卒業論文

有機合成における酵素番号予測のための特徴選択とクラスタリングを用いたケモインフォマティクス

Chemoinformatics Using Feature Selection and Clustering for Enzyme Commission Number Prediction in Organic Synthesis

富山県立大学 工学部 電子・情報工学科
1815070 武藤 克弥
指導教員 奥原 浩之 教授
提出年月: 2022年2月

目 次

図一覧

表一覧

記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

用語	記号
分子フィンガープリント	MFP
反応差分フィンガープリント	RFP
クラスタ i	C_i
C_i に属するデータ集合	$\mathbf{x_i}$
クラスタ間の距離	$d(C_1, C_2)$
クラスタ内の要素間の距離	$d(\mathbf{x_1}, \mathbf{x_2})$
個数	n
次元数	p
p 次元観測ベクトル	$\boldsymbol{x}_j$
i 番目のユニット	m_i
ユニット数	k
i 番目ユニットの重心	$\boldsymbol{r}_i$
i 番目ユニットの重みベクトル	$\boldsymbol{\xi}_i$
$\boldsymbol{x}_j$ と $\boldsymbol{\xi}_i$ のユークリッド距離	$\|\|\boldsymbol{x}_j - \boldsymbol{\xi}_i\|\|$
$\|\|\boldsymbol{x}_j - \boldsymbol{\xi}_i\|\|$ を最小化する $\boldsymbol{\xi}_i$	$\boldsymbol{\xi}_c$
$\boldsymbol{\xi}_c$ を持つ勝者ユニット	m_c
近傍関数	$h(t)$
ユニット m_c の近傍領域	N_c
学習率係数	$\alpha(t)$
N_c の散らばりに関する調整関数	$\sigma^2(t)$
反応物 i の特性値	RT_i
生成物 i の特性値	PD_i
記述子 j の特性値変化量	cv_j
i 番目の反応式の特徴ベクトル	$\boldsymbol{DF}_i$
記述子 u，v 間の相関係数	s_{uv}
記述子 u の特性値平均	$\bar{cv}_u$

第1章

はじめに

§ 1.1　本研究の背景

近年, ケモインフォマティクスと呼ばれる，化学に関するデータを情報技術を用いて分析する分野が発展してきている．化合物の特性や構造を分析したり，化合物の特徴を抽出し，機械学習における分類や化学反応の設計や予測といったことが行われている．

現在，新型コロナウイルスの世界的な流行をはじめとする多種の影響によって，新薬開発のニーズが高まっている．2026 年までの間に，ケモインフォマティクス業界は，年平均成長率 13 %で市場が成長すると予想されていることからも [?]，ケモインフォマティクスの需要は日々拡大している．

有機合成分野においては，ケモインフォマティクスや機械学習などの技術を取り入れて，化学反応の設計や予測をする研究が増加している．一方で，目的の生成物を得るために使用する反応触媒に，グリーンケミストリーの観点から，環境適応型の酵素を用いることが世界の風潮となってきている．酵素に代表される生体触媒は，人工的な化学触媒に比べて環境にやさしく，化学反応をより効率的に進めることから，化学触媒の代わりに生体触媒を用いて合成を行う取り組みが増加している．実際，目的の化合物を生成するために従来では 10 ステップの合成を行っていたものを，生体触媒を取り入れることで 3 ステップまで短縮したという研究事例もある [?]．これらのことから，目的物生成のために酵素反応を取り入れたうえで，反応設計を行うこと，あるいは，特定の反応に対して生体触媒として最適な酵素を予測することも重要な要素の一つとなってきている．

情報科学の観点からとらえると，酵素を触媒として取り入れる際，反応物 (基質) に対して特定の生体酵素を加えれば目的の生成物が得られる．つまり，基質と生成物が決まった場合，それに対して最適な酵素を予測するというのは容易に見えるかもしれない．ところが，実際には基質特異性と呼ばれる，酵素が基質に対して高い反応性を示すかどうかという酵素の特性によって，問題が複雑になる．有機合成化学を研究していて，酵素に関する知識を持ち合わせていたり，経験が豊富であれば，どの酵素が使えそうかある程度予測ができるかもしれない．しかし，先ほど述べた基質特異性に加えて，酵素のタンパク質配列を参照したりと，遺伝子分野にかかわる部分もあり，有機合成の知識だけでは解決が難しい場合がある．

§ 1.2　本研究の目的

生体触媒（Biocatalyst）を用いた有機合成化学において，目的とする生成物を効率よく得るために，酵素のデータベースを参照したり，酵素の研究を行っている専門家と協力するなどして，最適な酵素候補の目途をつけるという手法が取られる場合がある．実際は，酵素にも同様の性質を持っていたり，複数の企業製品が存在していたりと，触媒候補が複数存在する場合があるため，スクリーニングなどの実験の試行錯誤を繰り返しながら，最終的に 1 つに絞られていく．ここで，酵素の候補を探索したり，新たな酵素を設計する際に，有機合成化学の研究者自身で，酵素候補を探索することができれば，次の実験のステップまでスムーズに進めることができると考えられる．つまり，目的生成物を得るために，最適な酵素を迅速に予測・設計してくれるようなツールが存在すればよい．

本研究では，反応式を与えた際，その反応を触媒するのに必要な酵素を予測するシステムを考える．前述のとおり，1 つの酵素に絞り込むためには，様々な条件が絡む実験を必要とするため，おおまかな予測という形になる．しかし，有機合成化学の知識内で手順を進めていくことが可能となるため，十分に有効性があると考えられる．

酵素は酵素番号 (Enzyme Commission numbers: EC 番号) とよばれる，4 組の数字の組み合わせからなる番号が割り振られており，どの反応を触媒し，どの結合・基質に反応するかによって分類されている [?] [?]．与えられた反応に対して，酵素 (EC 番号) を予測できれば，その EC 番号の酵素から何を選択するかという次のステップに進むことができる．

EC 番号の情報の中には，反応物から生成物への，その酵素を使った代表的な反応が記載されている．そこで，本研究では，酵素を予測するターゲットとなる反応式内の反応物から生成物，また，EC 番号の代表的な反応式内の反応物から生成物，それぞれの物理・化学的特性値の変化を比較し，類似性が最も高い反応の酵素番号を提示して，最適な酵素を予測する．

主な流れとして，化学・酵素データベースから酵素の EC 番号および，代表的な反応式の情報を取得し，EC 番号と反応式の対応表を作成する．次に各反応式を，反応物と生成物に分解する．ターゲットの反応式も同様に分解し，各化合物の構造をコンピュータ上で扱うための表現に変換する．その後，複数の化合物の物理・化学特性値を計算し，各反応式において反応物から生成物への特性値の変化量を求める．この複数の特性値変化量を要素にもつ多次元ベクトルを，反応式の特徴ベクトルとして表現する．最終的に特徴ベクトルの次元削減を行い，クラスタリングによって反応式の特徴ベクトルを 2 次元平面上に出力する．得られた結果から，ターゲットの反応式に対して，最も近い場所に位置する，反応式の EC 番号に登録されている酵素を，用いるべき最適な酵素として予測する．

§ 1.3　本論文の概要

本論文は次のように構成される．

第 1 章 本研究の背景と目的について説明した．背景では，ケモインフォマティクスの概要，有機合成において，生体触媒を用いることのメリットとその課題について述べた．目的では，目的の生成物を得る際に用いる，最適な酵素を予測するための，EC 番号を予測するシステムの概要について述べた．

第 2 章 有機合成，ケモインフォマティクス，および酵素の概要を述べる．また，本研究で用いるデータベースについて述べる．

第 3 章 化学データベースからの情報抽出，ケモインフォマティクスでにおける化合物の構造表現法，EC 番号予測の概要を述べる．また，クラスタリング手法について述べる．

第 4 章 提案手法についての説明，および手順について説明する．

第 5 章 提案手法による数値実験の概要，実験結果と考察を述べる．

第 6 章 まとめと今後の課題について述べる．

第2章

複製・競合を考慮した動径基底関数ネットワーク

§ 2.1　競合動径基底関数ネットワーク

Dale 則

生体の脳の優れた特長は, 学習能力と並列分散処理能力である. これを工学的に応用するため, あるいは生体の情報処理を解明するため, 神経回路網をモデル化したニューラルネットワーク (Neural Network：NN) が研究されている.NN は大きく分けて, 素子であるニューロン, それらを結合するシナプス, そして動作規則により構成される. これらの相違により多様なモデルが考えられている. なかでも, 記憶にもっとも関係した情報処理は, シナプスにおいて行われているとされる. 記憶には種々のものが考えられるが, 本速報では短期記憶と長期記憶に着目し, 短期記憶はニューロンの発火頻度, 長期記憶は細胞膜の特性の変化により生じるものとする. シナプスの可塑性を記述する方程式は, これらの要因を含んだものとなっていなければならない. また, 実際の生体では, シナプス結合の性質が興奮性, 抑制性であるのかは送り出すニューロンにより決まる (Dale 則という). さらに, 微小な領域では成長や活動に必要な物質は競合によりシナプスに摂取される. これらの事実もシナプス可塑性のモデル化において重要な要因であると考えられる.

そこで, 本速報では発火頻度や膜の特性変化を生じる物質の時間変化と, 生理学的拘束条件である Dale 則や微小な領域での競合を考慮したシナプス可塑性方程式を導出する. さらに, 提案するシナプス可塑性方程式を解析することで, シナプスでは分岐を利用した情報処理が可能であることを示す.

脳のもつ柔軟性, 記憶や学習は, シナプス可塑性によるものである. シナプス可塑性は次の Hebb 則が提案され認められている.

$$\frac{dw_i^j}{dt} = A_i A_j \tag{2.1}$$

ここで,A_i は第 i ニューロンの活動度であり,w_i^j は第 j ニューロンから第 i ニューロンへのシナプス間感度を表わす. この方程式に従って, シナプス間感度が変化し続けると発散してしまう. 発散を防ぐために設けられる仮定がそのまま各モデルの特徴となっている 1). 発散をしないように修正されたシナプス方程式を用いて, 視覚第 1 次野における眼優位性コラムの形成を説明するモデル 2) や, トポグラフマッピングを説明するモデル 3) がある.

記憶は短期記憶 (Short Term Memory：STM) と長期記憶 (Long Term Memory：LTM) に大きく分けることができる 4).STM は電気刺激がシナプスの結合回路で保持され実現されるとするニューロン回路説が有力である.LTM はシナプスの膜の特性が変化することで

獲得されると考えられている. また,LTM が獲得されるためには STM が生じなければならず, これら二つの記憶は互いに影響を及ぼしていることもわかる.STM から LTM を引き起こすメカニズムとしてタンパク質リン酸化によるものが考えられている. これは高頻度刺激でシナプス間隔に放出された第 1 次メッセンジャが第 2 次メッセンジャを増加させる. これがタンパク質リン酸化をおこすというものである. 第 2 次メッセンジャとして cAMP や Ca2+ が知られている. 一般に LTM にはタンパク質の合成が必要であるとされる 4). このように, ニューロンの成長や活動維持のために必要な物質が存在する. ここでは, 第 1 次, 第 2 次メッセンジャをまとめて神経成長因子 (Nerve Growth Factor： NGF) と呼ぶこととする. シナプスの成長や活動に必要な物質 (NGF) は微小な領域では競合され消費される. このような競合による結合の消滅は, 運動ニューロンと筋繊維の問においても観察されモデル化されている 5). また, 一つのニューロンは生化学的に単一な性質であり, シナプス間感度が興奮性であるか抑制性であるかは, 送り出すニューロンによって決まっている. これを Dale 則という 6). これらの制約は, シナプス可塑性のモデル化において重要な要因となる.

神経細胞 (ニューロン) は脳を構成する最小単位である. ニューロンは細胞体と樹状突起, 軸索からなる. 軸索終末はシナプスと呼ばれ, 他のニューロンの細胞体にシナプス間隔を通して化学伝達物質を放出することにより情報を伝達する.NGF はシナプス間隔の微小な領域 B_{ik} において競合するものとする. ここで, 添字は第 i ニューロンの第 k 番目の微小領域を示す. 第 j ニューロンの微小領域 B_{jk} におけるシナプス前終末発火頻度を ξ_{ik}^{j} とし, これが作用する第 i ニューロンの細胞膜におけるシナプス後発火頻度を η_{ik} とする. シナプス間感度を w_{ik}^{j} とする. シナプス間感度 w_{ik}^{j} は Dale 則により, 第 j ニューロンが興奮性であるなら正, 抑制性であるなら負の値をとる. シナプス間感度の大きさの時間変化はシナプスの興奮性, 抑制性に依存せず,STM に関する発火頻度の項と LTM に関する NGF の項をあわせもつ以下の方程式に従うものとする.

$$\frac{dw_{ik}^{j}}{dt} = \alpha_{ik}^{j} w_{ik}^{j} + g_{ik}^{j} w_{ik}^{j} + f_{ik}^{j} \tag{2.2}$$

ここで, g_{ik}^{j} は微小領域 B_{ik} に供給される NGF のうち, その領域に付着している第 j ニューロンのシナプスが入手できる量である.f_{ik}^{j} は NGF と環境因子に依存するゆらぎである.α_{ik}^{j} は内的自然増加率であり，Hebb 則を表す．内的自然増加率 α_{ik}^{j} は以下のように定義される.

$$\alpha_{ik}^{j} = \int_{x \in B_{ik}} \eta_{ik}(x) \xi_{ik}^{j}(x) dx \tag{2.3}$$

ここで，NGF の量 g_{ik}^{j} は次の方程式に従う.

$$\begin{aligned} \frac{dg_{ik}^{j}}{dt} &= \epsilon_{ik}^{j}(G_{ik} - g_{ik}^{j}) - (\beta_{ik}^{j}\mu^{j} w_{ik}^{j} + \sum_{h \neq j} \beta_{ik}^{h}\mu^{h} w_{ik}^{h}) \\ &= \epsilon_{ik}^{j}(G_{ik} - g_{ik}^{j}) - \sum_{h} \beta_{ik}^{h}\mu^{h} w_{ik}^{h} \end{aligned} \tag{2.4}$$

ここで,G_{ik} は B_{ik} への NGF の供給速度であり, 膜の特性により, 決定される変数である.ϵ_{ik}^{j},β_{ik}^{h} は正の定数である. また，μ^{j} は第 j ニューロンが興奮性であるときに 1 となり, 抑制性であるときに-1 となる Dale 則を考慮するための識別子である.NGF の量の時間変化

もシナプスの興奮性, 抑制性によらず, そのシナプス間感度の大きさに依存する. また, 領域へ付着するシナプスが入手し得る NGF の量の時間変化に対し,NGF の供給速度の時間変化が無視できるとして G_{ik} を定数とみなす. シナプス間感度の時間変化は発火頻度に依存するため, シナプス間感度の時間変化に対し,NGF の量の時間変化は無視できるものとする. そこで, 隷従化原理 7) を適用することにより, 第 j ニューロンと第 h ニューロンが同時に NGF を消費することによる競合の効果 γ_{ik}^{jh} を考慮すると以下の Dale 則を考慮したシナプス可塑性の方程式が導かれる.

$$\begin{aligned}\frac{dw_{ik}^{j}}{dt} &= (G_{ik} + \alpha_{ik}^{j} - \frac{1}{\epsilon_{ik}^{j}} \sum_{h} \beta_{ik}^{h} \mu^{h} w_{ik}^{h}) w_{ik}^{j} + f_{ik}^{j} \\ &= (G_{ik} + \alpha_{ik}^{j} - \sum_{h} \gamma_{ik}^{jh} \mu^{h} w_{ik}^{h}) w_{ik}^{j} + f_{ik}^{j} \end{aligned} \tag{2.5}$$

ここでは，競合係数 γ_{ik}^{jh} を

$$\gamma_{ik}^{jh} = \frac{\beta_{ik}^{h}}{\epsilon_{ik}^{j}} = \int_{x \in B_{ik}} \xi_{ik}^{j}(x) \xi_{ik}^{h}(x) dx \tag{2.6}$$

で定義する.

