determine Suspected Drugs in Cases of Dysuria.	Drug Safety	42	1055-1069	2019
Kagawa R, Shinohara E, Imai T, Kawazoe Y, Ohe K.	Bias of Inaccurate Disease Mentions in Electronic Health Record-based Phenotyping	International journal of medical informatics	124	90-96	2019
Nakashima N, Nomura M, Ueki K, Kikuchi T, Hayashi M, Yamazaki K, Nakagami T, Ohara M, Gochi A, Matsumura Y, Kimura M, Ohe K, Kang D, Toya Y, Yamagata K, Yokote K, Ikeda S, Mitsutake N, Yamamoto R, Tanizawa Y.	Recommended configuration for personal health records by standardized data items for diabetes mellitus and associated chronic diseases	Initiative by six Japanese Associations. J Diabetes Investigation	10(3)	868-875	2019



発表者氏名	論文タイトル名	発表誌名	巻号	ページ	出版年
Seki T, Tamura T, Suzuki M	SOS-KANTO 2012 Study Group. Outcome prediction of out-of-hospital cardiac arrest with presumed cardiac aetiology using an advanced machine learning technique	Resuscitation	141	128-135	2019
大江 和彦	AIとビッグデータのための医療情報の標準化	医療機器学	89巻6号	545-551	2019
大江 和彦	ビッグデータと人工知能技術による診療支援システム	Dementia Japan	34巻1号	70-75	2020
早川 仁, 関 倫久, 河添 悦昌, 大江 和彦	パスワードデータベースを利用したグラフ畳み込み深層学習による悪性腫瘍の診断分類性能の検討	医療情報学連合大会論文集	第39回	352	2019
関 倫久, 河添 悦昌, 大江 和彦	SS-MIX2標準化ストレージを用いた入院後の死亡退院リスク予測モデルの開発	医療情報学連合大会論文集	第39回	249	2019

(医)

審査番号

11104-(2)

西暦 2019年01月25日

審査結果通知書  
実施許可通知書

倫理委員会の設置者

東京大学大学院医学系研究科・医学部長 殿

実施医療機関の長

東京大学医学部附属病院長 殿

倫理委員会委員長

東京大学大学院医学系研究科・医学部倫理委員会  
非介入等研究倫理委員会

神馬 征峰



審査依頼のあった件についての審査結果を下記のとおり通知いたします。

記

研究課題名	人工知能研究基盤としての多施設匿名化臨床データベース構築と活用研究（多施設共同観察研究）
審査結果	■承認する □条件付きで承認する □変更を勧告する □承認しない □該当しない □既承認事項の取り消し
審査事項 (審査資料)	<新規案件> □研究の新規実施 <継続案件> ■研究に関する変更 □その他（ ）
審査区分	□委員会審査（審査日：西暦 年 月 日） ■迅速審査（審査日：西暦2019年01月22日）
指摘事項および 理由・条件等	
備考	

研究責任者 大江 和彦 殿

依頼のあった研究に関する審査事項について上記のとおり決定しましたので通知いたします。

西暦 2019年01月24日

倫理委員会の設置者

東京大学大学院医学系研究科・医学部長

宮園 浩平（公印省略）

倫理委員会での審査結果が承認となりましたので、研究の実施を許可いたします。

西暦 2019年01月25日

実施医療機関の長

東京大学医学部附属病院長

齊藤 延人（公印省略）