Mikhailov の p167〜170 を展開する

ここで，Mikhailov の Wavas of Reproduction の形式に合わせるために (2.5) 式を以下のように変形する．ここでは揺らぎ f_{ik}^{j} を考えないものとする ($f_{ik}^{j} = 0$).

$$\begin{aligned}\frac{dw_{ik}^{j}}{dt} &= (G_{ik} + \alpha_{ik}^{j} - \frac{1}{\epsilon_{ik}^{j}} \sum_{h} \beta_{ik}^{h} \mu^{h} w_{ik}^{h}) w_{ik}^{j} \\ &= (G_{ik} + \alpha_{ik}^{j} - \frac{\beta^{j}}{\epsilon_{ik}^{j}} \mu^{j}(w_{ik}^{j}) - \sum_{h \neq j} \frac{\beta^{h}}{\epsilon_{ik}^{j}} \mu^{h} w_{ik}^{h}) w_{ik}^{j} \\ &= (\lambda_{ik}^{j} - \gamma_{ik}^{jj} \mu^{j} w_{ik}^{j} - \sum_{h \neq j} \gamma_{ik}^{jh} \mu^{h} w_{ik}^{h}) w_{ik}^{j} \end{aligned} \tag{2.7}$$

ここで，$\lambda_{ik}^{j} = G_{ik} + \alpha_{ik}^{j}$，$\gamma_{ik}^{jh} = \frac{\beta^{h}}{\epsilon_{ik}^{j}}$ である．さらに，両辺に μ^{j} を乗算すると (2.6) 式は以下のようになる.

$$\frac{d\mu^{j} w_{ik}^{j}}{dt} = (\lambda_{ik}^{j} - \gamma_{ik}^{jj} \mu^{j} w_{ik}^{j} - \sum_{h \neq j} \gamma_{ik}^{jh} \mu^{h} w_{ik}^{h}) \mu^{j} w_{ik}^{j} \tag{2.8}$$

(2.7) 式において $\mu^{j} w_{ik}^{j} = u_{ik}^{j}$ とおき，とある i と k についてのみ考えると

$$\frac{du^{j}}{dt} = (\lambda^{j} - \gamma^{jj} u^{j} - \sum_{h \neq j} \gamma^{jh} u^{h}) u^{j}$$

$$= \lambda^j (1 - \frac{\gamma^{jj}}{\lambda^j} u^j - \sum_{h \neq j} \frac{\gamma^{jh}}{\lambda^j} u^h) u^j \tag{2.9}$$

となる．また，$v^h = \frac{\gamma^{jh}}{\lambda^h} u^h$ と変数変換を行うと，次のように変形できる．

$$\frac{dv^j}{dt} = \lambda^j (1 - v^j - \sum_{h \neq j} \frac{\lambda^h}{\lambda^j} v^h) v^j \tag{2.10}$$

となる．Mikhailov によると，揺らぎがなく ($f^j_{ik} = 0$)，$\frac{\lambda^h}{\lambda^j} > 1$ のとき，競合が生じ任意のシナプス結合荷重がほかのすべてのシナプス結合荷重を抑制して生き残る．また，$0 < \frac{\lambda^h}{\lambda^j} < 1$ のときも競合は生じるが抑制は他のシナプス結合荷重を抑制するほど強くなく，複数のシナプス結合荷重が生き残る．

適者生存

本論文では，シナプス結合荷重間の競合を考慮したシナプス可塑性方程式を導出し，これを学習則として適用した動径基底関数ネットワークを提案する．動径基底関数ネットワークは階層型ニューラルネットワークに比較してニューロンごとの局所的な学習が可能であるなどの優れた点をもつため，関数近似問題やパターン識別に適用され成果を上げている．しかし，動径基底関数ネットワークでは未知の非線形関数を近似するためにあらかじめ必要なニューロン数が不明であるために冗長なニューロンを必要とする．一般に，ニューロンの増加は学習の遅延化や過学習の問題を生じることが知られている．そこで，これらの問題を解決するために，適者生存型学習則に基づいたシナプス可塑性方程式を適用した競合動径基底関数ネットワークを提案する．競合動径基底関数ネットワークではシナプス結合荷重間に競合を生じさせ，学習に必要なニューロンのみが自然に生き残り，学習の効率化を図ることができる．シミュレーションでは，競合動径基底関数ネットワークが通常の動径基底関数ネットワークに比較して高速に学習できることを示す．また，未知の確率密度関数に従い分布する標本点が与えられたとき，冗長なニューロンをもつ競合動径基底関数ネットワークが学習により不必要なニューロンを排除して，最適なニューロン数で元の確率密度関数を推定できることを示す．

ニューラルネットワークは素子であるニューロンとそれらをつなぐシナプス結合荷重から構成される．ニューラルネットワークへの入力は入力ニューロンで受け取られ，ニューロンとシナプス結合荷重による変化を受けて伝達され，出力ニューロンから出力される．このとき，ニューラルネットワークの学習とは入力に対応する望ましい出力を得るために，ニューロンの入出力特性とシナプス結合荷重を変化させることであると言える．ここで，任意の非線形関数は十分な数の中間ニューロンを用いた 3 層ニューラルネットワークで，有界閉集合上で一様に近似できるという関数近似定理 [1] が示されている．

ところで，入力からそれに対応する望ましい出力を得るために必要とされる中間ニューロンの数は未知である．そのため，学習後に中間ニューロン数を AIC を利用して決定する手法 [2] や，学習中に中間ニューロンを逐次追加していく手法 [3]，逐次削除していく手法 [4] などが提案されている．本論文では，ニューラルネットワークの初期状態に冗長なニューロンとシナプス結合荷重がぞんざいする場合において，学習中にシナプス結合荷重間に競合を生じさせて冗長なニューロンとシナプス結合荷重を削除する手法を提案する．

一般に，動径基底関数ネットワーク (Radial Basis Function Networks：以下，RBFN)[5] においても，冗長なニューロンを考慮することによるシナプス結合荷重の増加は，学習の遅延や過学習の問題を生じる．これらの問題を解決する方法として，望ましい出力と相関が高い入力を伝達しているシナプス結合荷重により伝達される入力が変化を受ける入力ニューロンの入出力特性のみを調節することが考えられる．そこで，本論文ではシナプス結合荷重間に入出力の相関に応じた競合が生じるようなシナプス可塑性方程式を導出する．そして，得られたシナプス可塑性方程式を適者生存型学習則として適用した RBFN を競合動径基底関数ネットワーク (Competitive Radial Basis Function Networks：以下，CRBFN) として提案する．更に，望ましい出力が動径基底関数の定数倍の足し合せで実現できる特別の場合においては，ターミナルアトラクタ [6] を利用することで望ましい時間で収束するシナプス可塑性方程式が導出できることを示す．シミュレーションでは，RBFN に比較して CRBFN が優れていることを示す．

本論文の構成は次のとおりである．2. では RBFN の概要について述べる．3. ではシナプス結合荷重間の競合を考慮したシナプス可塑性方程式の導出を行う．4. では導出されたシナプス可塑性方程式を適者生存型学習則として適用した CRBFN を提案する．5. では提案した CRBFN の性能シミュレーションにより確認する．6. ではまとめと今後の課題について述べる．

RBFN は非線形関数 $\eta(\mathbf{x})$ を動径基底関数 $\xi^i(\mathbf{x})$ の足し合せで近似するニューラルネットワークである動径基底関数としては規格化されたガウス型活性関数などが用いられる．N 個の入力ニューロンと 1 個の出力ニューロンからなる RBFN は図 1 のような構造をもつ．d次元の入力ベクトル $\mathbf{x}_j \in R^d (j = 1, 2, ..., M)$ はすべての入力ニューロンに入力される．第 i ニューロン $(i = 1, 2, ..., N)$ はパラメータ $\phi^i \equiv [\mathbf{m}^i, \Sigma^i]$ をもつ．ここで，$\mathbf{m}^i \equiv [m_1^i, m_2^i, ..., m_d^i]^{\mathrm{T}}$ であり，$\Sigma^i \in R^{d\times d}$ である．また，Σ^i は正定値対称行列である．第 i ニューロンは入力ベクトル $\mathbf{x}_j$ に対して

$$\xi^i(\mathbf{x}_j) = exp\{-\frac{1}{2}(\mathbf{x}_j - \mathbf{m}_i)^{\mathrm{T}}\Sigma_i^{-2}(\mathbf{x}_j - \mathbf{m}_i)\} \tag{2.11}$$

を出力する．ここで，添字の T はベクトルの転置を示す．出力値 $\xi^i(\mathbf{x}_j)$ はシナプス結合荷重 w^i を通して出力ニューロンへ伝達され，出力ニューロンでこれらは足し合され

$$s(\mathbf{x}_j) = \sum_i^N w^i \xi^i(\mathbf{x}_j) \tag{2.12}$$

が出力される．RBFN による関数近似は 2 乗誤差評価関数

$$E(\mathbf{w}) = \frac{1}{2}\sum_j^M \{\eta(\mathbf{x}_j) - s(\mathbf{x}_j)\}^2 \tag{2.13}$$

を減少させることにより実現される．ここで，$\mathbf{w} \equiv [w^1, w^2, ..., w^N]^{\mathrm{T}} \in R^N$ である．つまり，RBFN が学習により獲得しなければならないのは，第 i ニューロンのシナプス結合荷重 w^i，パラメータ $\mathbf{m}^i$ ならびにパラメータ Σ^i である．学習アルゴリズムに Delta ルール [7] を適用することで

$$\frac{dw^i}{dt} = -\Delta\frac{\partial E(\mathbf{w})}{\partial w^i}$$
$$= \Delta\sum_j^M\{\eta(x_j) - s(x_j)\}\xi^i(x_j), \tag{2.14}$$
$$\frac{dm_1^i}{dt} = -\Delta\frac{\partial E(\mathbf{w})}{\partial m_1^i}$$
$$= \Delta\sum_j^M\{\eta(x_j) - s(x_j)\}w^i\xi^i(x_j)\frac{(x_j - m_1^i)}{{\sigma^i}^2}, \tag{2.15}$$
$$\frac{d\sigma^i}{dt} = -\Delta\frac{\partial E(\mathbf{w})}{\partial \sigma^i}$$
$$= \Delta\sum_j^M\{\eta(x_j) - s(x_j)\}w^i\xi^i(x_j)\frac{(x_j - m_1^i)^2}{{\sigma^i}^3} \tag{2.16}$$

が得られる．但し，簡単のため $d = 1$ とし $\Sigma^i = \sigma^i$ とした．Δ は適当な正の定数である．ここで，RBFN についても 3 層ニューラルネットワークと同様な関数近似定理 [8] が示されていることを付記しておく．

ところで，未知の非線形関数を近似するために必要な動径基底関数の個数をあらかじめ知ることはできない．そのため，RBFN では初期状態においていくつかの冗長なニューロンを備えている．一般に入力ニューロン数 N は入力ベクトル数 M に比較して少なく設定される $(N \leq M)$．また，入力ニューロンのパラメータ $\mathbf{m}^i$ の決定法には入力ベクトルの部分集合から与える手法と，入力ベクトルには無関係として与える手法がある．パラメータ $\mathbf{m}^i$ を入力ベクトルとは無関係とし非線形関数を近似する手法には，式 (2.8) の 2 乗誤差評価のこう配に基づき Delta ルールを適用する最急降下手法と，$k - mean$ 法により入力ベクトル群をクラスタリングして入力ニューロンのパラメータ $\mathbf{m}^i$ を定めると共にパラメータ Σ^i を求め，最小 2 乗法によりシナプス結合荷重を決定するハイブリッド手法 [9] などがある．パラメータ $\mathbf{m}^i$ を入力ベクトルの部分集合から与える手法 [10] では，2 乗誤差評価の降下に大きく寄与するシナプス結合荷重から必要な数だけを取り出すことで RBFN の低次元化を行っている．

しかし，これらの RBFN の低次元化において必要となるニューロン数を決定する基準が確立されていない．そこで，本論文では RBFN の低次元化に適用できる適者生存型学習則を提案する．この手法はシナプス結合荷重間に競合を生じさせることにより，学習しながら自然な低次元化を行うことが可能となる．

ニューロンは他のニューロンにシナプス間隔を通して神経伝達物質を放出することにより情報を伝達する．第 i ニューロンの第 k 微小領域に付着する第 j ニューロンからのシナプス結合荷重 w_{ik}^j のシナプス可塑性を記述する方程式を

$$\frac{dw_{ik}^j}{dt} = \alpha_{ik}^j w_{ik}^j + g_{ik}^j w_{ik}^j + f_{ik}^j \tag{2.17}$$

で与える．シナプス結合荷重 w_{ik}^j は Dale 則により，第 j ニューロンが興奮性であるなら正，抑制性なら負の値をとる．α_{ik}^j は内的自然増加率，g_{ik}^j はシナプス結合荷重 w_{ik}^j におけ

る神経成長因子 (Nerve Growth Factor : 以下，NGF) の摂取量である．f_{ik}^{j} は揺らぎを表す．内的自然増加率 α_{ik}^{j} を

$$\alpha_{ik}^{j} = \int_{x \in B_{ik}} \eta_{ik}(x)\xi_{ik}^{j}(x)dx \tag{2.18}$$

で定義する．ここで，B_{ik} は第 i ニューロンの第 k 微小領域を表す．ξ_{ik}^{j} は微小領域 B_{ik} における第 j ニューロンのシナプス前終末発火頻度を表し，これが作用する第 i ニューロンのニューロン膜におけるシナプス後発火頻度を η_{ik} で表す．内的自然増加率 α_{ik}^{j} は Hebb 則を表すことがわかる．また，NGF の摂取量 g_{ik}^{j} の時間変化が

$$\frac{dg_{ik}^{j}}{dt} = \epsilon_{ik}^{j}(G_{ik} - g_{ik}^{j}) - \sum_{h} \beta_{ik}^{h}\mu^{h}w_{ik}^{h} \tag{2.19}$$

に従うものとする．ここで，パラメータ ϵ_{ik}^{j} は正の定数である．G_{ik} は細胞体から微小領域 B_{ik} への NGF の供給速度であり，ニューロンの特性により決定される定数である．Dale 則を考慮した識別子 μ^{j} は第 j ニューロンが興奮性であるなら 1，抑制性であるなら-1 となる．N_{ik} は微小領域 B_{ik} に付着するシナプス前終末の数である．競争係数 $\frac{\beta_{ik}^{h}}{\epsilon_{ik}^{j}}$ は，第 j ニューロンと第 h ニューロンが同時に NGF を消費することによる競合の効果であり，

$$\gamma_{ik}^{jh} = \frac{\beta_{ik}^{h}}{\epsilon_{ik}^{j}} = \int_{x \in B_{ik}} \xi_{ik}^{j}(x)\xi_{ik}^{h}(x)dx \tag{2.20}$$

で定義する．ここで，シナプス結合荷重 w_{ik}^{j} の時間変化に対し，NGF の摂取量 g_{ik}^{j} の時間変化が無視できるものとすると

$$g_{ik}^{j} = G_{ik} - \sum_{h} \gamma_{ik}^{jh}\mu^{h}w_{ik}^{h} \tag{2.21}$$

を得る．その結果，Dale 則を考慮したシナプス可塑性方程式として

$$\frac{dw_{ik}^{j}}{dt} = (\alpha_{ik}^{j} + G_{ik} - \sum_{h} \gamma_{ik}^{jh}\mu^{h}w_{ik}^{h})w_{ik}^{j} + f_{ik}^{j} \tag{2.22}$$

が導かれる．

簡単のため，以下では微小領域 B_{ik} に着目し，NGF の摂取量が一定 ($G_{ik} = 0$) で揺らぎのない場合 $(f_{ik}^{j}) = 0$ を考える．このとき，正定関数 $V(\mathbf{w}_{ik})$ として

$$V(\mathbf{w}_{ik}) = \frac{1}{2}\int_{x \in B_{ik}} \{\eta_{ik}(x) - s_{ik}(x)\}^{2}dx \tag{2.23}$$

を定義する．ここで，$\mathbf{w}_{ik} \equiv [w_{ik}^{1}, w_{ik}^{2}, ..., w_{ik}^{N_{ik}}]^{\mathrm{T}} \in R^{N_{ik}}$ であり，

$$s_{ik}(x) = \sum_{j=1}^{N_{ik}} \mu^{j}w_{ik}^{j}\xi_{ik}^{j}(x) \tag{2.24}$$

である．この式の右辺は微小領域 B_{ik} に付着しているすべてのシナプス前終末のシナプス前発火頻度 $\xi_{ik}^{j}(x)$ と，Dale 則を考慮したシナプス結合荷重 $\mu^{j}w_{ik}^{j}$ との積の総和であるの

で，$s_{ik}(x)$ を神経伝達物質放出量と呼ぶこととする．正定関数 $V(\mathbf{w}_{ik})$ はシナプス後発火頻度 $\eta_{ik}(x)$ と神経伝達物質放出量 $s_{ik}(x)$ の差を表す指標である．シナプス後発火頻度 $\eta_{ik}(x)$ が時間に依存しないと仮定すると，その時間変化が

$$\begin{aligned}
\frac{dV(\mathbf{w}_{ik})}{dt} &= \sum_{j=1}^{N_{ik}} \frac{\partial V(\mathbf{w}_{ik})}{\partial w_{ik}^j} \frac{dw_{ik}^j}{dt} \\
&= -\sum_{j=1}^{N_{ik}} \mu^j [\int_{x\in B_{ik}} \eta_{ik}(x)\xi_{ik}^j(x)dx - \sum_{h=1}^{N_{ik}} \int_{x\in B_{ik}} \xi_{ik}^j(x)\xi_{ik}^h(x)dx\mu^h w_{ik}^h] \frac{dw_{ik}^j}{dt} \\
&= -\sum_{j=1}^{N_{ik}} \mu^j w_{ik}^j (\alpha_{ik}^j - \sum_{h=1}^{N_{ik}} \gamma_{ik}^{jh}\mu^h w_{ik}^h)^2 \\
&\leq 0
\end{aligned} \tag{2.25}$$

となるため，正定関数 $V(\mathbf{w}_{ik})$ が Lyapunov 関数となることがわかる．

これらから，シナプス後発火頻度 $\eta_{ik}(x)$ を入力 x に対する望ましい出力，シナプス前発火頻度 ξ_{ik}^j を動径基底関数であるとみなすことで，シナプス結合荷重 w_{ik}^j は競合を行いながら RBFN と同様に望ましい出力と動径基底関数の 2 乗誤差評価関数を減少させることが示される．本論文では，式 (7) のシナプス可塑性方程式を適者生存型学習則ということとする．

§ 2.2　基底関数の複製とターミナルアトラクタ

ニューラルネットワーク（Neural Networks：以下，N.N.）の一つに動径基底関数ネットワーク（Radial Basis Function Network：以下，RBFN）[1] がある．RBFN はニューロンごとの局所的な学習が可能であり，ほかの N.N. に比較して学習が高速であることが知られている．また，十分な数のニューロンを用いた RBFN は任意の非線形関数を有界閉集合上で一様に近似できることが関数近似定理 [2] で示されている．一般に，未知の非線形関数を近似するために必要なニューロンの数があらかじめには不明である．もし，N.N. に冗長なニューロンが多数存在する場合は，学習の遅延や過学習の問題を生じることとなる．そこで，冗長なニューロンを削除する手法が提案されている [3], [4]．これに対し，N.N. に関数近似に必要な数のニューロンが存在しない場合は，関数近似をすること自体が不可能となる．そこで，新たに必要なニューロンを追加する手法が提案されている [5]．これら従来の研究には，しきい値などを考え動径基底関数の削除と追加を行うものもある．ところが，このような手法では基準となるしきい値の決定自体が困難であることが予想できる．また，教師信号が動的に変化する環境では削除する手法と追加する手法を組み合わせて学習を行わなければならない．当然それぞれにとって良い手法を，ただそのまま組み合わせただけでは，動径基底関数の数が振動するなどして望ましい結果が得られるとは限らない．

我々は先に，冗長なニューロンを削除できるシナプス可塑性方程式を導出し，これを適者生存型学習則としてシナプス結合荷重の更新則に適用した競合動径基底関数ネットワーク（Competitive Radial Basis Function Network：以下，CRBFN）[6] を提案した．CRBFN の特長は望ましい出力と相関が高い入力を伝達しているシナプス結合荷重が生き残り，生

き残ったシナプス結合荷重に関係する入力ニューロンの入出力特性のみが調節されることである．そのため，CRBFN では競合により冗長なニューロンを消滅させることが可能であり，その結果，学習の高速化と過学習の回避が行われる．

しかしながら，CRBFN でも教師信号が変化するような環境の変化には対応しきれていなかった．その理由は CRBFN には新しい動径基底関数を追加する能力がないからである．ここでいう環境の変化とは入出力間の写像を与える関数そのものが変化する場合や，既に観測され学習に用いられていた入出力の組が不要となり取り除かれたり，新たに観測された入出力の組が学習に用いられたりするような変化などを想定している．そこで本研究では，まず新しい動径基底関数を追加する手法を提案する．この手法はシナプス可塑性方程式（Delta ルール，適者生存型学習則）に関する考察から得られるものであり，必要な動径基底関数を効率的に追加することができる．そして，我々が先に提案した CRBFN にこの手法を組み合わせたニューラルネットとして複製・競合動径基底関数ネットワーク（Reproductive CRBFN：以下，RC-RBFN）を提案する．この RC-RBFN は，環境の変化に適応する能力を備えたものとなっている．本論文では，最終的に得られる結果の有効性から適者生存型学習則に対する動径基底関数の複製アルゴリズムについて議論を展開しているが，同様な議論が従来の Delta ルールに対しても可能であることも付録で述べる．

本論文の構成は次のとおりである．2. では CRBFN の概要について述べる．3. ではシナプス可塑性方程式に関する考察から，動径基底関数のパラメータの従う確率密度関数の導出を行う．4. では導出された確率密度関数を利用した動径基底関数の複製アルゴリズムを提案する．5. では提案した RC-RBFN の性能をシミュレーションにより確認する．6. ではまとめと今後の課題について述べる．

RBFN は非線形関数 $\eta(x)$ を動径基底関数の足し合せで近似するニューラルネットワークである．動径基底関数としては規格化されたガウス型活性関数などが用いられる．M 個の入力ニューロンと 1 個の出力ニューロンからなる RBFN は図 1 のような構造をもつ．d 次元の第 i 入力ベクトル $x_i \in R^d, (i = 1, 2, ..., N)$ はすべての入力ニューロンに入力される．第 j 入力ニューロン $(j = 1, 2, ..., M)$ はパラメータ ϕ_j をもつ．パラメータ ϕ_j は平均ベクトルと共分散行列の集合 $\{\mathbf{m}_j, \Sigma_j\}$ であるものとする．ここで，$\mathbf{m}_j = [m_i^1, m_i^2, ..., m_i^d]^{\mathrm{T}}$ であり，Σ_j はその逆行列 Σ_j^{-1} の第 kl 要素に σ_j^{kl} をもつ d × d の行列である．また，Σ j は正定値対称行列である．第 j 入力ニューロンは入力ベクトル xi に対して

を出力する．ここで，添字の T はベクトルの転置を示す．以後，このような出力を行う入力ニューロンのことを動径基底関数ということとする．出力値 ξ (xi, φ j) はシナプス結合荷重 wj を通して出力ニューロンへ伝達され，出力ニューロンでこれらは足し合わされ

が出力される．ここで，w = [w1, w2, · · · ,wM]T ∈ RM であり，φ で集合φ 1, φ 2, · · · , φ M を表す．ニューラルネットワークによる関数近似は，非線形関数 η (x) をネットワークの出力 s(x,w, φ) で表すことである．そのため，RBFN による関数近似は累積 2 乗誤差関数 (3) の値を減少させることにより実現される．ここで，

(4) は 2 乗誤差関数である．つまり，RBFN が学習により獲得しなければならないのは，第 j 動径基底関数のシナプス結合荷重 wj，パラメータ mj 並びにパラメータ Σ j である．ここで，従来の RBFN と CRBFN の学習アルゴリズムの相違について述べる．一般の RBFN の学習アルゴリズムは式 (3) の累積 2 乗誤差関数に Delta ルール [7] を適用した (7) で与えられる．ただし， は適当な正の定数であり，mkj はパラメータ mj の第 k 要素である．ま

た，Δ wj ≡ dwj/dt，Δ mkj ≡ dmkj /dt，Δ σ kl j ≡ d σ kl j /dt である．ところで，未知の非線形関数を近似するために必要な動径基底関数の個数をあらかじめ知ることはできない．そのため一般に，RBFN では初期状態においていくつかの冗長な入力ニューロンを備えている．このことは，学習の遅延化や過学習を招く原因の一つであった．CRBFN では，パラメータ mj 並びにパラメータ Σ j の学習アルゴリズムは従来の RBFN と同じであり，式 (6)，(7) により与えられる．しかし，シナプス結合荷重 wj に対しては Dale 則を考慮したシナプス可塑性方程式である適者生存型学習則 (8) が適用される [8]．シナプス結合荷重 wj は Dale 則により，第 j ニューロンが興奮性であるなら正，抑制性であるなら負の値をとる．また，μ k は第 k ニューロンが興奮性（wk ¿ 0）であるなら 1，抑制性（wk ¡ 0）であるなら − 1 となる Dale 則を考慮した識別子である．ここで，α j (φ) は内的自然増加率であり (9) で定義される．γ jk(φ) は競争係数であり，第 j ニューロンと第 k ニューロンとの競合の効果を表し

(10) で定義される．また，CRBFN の出力は (11) で得られることとする．このとき，累積 2 乗誤差関数 E(w, φ) は式 (8) に対する Lyapunov 関数であることが (12) により示される．CRBFN の学習則は累積 2 乗誤差関数 E(w, φ) の値を減少させるために第 j 動径基底関数のシナプス結合荷重を式 (8) で更新する．学習中に となった第 j シナプス結合荷重は消滅したものとして，生き残っているシナプス結合荷重，並びにそれらにより伝達される入力が変化を受ける動径基底関数のパラメータについてのみ学習を続ける．この学習則はシナプス結合荷重の更新則に特徴があるものの，平均ベクトルと共分散行列の更新則は従来の最急降下法を用いている．そこで，本研究では更に CRBFN において平均ベクトルの更新則を改良することにより，動径基底関数を複製する競合動径基底関数ネットワークを新たに提案する．3. パラメータが従う確率密度関数の導出ここでは，CRBFN の平均ベクトル，共分散行列とシナプス結合荷重が学習終了時にとる同時確率密度 p(w, φ) を導出する．まず，シナプス結合荷重 wj を

(13) と変数変換する．yj の定義域は任意の実数である．このとき，式 (8) は (14) となる．ただし， は省略した．式 (14) は積分条件

(15) を満たすためポテンシャル (16) を考えることができ，変数 yj の時間変化はポテンシャル V

(17) で導くことができる．関係式 (13) からポテンシャル V は

(18) と書き直すことができる．この結果，(19) であることが示されるので，累積 2 乗誤差関数 E(w, φ) の最小化はポテンシャル V (w, φ) の最小化と等価であることがわかる．今，式 (14) に従う yj はポテンシャル V (y, φ) の最急降下方向に更新される．その結果，ひとたび極小解に収束すると，そこから逃れることができなくなる．そこで，極小解から脱出させるための手法として，yj

の更新則を yj(t + Δ t) = yj(t) − ∂ V (y, φ) ∂ yj Δ t +

Q Δ t nj (t) (20) のようにノイズを考慮し離散近似した見本過程で与えることが考えられる．ただし，nj (t) は独立な確率変数であり，平均 0，分散 1 の正規分布 N(0, 1) に従う．Q は任意の正の定数である．このとき，学習終了時に CRBFN の平均ベクトル，共分散行列とシナプス結合荷重が満たす同時確率密度 p β (w, φ) は p β (w, φ) = Z − 1 β exp − β V (w, φ) (21) で得ることができる [9]．ここで， β = 2/Q である．Z β は分配関数であり (22) で定義される．また，式 (21) はポテンシャル V (w, φ) と累積 2 乗誤差関数 E(w,

φ) の関係式 (19) より (23) と書き直すことができる．ここで，β = (2Q) − 1 である．また，Z β は分配関数である．シナプス可塑性方程式として Delta ルールを用いている従来の RBFN に関しても，同様にしてパラメータが従う確率密度関数が導出できることを付録で示す．以上のようにして，パラメータが従う確率密度関数が導出できたことにより，与えられた条件のもとで累積 2 乗誤差関数 E(w, φ) を最小とするパラメータの値が検出できることを示す．ここでは，教師信号 η (x) を η (x) = 3N(− 1.5, 1) + 2N(1, 0.5) (24) で与えることとする．N(m, Σ) は平均 m，分散 Σ の正規密度関数を表す．この教師信号を動径基底関数を一つ（シナプス結合荷重 w = 1，パラメータ Σ = 0.2）だけ用いて近似することを考える．この場合，近似しようとしている非線形関数 η (x) の複雑さに対し，必要とされる動径基底関数が十分に存在していないため，累積 2 乗誤差関数 E(w, φ) を 0 にすること自体が不可能である．しかし，この動径基底関数のパラメータ m が従う条件付き確率密度関数

Fig. 2 The conditional probability density function of the parameter m.

(25) は導出することができ，それは図 2 のようになる．この結果から，シナプス結合荷重 w = 1，パラメータ Σ = 0.2 をもつ動径基底関数が与えられた条件のもとで累積 2 乗誤差関数 E(w, φ) を最小とするためには，パラメータ m を条件付き確率 p β (m—w, Σ) を最大とする値に定めればよいことがわかる．また，もし同じ形質（パラメータ w = 1，Σ = 0.2）をもつ動径基底関数を一つ追加することができるなら，条件付き確率 p β (m—w, Σ) を極大とするパラメータ m へ配置することが最も累積 2 乗誤差関数 E(w, φ) を小さくできることもわかる．式 (25) を更に，シナプス結合荷重 w とパラメータ Σ で積分をとれば確率 p β (m) が算出できる．そこで，教師信号を復元するために確率 p β (m) を極大とするパラメータ m に動径基底関数を配置するような一撃アルゴリズムを考えることもできる．しかしながら，多次元の場合にはシナプス結合荷重 w とパラメータ Σ の積分が困難であることから，本研究では確率 p β (m) を用いずに，条件付き確率を用いて逐次的にパラメータ m を求めていく方法を考える．4. 動径基底関数の複製アルゴリズム 4. 1 自由エネルギーの導出一般に，式 (6) に従いパラメータ mkj を更新し続けると極小解にとらわれ，累積 2 乗誤差関数 E(w, φ) の値を 0 にすることができないことがある．または，近似しようとしている非線形関数 η (x) の複雑さに対し，必要とされる動径基底関数が十分に存在していないときには，2 乗誤差関数 E(w, φ) の値を 0 にすること自体が不可能である．ところで，確率的な要素や未知の教師信号などが存在しないものとするなら，すべての入力ベクトル xi ごとに動径基底関数を作成し，シナプス結合荷重が wi = η (xi) かつパラメータ Σ i → 0 であるときに，パラメータ mi が xi となることで近似的に 0 とできる場合がある．ここで，0 は零行列を表す．もちろん，多くの問題ではすべての入力ベクトルについて動径基底関数を用意しなくても，このようなことが可能であるものと思われる．そこで本研究では，累積 2 乗誤差関数 E(w, φ) の値がある正数より大きな値に収束し，学習が収束したと判断されるときに，新たに必要な動径基底関数を追加する手法を提案する．ここで提案する手法では，前章で導出した確率密度関数を利用しているため，学習が収束した時点で得られている動径基底関数の一部の形質（シナプス結合荷重 wj，パラメータ Σ j ）が新たに追加される動径基底関数がもつパラメータに引き継がれている．そのため，効率的に最も累積 2 乗誤差関数を小さくするパラメータ m に動径底関数を追加していくことができる．なおかつ，最悪の場合にはすべての入力ベクトル xi をパラメータ mi とする動径基底関数を作成することができる．そこで，この手法を CRBFN に組み入れたニューラルネッ

トワークを複製・競合動径基底関数ネットワークと呼ぶこととする．ところで，累積 2 乗誤差関数 E(w, φ) の最小化は各入力ベクトル xi ごとに 2 乗誤差関数 E(xi,w, φ) を最小化することに等価である．そこで，各入力ベクトル xi に依存した平均ベクトル mj[i] を考える．そして，学習収束の時点で得られている第 j 番目の動径基底関数に着目すると，入力ベクトル xi の条件付き確率密度関数は (26) と導出できる．ここで，パラメータ φ

j は着目した第 j 番目の動径基底関数のシナプス結合荷重 wj と共分散行列 Σ j の集合であり，パラメータ φ j は着目した第 j 番目の動径基底関数以外のシナプス結合荷重，共分散行列並びに平均ベクトルの集合である．以後は

記法の簡便のため，パラメータ φ

j とパラメータ φ

j は省略する．また, 分配関数は (27) で定義される．条件付き確率密度関数 p β (xi—mj[i]) は，確率の正規化と 2 乗誤差関数 E(xi,mj[i]) の条件付き期待値 (28) が一定となるという二つの制約のもとで，エントロピー (29) を最大にする確率密度関数として導出できる [10]．ここで，記号

β は p β (xi—mj[i]) を掛けて xi に関する和をとる演算を表すものとする．このとき，自由エネルギーを F β (mj) = − 1 β log Z β (mj) (30) で定義すれば，S β (mj) = − F β (mj) + β E(mj)

β (31) と表すことができる．式 (31) はエントロピー S β (mj) を最大化する条件付き確率密度関数 p β (xi—mj[i]) は，自由エネルギー F β (mj) を最小化するものであることを示している．このような自由エネルギーは，データのクラスタリングのための手法であるメルティング [11] においても同様に定義されている．メルティングとは，mj[i] = xi (i = 1, 2, · · · ,N) かつ β が∞ である初期状態から，徐々に β を 0 へ近づけていきながら，パラメータ mj[i] を自由エネルギー F β (mj) の最急降下方向に更新していくものである．その結果，パラメータ mj[i] は徐々に同じ値をとりはじめ，最終的に一つの値 mj[i] = mj (∀ i) に収束する．4. 2 複製する位置の決定法そこで，RC-RBFN ではパラメータ mkj の更新則を式 (6) の Δ mkj の代わりに (32) で与えることとする．ここで，(33) である．特にであり，初期の状態がである場合は Δ 0mkj = Δ mkj (34) であることが示される．この場合は，RC-RBFN のパラメータ mkj の更新則が従来の RBFN のパラメータ mkj の更新則そのものとなっていることがわかる．このとき，β = 0 で固定したままパラメータを Σ j → 0 にすると，Δ 0mkj= 0 とするパラメータ mj[i] は (35) を満たし，mj[i] = xi (∀ i) であることがわかる．つまり，教師入力信号がパラメータ mj の収束点として検出されることとなる．得られた結果を確認するために，式 (24) の教師信号から適当に 5 点選んだのが表 1 である．これを教師入力信号 xi (i = 1, 2, · · · , 5) とする．式 (32) において β を 0，パラメータを Σ を徐々に小さくしたときに，Δ 0m[i] = 0 となる点をプロットしたのが図 3 である．パラメータ Σ → 0 において m[i] = xi (i = 1, 2, · · · , 5) へ収束していることがわかる．ただし，この

Fig. 3 The parameter m satisfying Δ 0m = 0. 例では j = k = 1 であるため，添字の j と k は省略した．あるいは，逆にパラメータ Σ j を固定したまま β →∞ にすると，Δ ∞ mkj = 0 とするパラメータ mj[i] は (36) を満たし，xi (∀ i) を含む任意の値となることがわかる．これらの結果から，提案する RC-RBFN のパラメータ mkj の更新則 Δ β mkj では，Δ 0mkj で従来の RBFN のパラメータ mkj の更新則を実現し，更に，Σ j → 0 とすればすべ

ての入力ベクトル xi の第 k 要素 xki (∀ i) をパラメータ mkj の安定な収束点として検出できることがわかる．あるいは，Δ∞ mkj とすることで，すべての入力ベクトル xi の第 k 要素 xki (∀ i) を含む任意の値を安定な収束点とすることができる．ここで，提案手法とゆう度解析との関係について示す．まず，次のような自由エネルギー (37) を考える．ただし，分配関数は (38) で与えられる定数である．このとき，式 (30) は

(39) に変形することができる．ここで，確率密度関数 (40) である．つまり，自由エネルギー F β (mj) のパラメータ mkj に関する最小化は，対数ゆう度関数に関する最大化に等価であることを示すことができる．このような対数ゆう度関数と自由エネルギーとの関係は，EM アルゴリズムに関しては既に詳しい議論がされている [12]．以上のことから，動径基底関数の複製を考慮した RC-RBFN の学習則を次のように提案する．[RC-RBFN の学習則] STEP 1. シナプス結合荷重 wj を式 (8) のシナプス可塑性方程式により更新，パラメータ mkj を式 (32) の Δ 0mkj により更新，パラメータ Σ j は式 (7) により更新する．STEP 2. 累積 2 乗誤差関数がとなったら学習終了．ある正数 ¿ 0 より大きな値に収束したなら STEP 3. へいく．STEP 3. 学習収束時に得られているすべての動径基底関数について，β を 0 から徐々に大きくしていきながら，式 (32) に従いパラメータ mkj を更新する．STEP 4. 分岐により Δ β mkj = 0 となる点が増えたとき，第 j 動径基底関数を第 p 動径基底関数として複製する．そのとき，シナプス結合荷重 wp，パラメータ Σ p 並びにパラメータ mnp (n —= k) は形質としてもとの第 j 動径基底関数のものを引き継ぎ，パラメータ mkp は新たに増えた点とする．STEP 1. へ戻る．

ここで，本論文で展開した議論を従来の RBFN に対して適用すれば，動径基底関数を効率的に追加することが可能な学習則が導けること端的に述べる．式 (5) で与えられる Delta ルールに基づいて導出された RBFN のシナプス結合荷重の更新則を内的自然増加率 α j (φ)，並びに競争係数 γ jk(φ) を用いて記述すると

(A·1) となることがわかる．これは積分条件

(A·3) を考えることができる．ただし， は省略した．この結果，(A·4) であることが示されるので，やはり累積 2 乗誤差関数 E(w, φ) の最小化はポテンシャル V (w, φ) の最小化と等価であることがわかる．その結果，学習終了時に RBFN の平均ベクトル，共分散行列とシナプス結合荷

重が満たす同時確率密度 p β (w, φ) を考えることができ，自由エネルギーを定義することで式 (32) に相当する平均ベクトルの更新則が導出できる．詳しい展開は冗長なため省略する．もちろん，式 (5) で与えられる RBFN のシナプス結合荷重の更新則がそのままで積分条件を満たすことから，ポテンシャル V (w, φ) を導出せずに，累積 2 乗誤差関数 E(w, φ) をポテンシャルとみなしても同様の議論が可能である．しかしながら，式 (5) とそれから得られる追加アルゴリズムを利用した学習則では動径基底関数を削除する効果は望めない．そこで，本論文では適者生存型学習則に対する動径基底関数の複製について議論を展開した．

ターミナルアトラクタの話 (適者 3.2)(5.1 かも?)

§ 2.3　化学・酵素データベース

化学・生物分野において用いられているデータベースについて，いくつか説明する．

Kyoto Encyclopedia of Genes and Genomes(KEGG) [?]

遺伝子・タンパク質情報，タンパク質相互作用を可視化したKEGG PATHWAY，酵素情報を表したKEGG ENZYME，主に酵素反応の反応式について記したKEGG REACTION，生態系に関連する化合物を集めたKEGG COMPOUND等のデータからなるデータベースである．KEGG ENZYMEでは各酵素の情報を該当するEC番号から検索して得ることができ，酵素の別名，その酵素を用いた生体内の反応式，基質・生成物情報，遺伝情報，文献情報等について書かれている．KEGG REACTIONには酵素を用いて起こる化学反応についての情報を記している．それぞれの反応はRから始まる5桁の数字で管理されており，反応に用いられる酵素とEC番号，化合物名・C番号・構造式でそれぞれ表した反応式等が書かれている．KEGG COMPOUNDではCから始まる5桁のC番号で化合物を管理しており，主にKEGG PATHWAY中やKEGG REACTION中に現れる化合物を扱っている．C番号，名前，分子式で検索することができ，そのリンク先には，別名，分子式，分子量，構造式，登場するR番号，PATHWAY MAPのMAP番号，EC番号のリンク先，他のデータベースへのリンク先などが掲載されている．サイト内のリンクのつながりによって，EC番号からR番号，R番号からC番号とたどることができる．図??および図??にKEGGデータベースの例を示す．

SciFinder^n [?]

Chemical Abstracts Service(CAS)が提供する，データベース．主に，「Substances(化学物質情報)」，「Reactions(反応情報)」，「References(文献情報)」，「Suppliers(カタログ情報)」，「Biosequences(配列情報)」の項目から検索することができる．

「Substances」では化学物質の名前，CAS登録番号，分子式やスペクトル，物性値などで検索できる [?].「Reactions(反応情報)」では化学物質名，構造式などから検索され，その化合物が反応式・生成物として用いている反応式を調べることができる．また，生成物の収率，反応に用いる試薬や溶媒，文献情報などを条件に入れてフィルター検索もできる．「References」ではキーワード，著者名，文献番号，雑誌情報，機関名などで検索される．「Suppliers(カタログ情報)」では，検索した化合物を取り扱う企業などのカタログ情報を取り扱っている．検索結果には取扱業者名と純度情報，化合物の購入サイトへのリンクと取り扱い分量等が表示される．「Biosequences」ではDNA,RNA, タンパク質の配列情報や類似する配列などで検索される．

SciFinder^{n}では，構造式をユーザ自信が描画・編集して検索することが可能である．化合物の構造が一致するもの，または構造の類似度に基づいて検索できる他，関連する反応式，文献情報，提供元の情報も参照できる．さらに，作成した構造式を生成物として，逆合成ルートを設計・予測する「Retrosynthesis Planner」というツールが存在する．ここでは合成ステップ数やコストなどを設定し，複数パターンの合成プランが設計される．各合成ルートは既知の反応または，予測された反応で構成され，最大の収率が表示される．図??にモルヌピラビルをターゲットとして，逆合成ルートの設計・予測をした様子を示す．

PubChem [?]

化合物名，分子式，化合物の2D(もしくは3D)形式の構造イメージ，化学・物理特性，生物学的活性情報，毒性情報，文献情報等のデータを収録している．データ提供者からアッ

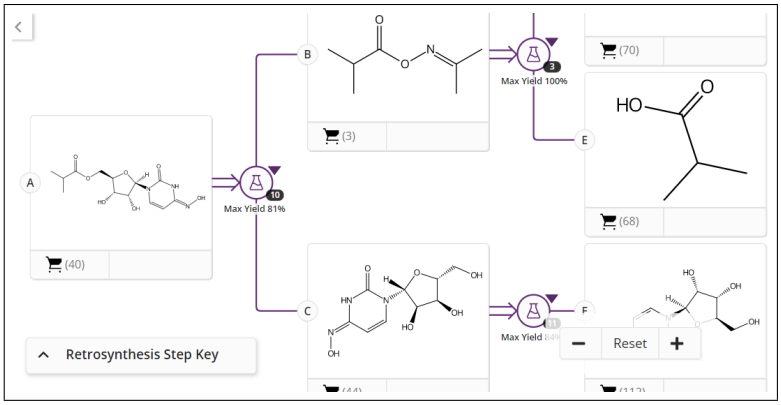
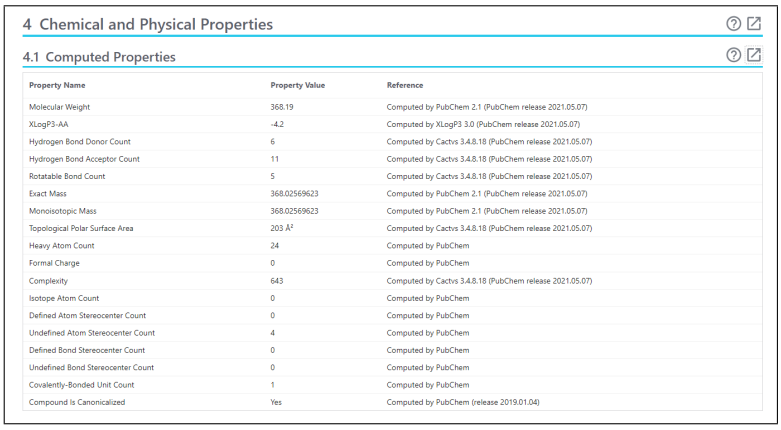


図 2.1: SciFinder^n による逆合成設計予測



4 Chemical and Physical Properties

4.1 Computed Properties

Property Name	Property Value	Reference
Molecular Weight	368.19	Computed by PubChem 2.1 (PubChem release 2021.05.07)
XLogP3-AA	-4.2	Computed by XLogP3 3.0 (PubChem release 2021.05.07)
Hydrogen Bond Donor Count	6	Computed by Cactvs 3.4.8.18 (PubChem release 2021.05.07)
Hydrogen Bond Acceptor Count	11	Computed by Cactvs 3.4.8.18 (PubChem release 2021.05.07)
Rotatable Bond Count	5	Computed by Cactvs 3.4.8.18 (PubChem release 2021.05.07)
Exact Mass	368.02569623	Computed by PubChem 2.1 (PubChem release 2021.05.07)
Monoisotopic Mass	368.02569623	Computed by PubChem 2.1 (PubChem release 2021.05.07)
Topological Polar Surface Area	203 Å²	Computed by Cactvs 3.4.8.18 (PubChem release 2021.05.07)
Heavy Atom Count	24	Computed by PubChem
Formal Charge	0	Computed by PubChem
Complexity	643	Computed by Cactvs 3.4.8.18 (PubChem release 2021.05.07)
Isotope Atom Count	0	Computed by PubChem
Defined Atom Stereocenter Count	0	Computed by PubChem
Undefined Atom Stereocenter Count	4	Computed by PubChem
Defined Bond Stereocenter Count	0	Computed by PubChem
Undefined Bond Stereocenter Count	0	Computed by PubChem
Covalently-Bonded Unit Count	1	Computed by PubChem
Compound Is Canonicalized	Yes	Computed by PubChem (release 2019.01.04)

図 2.2: PubChem の例

プロードされた，約 2.8 億種の化学物質情報や約 140 万種の生物学的実験データ，標準化された約 1.1 億種の化学構造情報，また，約 10 万種の遺伝子データなどから構成される [?]．さらに，PubChem Compound，PubChem Substance，PumChem BioAssay の 3 つのデータベースがある．

PubChem Substance では，研究者がアップロードしたデータを管理している．複数の提供者から重複するデータがアップロードされることがあるため，標準化によって，同様の情報を集約し，PubChem Compound に格納される [?]．また，PumChem BioAssay では，データ提供者の実験環境によってばらつきが生じる生物活性データ等を，実験に用いられた化合物と，実験結果ごとに紐づけを行うことで管理している．それぞれのデータベース中のデータには，SID(SubstanceID)，CID(CompoundID), AID(AssayID) が割り振られている．特に SID は KEGG のほとんどの C 番号と対応している．図??に PubChem のデータベースの例を示す．

BRENDA [?]

酵素に関するデータを，文献の情報をもとに網羅したデータベース．酵素名，生物種，CAS 登録番号，EC 番号，特性値などで検索することができる．例として，EC 番号で検索した様子を図??に示す．検索した EC 番号のページに行くと，その酵素に関係している単語の

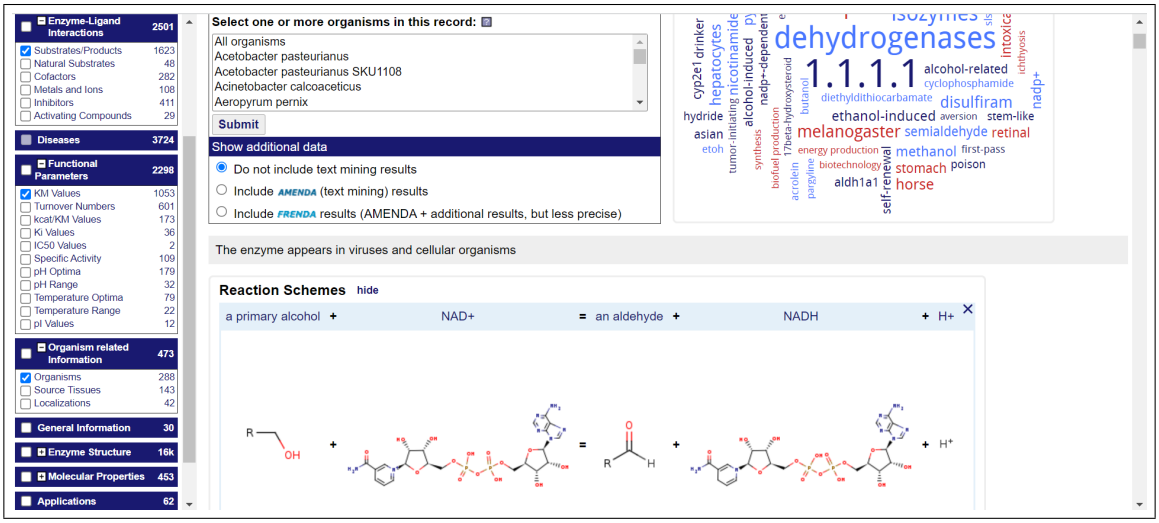


図 2.3: BRENDA 上の EC1.1.1.1 に関する情報

ワードマップや用いられている反応式が書かれている．図??の画面左にある画面から目的とする詳細情報を表示できる．例えば，Substrates/Products では，検索した EC 番号の酵素を使った反応の基質・生成物のペアが記されている．Organisms では，酵素を作る由来となった生物種のリストが表示されている．また，「Functional Parameters」ボックス内の KM Values では，酵素の由来となった生物種・基質ごとの Km 値 (基質と酵素の親和性を表す指標) を見ることができる．

第3章

ケモインフォマティクスと情報技術

§ 3.1　化学データベースからの情報抽出

Web サイト等から収集した大量の情報の中から，自然言語処理を用いて有用な情報を抽出するテキストマイニングにおいては，スクレイピングが用いられることがある．スクレイピングとは，Web サイトから文章をプログラミングによって自動取得する方法であり，効率的にデータを収集できる．一方でデータベースを管理している Web サイト等においては，独自のアプリケーション・プログラミング・インターフェース (Application Programming Interface: API) を備えている場合があり，指定された形式でプログラムを記述すれば，データベース上の情報を自動的に取得することができる．

化学データベースにも公式の API が公開されているものがいくつか存在する．KEGG では KEGG API [?]，PubChem では POWER USER GATEWAY(PUG) [?] と呼ばれる API が公開されており，本節ではこの 2 つの API について説明する．

KEGG API の構成

KEGG API のフォーマットは以下のようになる [?]．<operation>の部分に上記の 7 つのいずれかを指定する，例えば，「list」を指定した場合．以下のフォーマットに従う．<dbentries>で目的のデータがある KEGG データベース名を指定する．例えば，「pathway」を指定することで，完成するリンク先へ行くと，各 Pathway のマップ番号と，Pathway 名の対応リストを取得できる．図??に番号と Pathway 名の対応表を示す．このように，「http://rest.kegg.jp/」以下の部分で指定された識別子を設定することで，データが保存されている URL に移動することができ，各プログラム言語で実装されている，リンク先の中身を取得するコードによって，必要なデータを取得することができる．

PUG

Common Gateway Interface(CGI) を経由して，PubChem のデータをプログラミングによって，取得する機能を提供するシステム [?]．データのやり取りは URL ではなく XML を用いる．XML によるリクエストを CGI へ送り，リクエストの内容が実行された後，結果が

```
http://rest.kegg.jp/<operation>/<argument>[/<argument2[/<argument3> ...]]
<operation> = info | list | find | get | conv | link | ddi
```

図 3.1: KEGG API の URL 構成 1

```
http://rest.kegg.jp/list/<dbentries>

<dbentries> = Entries of the following <database>
<database> = pathway | brite | module | ko | genome | <org> | vg | vp | ag |
    compound | glycan | reaction | rclass | enzyme | network | variant |
    disease | drug | dgroup | <medicus>
```

図 3.2: KEGG API の URL 構成 2

表 3.1: リンク先の対応表

path:map00010	Glycolysis / Gluconeogenesis
path:map00020	Citrate cycle (TCA cycle)
path:map00030	Pentose phosphate pathway
path:map00040	Pentose and glucuronate interconversions
path:map00051	Fructose and mannose metabolism
path:map00052	Galactose metabolism
path:map00053	Ascorbate and aldarate metabolism
path:map00061	Fatty acid biosynthesis
path:map00062	Fatty acid elongation
path:map00071	Fatty acid degradation
path:map00073	Cutin, suberine and wax biosynthesis
path:map00100	Steroid biosynthesis
path:map00120	Primary bile acid biosynthesis
path:map00121	Secondary bile acid biosynthesis
path:map00130	Ubiquinone and other terpenoid-quinone biosynthesis
path:map00140	Steroid hormone biosynthesis
path:map00190	Oxidative phosphorylation
path:map00195	Photosynthesis
path:map00196	Photosynthesis - antenna proteins
path:map00220	Arginine biosynthesis
path:map00230	Purine metabolism
path:map00232	Caffeine metabolism
path:map00240	Pyrimidine metabolism
path:map00250	Alanine, aspartate and glutamate metabolism
path:map00253	Tetracycline biosynthesis
path:map00254	Aflatoxin biosynthesis
path:map00260	Glycine, serine and threonine metabolism
path:map00261	Monobactam biosynthesis
path:map00270	Cysteine and methionine metabolism
path:map00280	Valine, leucine and isoleucine degradation

XML で返信される仕組みとなっている．例として，CID1 と CID99 の化合物の構造を SDF ファイル形式の gzip 圧縮でダウンロードする場合，図**??**のような XML 構造のリクエスト応答となる．PubChem ではアクセス簡略化のため，PUG-SOAP と PUG-REST というシステムが実装されている．本研究では PUG-REST を用いるため，PUG-REST について説明する．

PUG-REST

PUG や PUG-SOAP で用いられている XML 形式の記述を必要とせず, 簡単な記述方でデータを取得することができる API．PUG-REST のリクエストは以下のような URL で表記される [**?**]．<input specification>はさらに<domain>/<namespace>/<identifiers>で構成されており, 何のデータを取ってくるのかを定める．<domain>では, substance, compound, assay などの対象とするデータベースを指定する．また，<namespace>では CID(cid) や化合物名 (name), 分子式 (formura) 等を指定し, <identifiers>では, CID の番号, 化合物名・分子式の文字列といった，<namespace>に対する具体的な名前を指定する．<operation specification>では<input specification>で指定したデータ保管場所にアクセスした際, どのような操作を所望しているのかを記述する．例えば, <input specification>で CID 番号の情報を記述している状態で, synonyms を指定するとその CID 番号の化合物名に対する同義語のリストが返される．同様のケースで，<compound property>で property/XXX,YYY,$\cdots$,ZZZ/を指定すると，その化合物の物性値や化学的特性値を複数取得することができる．<output specification>の

```
https://pubchem.ncbi.nlm.nih.gov/rest/pug/<input specification>/
<operation specification>/[<output specification>][?<operation_options>]

<input specification> = <domain>/<namespace>/<identifiers>
<operation specification> = record | <compound property> | synonyms | sids |
    cids | aids | assaysummary | classification | <xrefs> | description |
    conformers
<output specification> = XML | ASNT | ASNB | JSON | JSONP [ ?callback=<
    callback name> ] | SDF | CSV | PNG | TXT
```

図 3.3: PUG-REST のリクエスト

図 3.4: PUG における XML 応答の例 [?]

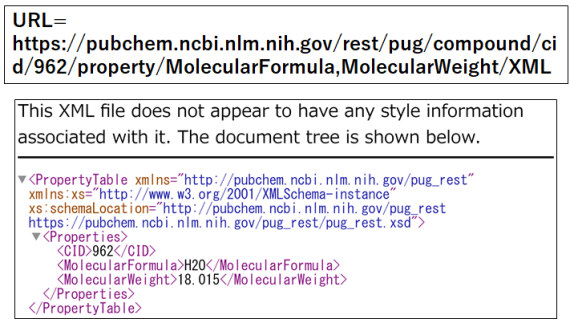


図 3.5: 水 (H_2O) の情報を取得するリクエスト URL とデータ取得結果

部分では取得したいデータをどのような形式で出力するかを指定する．基本的には，<input specification>/<operation specification>/<output specification>の部分を指定すれば良く，例として，水 (CID968) の分子式 (MolecularFormula) と分子量 (MolecularWeight) を XML で取得した場合を図??に示す．

§ 3.2　化合物の構造表現法と EC 番号予測手法

化合物同士の構造比較について述べる前に，ケモインフォマティクスで一般的に使われている化合物の構造表現について説明する．

MOL ファイル

化合物の構造情報を記したテキスト形式のファイル．「.mol」の拡張子で保存されることが多い．ファイル内には結合している原子と各原子の 3 次元座標リストやどの原子同士が結びついているかのリストが記述されている．通常の構造式と mol ファイルを比較したものを図??に示す

SDF ファイル

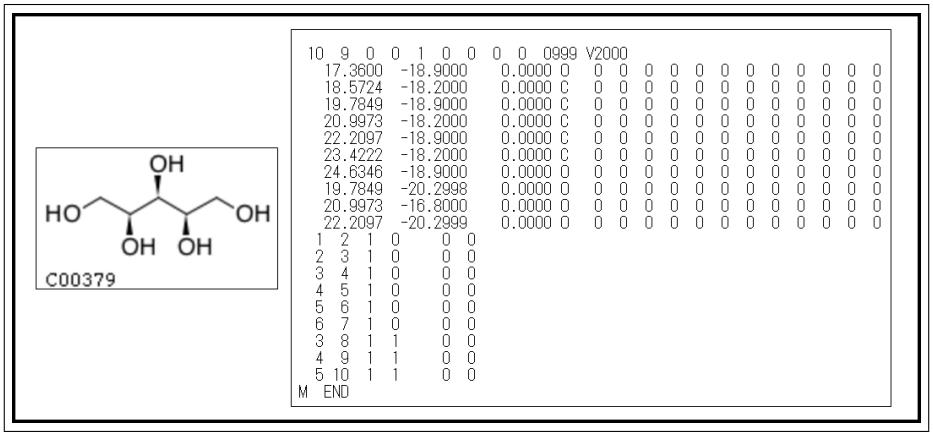


図 3.6: KEGG COMPOUND で取得できる構造式と MOL ファイルの例

MOL ファイルを複数統合した拡張子「.sdf」のファイル．2 つ以上の分子の MOL ファイルをデータベースから同時に入手する際は，この形式となることが多い．

SMILES

化合物の構造を文字列で表したもの．以下の規則に従って文字列に変換していく [?].

1. 原子は元素記号で表し，2 文字で区別がつきにくい原子 (Nb と NB 等) は [ ] で囲む
2. 水素原子は省略する
3. 隣接する原子は隣に記す
4. 二重結合は =，三重結合は#で表し，単結合・芳香族結合は省略する (芳香族原子は小文字の c などで表記する)
5. イオンなどで結合がない部分は「.」で分ける
6. 構造が分岐する箇所は ( ) で表記する
7. 環構造は切断して切断箇所を記すとともに (C1 など)，鎖錠構造で表す．

以上の規則に基づいて作成されたものを generic SMILES と呼ぶが，以下の規則を加えたものを isomeric SMILES と呼ぶ

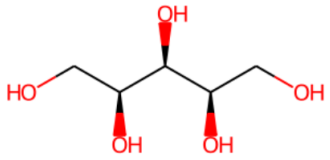
1. 同位体 (例えば炭素) がある場合 [13C] という表記にする
2. 立体異性体を区別するための絶対配置を「@」または「@@」で表現する
3. 二重結合などで生じる幾何異性を「/」と「¥」で表す

フィンガープリント

化合物の構造や特徴をビット列で表現したもの．構造のどの部分に注目するか，または性質などで種類がある．例えば，MACCS Keys のフィンガープリント [?] では，166 種の特徴的な構造を化合物が持っているかどうかを 0 と 1 で表現している．フィンガープリントは主に化合物同士の類似性比較で用いられる．

化合物の数値化 (特徴ベクトル)

```
from rdkit import Chem
Xylitol = Chem.MolFromMolFile('Xylitol.mol')
Xylitol
```




```
Chem.MolToSmiles(Xylitol)
```

```
'OC[C@H](O)[C@@H](O)[C@H](O)CO'
```

```
from rdkit import Chem
from rdkit.Chem import Descriptors
Xylitol = Chem.MolFromMolFile('Xylitol.mol')
Descriptors.MolWt(Xylitol)
```

```
152.14600000000002
```

図 3.7: rdkit を用いた化合物の情報

```
import pubchempy
pubchempy.get_properties([ 'MolecularFormula','MolecularWeight',
                           'IsomericSmiles' ], 'glucose',
                          'name', as_dataframe=True)
```

CID	MolecularFormula	MolecularWeight	IsomericSMILES
5793	C6H12O6	180.16	C([C@@H]1[C@H]([C@@H]([C@H](C(O1)O)O)O)O)O

図 3.8: PubChempy でグルコースの情報を取得した結果

機械学習で様々な予測を行うためには，化合物を数値化して表現する必要がある．その方法として，前述のフィンガープリントではビット列で化合物を数値化しているが，物性値を特徴として用いられることも多い．一般的に複数の物性値が用いられ，多次元の特徴ベクトルとして化合物の特徴を表現する．これらの構造情報や物性値で化合物の特徴を表したものは記述子と呼ばれている．

RDKit [?] を用いた化合物のデータ化

RDKit は Python 提供されている，化合物の構造を扱うライブラリである．SDF ファイルや MOL ファイルを読み込んで構造式の画像を出力したり，SMILES やフィンガープリントに変換することができる．RDKit では読み込んだ構造式から，化合物の記述子を計算することができるため，化合物同士の類似性を評価したり，機械学習に発展させることができる．例として，化合物の分子量を意味する MolWt を知りたい場合，MOL ファイルから読み込んだ化合物のインスタンスを生成し，RDKit の Descriptors クラスにある MolWt メソッドに生成したインスタンスを渡すことで，MolWt が計算され出力される．図??に，rdkit を用いて化合物の構造式と SMILES を出力した様子，および化合物の MolWt を計算した結果を示す．

PubChemPy [?]

PUG REST を用いて PubChem のデータを取得するための Python ライブラリ．化合物名や CID を引数にして，対象化合物の物性値や SMILES を取得することができる．例として，グルコースの分子式，分子量，IsomericSMILES を取得した結果を図??に示す．

EC 番号予測手法

上記で紹介した構造表現法や化学・酵素データベースを用いて，酵素反応の予測や分類を行う研究が多く行われている．2.2 で示した通り，酵素は EC 番号によって管理されているが，同時にその酵素を用いた代表的な化学反応が反応式として登録されている．ここで，代表的な化学反応とは，生体内など自然界で起こる反応を指し，各 EC 番号に 1 つまたは複数登録されている．例として KEGG ENZYME の EC3.1.1.2 では代表的な反応式 3 種が R

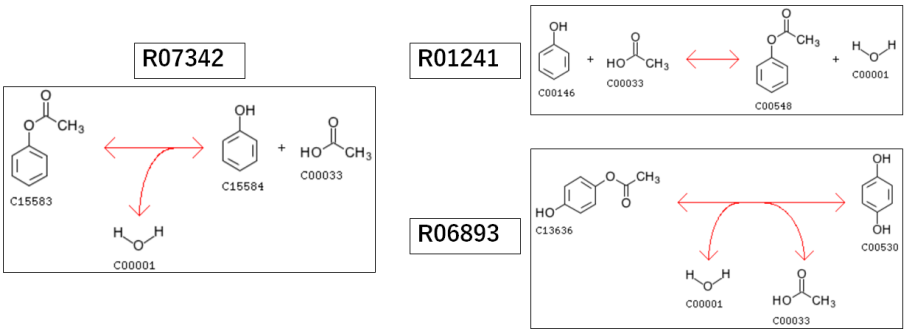


図 3.9: EC3.1.1.2 の代表的な反応式

番号として表記されており，図??のような化学反応となる．これらの反応式に対して，EC 番号の分類問題を考え，より多くの反応式を正しい EC 番号に分類できるように分類モデルを検討する研究が主に行われている．以下で 2 つの手法を示す．

EC 番号予測手法の 1 つとして，アミノ酸配列の類似性を用いるものがある．酵素はタンパク質であるため，アミノ酸配列で表現される．アミノ酸配列の類似性に基づいて，該当する EC 番号を予測する．もう 1 つの手法として，基質と生成物の構造に着目したものがある．構造をフィンガープリントなどで表したものや [?]，構造として特徴的な部分の化学変化に注目したもの [?] などがある．

フィンガープリントを用いる手法では，各基質と生成物を分子の部分構造 (フラグメント) に着目したフィンガープリントで表している．その後，基質フィンガープリントから生成物フィンガープリントを引いた反応差分フィンガープリントを定義する．そして，EC 番号が正解ラベルとして与えられている反応差分フィンガープリントとのユークリッド距離を求め，最小距離となるものの EC 番号を割り当てるという方法を用いている．例えば，KEGG REACTION の R00005 に登録されている反応式 $C01010+C00001 <=> 2C00011+2C00014$ に対して，各分子の分子フィンガープリントを MFP として，反応差分フィンガープリント RFP を以下のように定義している．

$$RFP_{R00005} = MFP_{C01010+C00001} - MFP_{2C00011+2C00014} \tag{3.1}$$

構造の特徴的な部分の化学変化を用いる手法では，RDM パターンと呼ばれる，基質と生成物の各構造に対して，反応中心原子 (R atom)，その近傍の原子で異なっている領域 (D atom) と一致している領域 (M atom) を定義している．EC 番号の基質と生成物の RDM パターンと，入力した反応の RDM パターンの類似性を比較することで，入力反応の EC 番号を予測している．

§ 3.3　クラスタリング手法

本研究では 2 つのクラスタリング手法を用いるが，それに伴いここではクラスタリングについて述べる．クラスタリングは観測されたデータのみを扱う教師なし学習の一つで，特定の基準に従い類似しているデータどうしでクラスタを形成し，分類する手法である．データが 1 つのクラスタのみに属するクラスタリングをハードクラスタリングと呼ばれており，

種類によっては，複数のクラスタに属することを許容するソフトクラスタリングも存在する．クラスタリングは主に階層的クラスタリングと非階層的クラスタリングに分けられる．階層的クラスタリングではさらに，分割型のものと凝集型のものに分けられる．分割型ではデータを全て 1 つのクラスタとみなしたのち，細かいクラスタに分割していく手法である．凝集型では，データそれぞれを 1 つのクラスタとみなし，特定の基準にしたがって複数データが属するクラスタを形成する．複数データを持つクラスタ同士も連結され，新たなクラスタを形成し，指定したクラスタ数になるまで繰り返される．

階層型では凝集型が主に用いられ，以下では，凝集型におけるクラスタを形成していく基準について述べる．なお，クラスタ C_1，C_2 に属するデータの集合をそれぞれ $\mathbf{x_1}$,$\mathbf{x_2}$，$\mathbf{x_1}$ と $\mathbf{x_2}$ の距離を $d(\mathbf{x_1}, \mathbf{x_2})$ としたときのクラスタ間の距離を $d(C_1, C_2)$ とする [?].

最短距離法

2 つのクラスタ内のデータどうしで，最も距離が近い組を基準として，新しきクラスターを作成する．計算量は少ないが，外れ値に弱いとされている．

$$d(C_1, C_2) = \min_{\mathbf{x_1} \in C_1, \mathbf{x_2} \in C_2} \{d(\mathbf{x_1}, \mathbf{x_2})\} \tag{3.2}$$

最長距離法

最短距離法に対して，最も距離が遠い組を基準としたもの，外れ値には弱いが，クラスタサイズが一定になる傾向がある．

$$d(C_1, C_2) = \max_{\mathbf{x_1} \in C_1, \mathbf{x_2} \in C_2} \{d(\mathbf{x_1}, \mathbf{x_2})\} \tag{3.3}$$

群平均法

2 つのクラスタ内の要素同士の距離を合計し，各クラスタサイズで割った平均を基準としたもの．外れ値の影響が少なく．クラスタが帯状に並ぶ鎖効果が起こりにくいとされている．

$$d(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{x_1 \in C_1} \sum_{x_2 \in C_2} d(x_1, x_2) \tag{3.4}$$

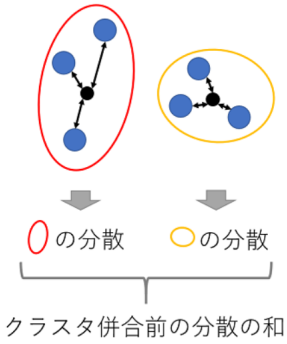
ウォード法

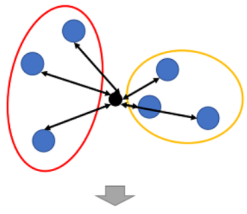
あらかじめ 2 つのクラスタを結合し，結合したクラスタ内の重心に対する，データの分散 $E(C_1 \cup C_2)$ に対して，結合前の各クラスタ内のデータの分散 $E(C_i)$ を引いた差が，最小となるクラスタのペアを結合する方法．計算量は多くなるものの，分類感度が良いとされ，階層的クラスタリングで最も用いられている．ウォード方のイメージを図**??**に示す．クラスタ C_1 の重心を $\boldsymbol{c_i}$ として，以下のように表される．

$$\boldsymbol{c_i} = \sum_{\boldsymbol{x} \in C_i} \frac{\boldsymbol{x}}{|C_i|} \tag{3.5}$$

$$E(C_i) = \sum_{\boldsymbol{x} \in C_i} d(\boldsymbol{x}, \boldsymbol{c_i})^2 \tag{3.6}$$

$$d(C_1, C_2) = E(C_1 \cup C_2) - E(C_1) - E(C_2) \tag{3.7}$$





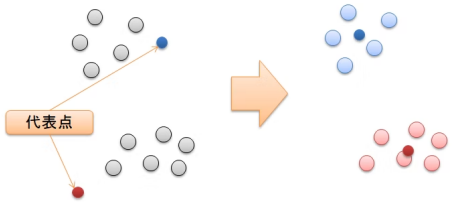


図 3.11: k-means 法のイメージ [?]

図 3.10: ウォード法のイメージ [?]

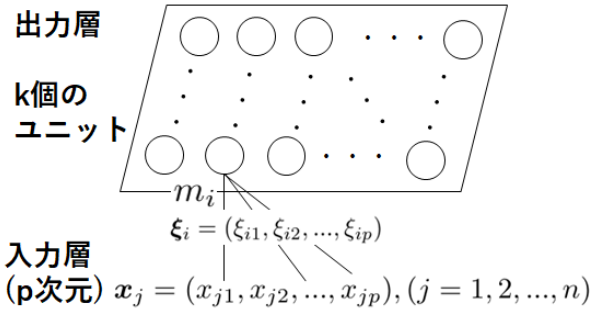


図 3.12: SOM における入力データのマッピング

非階層的クラスタリング

非階層的クラスタリングでは，あらかじめクラスタリング数を決めておき，各手法で定められている基準にしたがって，データを分類する．非階層的クラスタリングの手法をいくつか以下に示す．

k-means 法

データに対して，ランダムにクラスタを割り振り，重心に基づいてクラスタを再構成していく手法．以下の手順に沿ってクラスタリングを行う．

1. 最初に指定した k 個のクラスタリングに，データ点をランダムに割り振る
2. 各クラスタ内の各データに対して重心を計算し，データ点が最短距離にある重心のクラスタに属するように，データ点へのクラスタを振り直す．
3. 振り直しで全てのデータ点のクラスタが固定されるまで，上記の手順を繰り返す．

自己組織化マップ (Self-Organizing Map: SOM) [?]

多次元データを低次元にマッピングし，可視化するクラスタリング手法．以下そのアルゴリズムを示す [?]．n 個の p 次元観測ベクトル $\boldsymbol{x}_j = (x_{j1}, x_{j2}, ..., x_{jp}), (j = 1, 2, ..., n)$ を，ユニット $m_i (i = 1, 2, ..., k)$ で構成された，2 次元平面上に写像する．図??にそのイメージを示す．このとき各ユニットの重心を $\boldsymbol{r}_i = (r_{i1}, r_{i2})$ とし，これを m_i の位置ベクトルとする．さらに，各ユニットは，重みベクトル $\boldsymbol{\xi}_i = (\xi_{i1}, \xi_{i2}, ..., \xi_{ip}), (i = 1, 2, ..., k)$ を持っているとする．ここで，$\boldsymbol{x}_j$，m_i をそれぞれ，入力層，出力層と呼び，次の手順によって出力層を更新する．(ただし，$\boldsymbol{\xi}_i$ はランダムな値で初期化を行う)

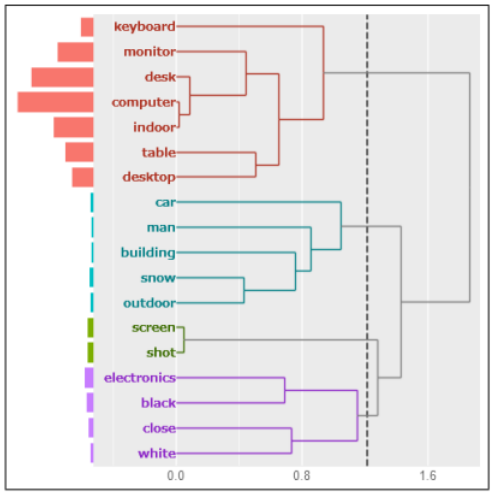


図 3.13: 階層的クラスタリングによる行動識別

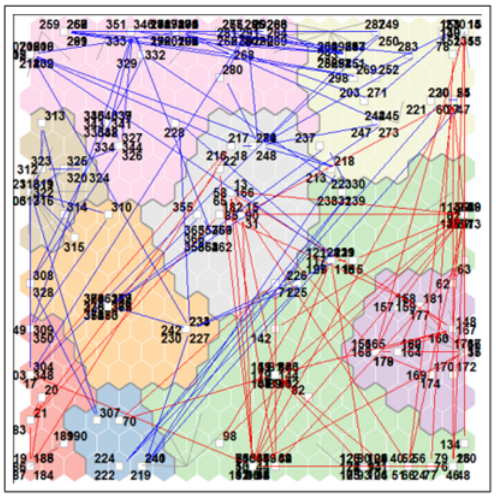


図 3.14: SOM を用いた行動時系列分析

1. $j = 1$ から n までの順に，各 $\boldsymbol{x}_j$ に対してユークリッド距離 $||\boldsymbol{x}_j - \boldsymbol{\xi}_i||$ を求める．
2. $||\boldsymbol{x}_j - \boldsymbol{\xi}_i||$ を最小値にする $\boldsymbol{\xi}_i$ を $\boldsymbol{\xi}_c$ と置く．この $\boldsymbol{\xi}_c$ を持つユニットを勝者ユニット m_c と呼び，勝者ユニット m_c とその近傍のユニットが持つ重みベクトルを次のように更新する．

$$\begin{cases} \boldsymbol{\xi}_i = \boldsymbol{\xi}_i + h(t)\{\boldsymbol{x}_j - \boldsymbol{\xi}_i\} & i \in N_c \\ \boldsymbol{\xi}_i = \boldsymbol{\xi}_i & i \notin N_c \end{cases} \tag{3.8}$$

N_c は m_c の近傍領域を表し，m_c と N_c に含まれる m_i が $\boldsymbol{x}_j$ に近くなるように更新される．また，$h(t)$ は以下で定義される近傍関数であり，m_c が最も $\boldsymbol{x}_j$ に近づくように働きかける．ただし，$\alpha(t)$ を学習率係数 (学習回数 t の増加とともに減少)，$\sigma^2(t)$ は N_c の散らばりに関する調整関数とする．

$$h(t) = \alpha(t) \exp\left[\frac{-||\boldsymbol{r}_c - \boldsymbol{r}_i||}{2\sigma^2(t)}\right] \tag{3.9}$$

3. j で更新した $\boldsymbol{\xi}_i$ を記憶した状態で，$j + 1$ として 1,2 を繰り返す．
4. 3 までを 1 回の学習とし，指定した回数まで学習を行う
5. 学習後，ユークリッド距離 $min||\boldsymbol{x}_j - \boldsymbol{\xi}_i||$ を満たす $\boldsymbol{\xi}_c$ を持つ勝者ユニット m_c に $\boldsymbol{x}_j$ をマッピングする

クラスタリングを用いた研究例として，ヒトの行動パターンを解析し，行動識別を行ったものがある [?]．まず，画像認識 API を用いて，視界に映っている物体を認識し，その物体名をテキストデータに出力している．次に，テキストマイニングデータのクラスタリングを行うソフトウェア KH Corder [?] を用いて，物体名の同時出現頻度に関して，階層的クラスタリングを行っている．それによって，クラスタ内に含まれる物体名から行動全体のイベント性を分析している．また，SOM によるクラスタリングも行われている．観測されたデータから順番に SOM の 2 次元マップ上にプロットしていき，プロット点を線で結んでいくことで，行動の時系列を作り，複数の測定における行動の類似性を分析している．階層的クラスタリングと SOM を用いた行動分析の様子を図??および図??に示す．

第4章

提案手法

§ 4.1　特性値変化量を用いたEC番号予測

医薬品などの新規化合物を開発する分野において，それに必要な有機合成を効率的かつ，なるべく環境に負荷を与えない形で行えるほうが望ましい．その点，生体触媒の酵素を用いると，反応物の特定の部位だけの選択的合成，反応の効率化など，グリーンケミストリーの優れた反応となるため，酵素を用いる機会が増加している．それに伴い特定の反応を行うために最適な酵素を選択することも重要となってきている．一方で，基質特異性などの酵素の性質は，生物分野に関わる内容であるため，有機合成の知識のみでは解決が難しい．酵素研究の専門家と協力して，または，酵素データベースなどを参照して最適な酵素候補の目途をつけ，その後のスクリーニングなどで，1つの酵素に絞っていく．ここで，目的とする反応情報を与えた際に，酵素候補を予測するシステムがあれば，酵素候補の探索にかかる時間を著しく短縮することができ，次のスクリーニングの段階まで研究をスムーズに進めることができる．その様子を図??に示す．また，酵素はEC番号で分類されており，EC番号には生物の体内で起こる酵素を用いた代表的な反応が反応式として記載されている．EC番号の酵素を用いた様々な生物由来の酵素製品が開発されているが，EC番号を予測することで，スクリーニング候補として，そのEC番号内の酵素に絞り込むことができる．

そこで，本研究では，ターゲットとなる反応式を与えた際に最適なEC番号を予測する．すなわち，EC番号内の多数の酵素を生体触媒として最適な酵素候補として提示する．予測の方法として，対象とする反応式 (ターゲット反応式) とEC番号の代表的な反応式 (EC反応式) の構造変化を比較する．図??に比較のイメージを示す．ターゲット反応式で目的とする生成物は，逆合成的な思考でどの反応物を用いれば得られるのか分かっている．ここで，ターゲット反応式における反応物から生成物への構造変化が，EC反応式における反応物から生成物の構造変化に類似しているならば，EC反応式で用いられている酵素をターゲットで使用することで，反応の効率が上がり，高い収率でターゲット生成物が得られるという仮定を置く．これは化学の分野で用いられている類似性の概念 [?] に基づいている．

反応による構造変化を，反応式の類似性の評価指標とした理由として，ターゲット反応式の化合物と，各EC反応式の化合物どうしの比較では反応式の類似性を正確に評価できないためである．例えば，ターゲット反応式の反応物が，あるEC反応式の反応物に最も類似していると評価されたとしても，生成物は異なるEC反応式の生成物に最も類似されていると，評価される可能性がある．また，反応物や生成物は複数ある場合が多いため，より反応式の類似性を評価することが難しくなる．そのような理由から反応による構造変化を類似性の比較として用いる．

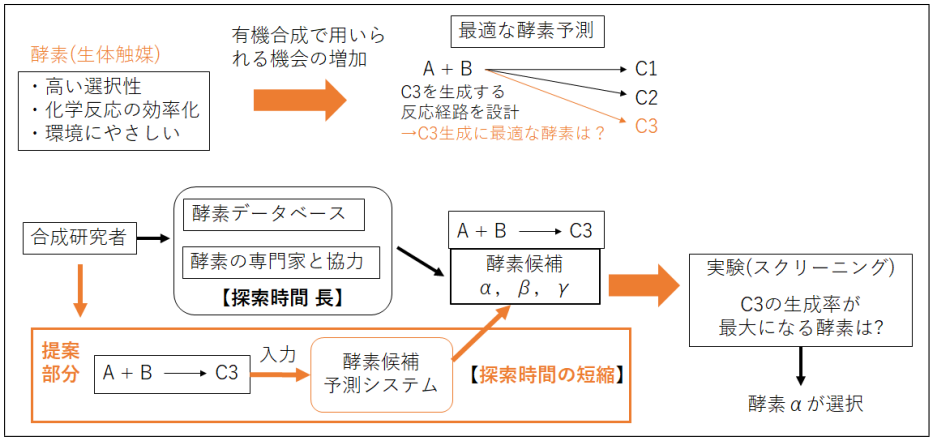


図 4.1: 従来の酵素探索と提案する酵素探索の比較

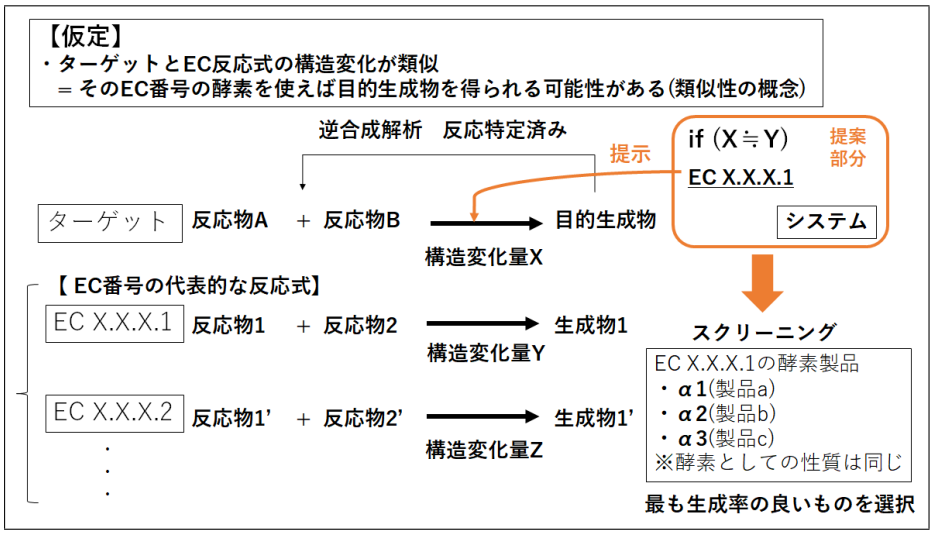


図 4.2: 反応式の類似性比較

反応物から生成物への構造変化を捉える特徴として，記述子を用いた物性値・化学特性値の変化量 (特性値変化量) を用いる．従来研究では，反応物から生成物に変化する際の，構造記述子の変化が，反応変化の特徴として用いられている [?]．ここでは，反応物の生成物の部分構造に注目したフィンガープリントを求め，その差分を比較することで，類似する EC 反応式の酵素を予測している．しかし，フィンガープリントには様々な種類があり，それぞれ化合物のどのような特徴を説明しているのかが異なっている．つまり，1 つのフィンガープリントでは反応変化の特徴を全てとらえるのは難しい．一方で，物理・化学的な特性値を表す記述子も，化合の構造を表現する指標として考えられる．RDKit では 208 種類の特性値に関する記述子が実装されており，読み込んだ分子構造式から簡単に特性値を計算できる．そのため，RDKit 記述子を多数用いて，多次元の特性値変化量を要素に持つ特徴ベクトルを求めることで，ターゲット反応式と EC 反応式の反応時の構造変化を表現する．

差分フィンガープリントと同様に特性値変化量を以下のように定義する．各反応の反応物と生成物の個数をそれぞれ 2 個としたとき，反応物 i の特性値を RT_i，生成物 i の特性値を PD_i とする．このとき，各 n 種の記述子に対する特性値変化量 $cv_j(j = 1, 2, ..., n)$ およ

表 4.1: 各反応式に対する記述子ごとの特性値

	記述子 1	記述子 2	$\cdots$	記述子 n
$\boldsymbol{DF}_1$	cv_{11}	cv_{12}	$\cdots$	cv_{1n}
$\boldsymbol{DF}_2$	cv_{21}	cv_{22}	$\cdots$	cv_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\boldsymbol{DF}_m$	cv_{m1}	cv_{m2}	$\cdots$	cv_{mn}

び，m 個の反応式の特徴ベクトル $\boldsymbol{DF}_i(i = 1, 2, ..., m)$ を以下のように表す．

$$cv_j = (PD_1 + PD_2) - (RT_1 + RT_2) \tag{4.1}$$

$$\boldsymbol{DF}_i = (cv_{i1}, cv_{i2}, ...cv_{ij}, ..., cv_{in}) \tag{4.2}$$

これらをもとに，表**??**のような $i \times j$ の各反応式の特性値表を作成する．行ラベルはターゲット反応式 T と EC 番号，列ラベルは記述子名となる．

特徴ベクトル比較のために必要となる化合物などのデータは，KEGG と PubChem から収集する．これらのデータベースを用いる理由として 2 つ挙げられる．1 つ目は，API でデータを取得するフォーマットが整っていることである．API によって必要となるデータを簡単に取得できることは，プログラミングで自動収集するシステムの，開発のしやすさにつながり，効率的なデータ収集を行える．2 つ目はリンクによってデータベースどうしの行き来がしやすい点にある．異なるデータベースへの参照リンクが多いほど，多種多様なデータを収集をしたり，1 つのデータベース内では見られないデータ間の関係を得ることができる．必要となるデータを API で取得し，集めたデータ関係を分析する，または，新たなデータ関係を見出すデータベースを構築することも可能となる．

KEGG では図**??**のように, KEGG ENZYME, KEGG REACTION, KEGG COMPOUND 間で，リンクによって EC 番号から R 番号，R 番号から C 番号とたどることができる．この関係をもとに EC 番号と代表的な反応式な反応式を構成する各化合物の ID を取得する [?].

PubChem Compound では化合物の特性情報など KEGG にはない情報が記載されており，CID で管理されている．さらに，CID は PubChem Substance において SID とともに併記されていることが多く，SID は KEGG COMPOUND の化合物情報にリンクとして表記されている．これによって，R 番号の C 番号で書かれた反応式からそれぞれの化合物の詳細情報を得ることができる．今回は PubChem から化合物の SMILES 情報を取得し，C 番号と SID・CID の対応によって SMILES 形式の反応式と，EC 番号の対応表を作成する．図**??**に C 番号と SID の対応関係を表す． SMILES 形式の化合物を RDKit で読み込むことで，化合物の構造オブジェクトに変換できる．それにより，構造式をコンピュータ上で表現するとともに，RDKit の記述子を用いて特性値を計算し，化合物を数値で表現する．各反応式において，生成物と反応物の差分を取り，作成した SMILES 反応式と EC 番号の対応表より，EC 番号と各反応式の特性値変化量の特徴ベクトルを取得する．

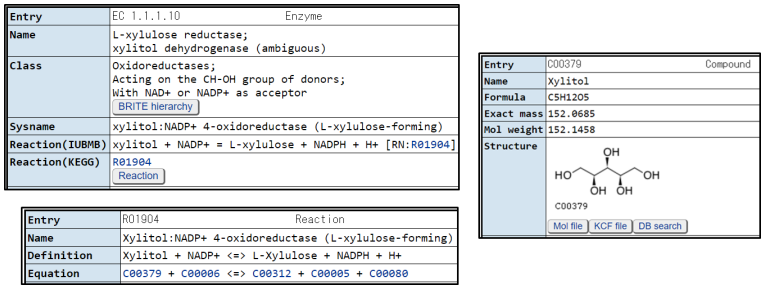


図 4.3: EC 番号，R 番号，C 番号の参照 [?](一部抜粋)

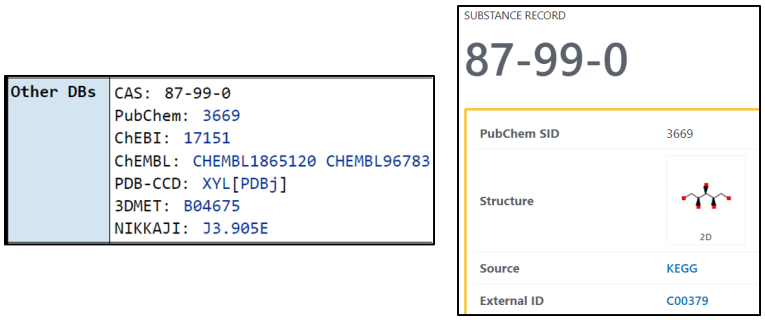


図 4.4: KEGG の C 番号と PubChemSID の対応 [?](一部抜粋)

§ 4.2　凝集型クラスタリングによる次元削減

反応変化の特徴として，特性値変化量からなる多次元の特徴ベクトルを用いるが，次元のサイズが大きすぎることが，問題となるケースがある．一般的には多重共線性や次元の呪いに絡んでくる．多重共線性とは，説明変数間に高い相関があるときに起きる現象で，汎化性能や分類精度の低下の原因とされている．次元の呪いは，用いる変数が多い場合に起こり，過学習の原因とされる問題である．今回のケースでは相関の高い記述子のペアが存在すると，同じような記述子が存在することになり，構造変化の特徴の一部として，他の記述子よりも重みづけが大きくなると考えられる．そのため，多重共線性の問題を解決しつつ，次元の削減も同時に行う．

多重共線性を解決するためには，相関の高いペアの変数に対して，どちらか片方を取り除く方法が取られることが多い．しかし，誤って重要な変数を除去していまう可能性や 3 個以上の変数間の高い相関には対処できない等の問題がある．そのため，相関に基づき，記述子間で凝集性クラスタリングを行うことで多重共線性をなくす方法を用いる [?]．ここでは，最長距離法をクラスタ間の距離としてクラスタリングを行う．図??にそのイメージを示す．

最初の段階では，記述子どうしのマージが行われ，要素数が 2 つのクラスタが形成される．次にクラスタどうしのマージとなり，最長距離法を用いるが，このとき異なるクラスタの記述子間で最も相関の低いペアに注目する．それらのペアの中で相関が最も高いペアのクラスタどうしは，クラスタ間での相関が最も高い関係と考えられる．つまり，最長距離法を用いることで，多数の記述子間の相関を考慮した，多重共線性の対策となる．次元削減としては，同クラスタ内の記述子を合成した合成記述子を作成することで，相関の高

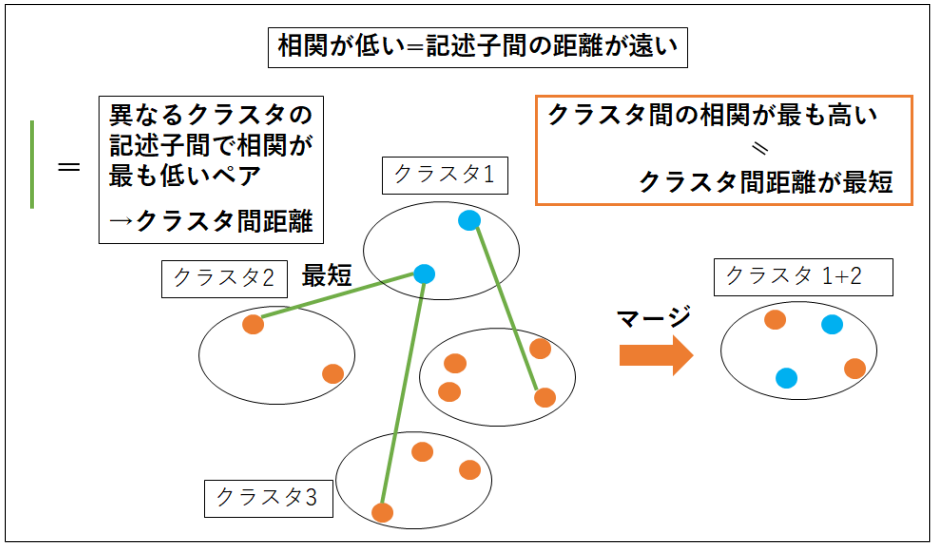


図 4.5: 最長距離法による記述子のクラスタリング

表 4.2: 相関係数の逆数を要素に持つ距離行列

	記述子 0	記述子 2	···	記述子 n
記述子 1	0	$1/s_{12}$	···	$1/s_{1n}$
記述子 2	$1/s_{21}$	0	···	$1/s_{2n}$
⋮	⋮	⋮	⋱	⋮
記述子 n	$1/s_{n1}$	$1/s_{n2}$	···	0

い複数の記述子を新しい 1 つの記述子として表現する．

クラスタプログラムは Python の sklearn に実装されている凝集性クラスタリングである，AgglomerativeClustering ライブラリ [?] を用いる．記述子 u，v 間の相関係数を s_{uv} としたとき，以下のように表される．

$$s_{uv} = \frac{\sum_{i=1}^{m}(cv_{iu} - c\bar{v}_u)(cv_{iv} - c\bar{v}_v)}{\sqrt{\sum_{i=1}^{m}(cv_{iu} - c\bar{v}_u)}\sqrt{\sum_{i=1}^{m}(cv_{iv} - c\bar{v}_v)}} \text{(ただし，} \bar{cv}_u\text{，} \bar{cv}_v \text{ は記述子 } u,v \text{ の特性値平均)} \tag{4.3}$$

s_{uv} に対して，逆数を取った，$1/s_{uv}$ を記述子間の距離とし，Python で表??のような距離行列を作成してクラスタリングする．ここでは，相関係数が 1 となる要素を 0 としている．AgglomerativeClustering では，入力データとして通常の特徴ベクトルだけでなく，距離行列を用いることができ，記述子間でマージするときの閾値を指定することができる．今回は相関係数 $s_{ij} \geqq 0.9$ すなわち，$1/s_{ij} \leqq 1/0.9 \approx 1.11$ で記述子をマージする．クラスター間でのクラスタリングにおいて，最遠距離法を用いたとき，クラスター間距離 $d(C_1, C_2)$ は，式??より以下のようになる．

$$d(C_1, C_2) = \max_{\mathbf{u}\in C_1, \mathbf{v}\in C_2} \frac{1}{s_{uv}} \qquad \text{(ただし，} \frac{1}{s_{uv}} \leqq 1.11\text{)} \tag{4.4}$$

これらを用いて次の手順で記述子間のクラスタリングを行う．

1. $1/s_{ij} \leqq 1.11$ を満たす，記述子のペアにおいて，互いの距離が最短となるものをマージする．
2. $1/s_{ij} \leqq 1/0.9 \approx 1.11$ となる記述子ペアが存在するクラスタ間で，$d(C_1, C_2)$ が最小となるクラスタ C_1，C_2 をマージする．条件を満たす記述子ペアが存在しなくなるまで繰り返す．
3. クラスタリングを終了後，クラスタ番号とそのクラスタに所属する記述子の対応表を取得する．
4. 同クラスタ内の各記述子の特性値列に対し標準化，および平均化を行い，合成記述子を新たな記述子として利用する．

表**??**において，同クラスタの記述子同士をまとめ，合成記述子 clusterX(X はクラスタ番号) と置き換えることで，次元削減を行う．

§ 4.3　SOM による反応式クラスタリング

次元削減した特徴ベクトルを用いて，反応式間の類似度を比較する手法として，SOM によるクラスタリングを行う．SOM を用いることの利点として，2 つ挙げられる．1 つ目は，低次元空間への可視化が可能になる点である．高次元の特徴ベクトルの場合，反応式どうしの位置関係が把握しにくいが，2,3 次元まで圧縮することで，その関係を把握することが可能となる．2 つ目として，クラスタリングによる類似比較が挙げられる．類似度を比較する手法として，コサイン類似度や相関係数等が用いられるが，複数の反応式間の類似度を調べたいときには，直感的な理解が難しい場合がある．その際に，クラスタリングを用いることによって，全ての反応式の類似性を把握することができる．これらのことから，ターゲット反応式の近くに分布する類似性の高い EC 反応式を複数同時に確認できる他，他の反応式どうしの類似性も見ることができるようになる．

用いる SOM のプログラムとして，KH Corder で出力される SOM の R 言語ファイルを参考に作成された，R 言語使用のソースコードを用いる [**?**]．入力するデータは次元削減後の各反応式の特徴ベクトルを全体に対して標準化したものを用いる．SOM のプログラム中には R 言語のパッケージとして実装されている som を使用している [**?**]．プロット点のラベルはターゲット (T) と反応式の EC 番号を扱う．

ユニット数は 400(20×20) であり，ユニットの形状を六角形とする．学習は大まかな順位付けを行う段階と収束段階の 2 段階に分けて行う (KH Coder3 リファレンス・マニュアルに記載)．今回は 1 段階目 1,000 回，2 段階目は 200,000 とした．SOM の実行後，各勝者ユニット上に反応式の特徴ベクトルがマッピングされ，色分けによる凝集型クラスタリングが実行される．このクラスタリングはユークリッド距離によるウォード法によって行われ，今回はクラスタ数 9 で色分けされる．

第5章

実験結果並びに考察

§ 5.1　数値実験の概要

本研究の実験の流れについて説明する．まず, 化合物の特徴ベクトルを求めるため, KEGG と PubChem から各反応式の情報を取得する．次に，反応式内の反応物・生成物の SMILES を出力する．さらに，RDKit にある 208 種の記述子を用いて，化合物の物理・化学特性値を計算し，特性値変化量を求めることで，ターゲット反応式と EC 反応式を，208 次元の特徴ベクトルで表現する．さらに，不適切な値を含む記述子を除外し，凝集型クラスタリングによって相関の高い記述子同士をまとめて，新たな合成記述子を作成することで，次元削減を行う．最後に，SOM によって反応式をクラスタリングし，ターゲット反応式に対して適切な酵素を予測する．

具体的なデータ整理，前処理，および分析条件について以下で説明していく．

ターゲット反応式と提案手法の評価方法

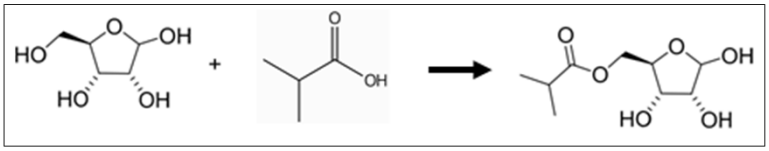
今回は, モルヌピラビルを生成する過程における, 1 ステップ目の合成の反応式に焦点を当てる [?]．図??にターゲット反応式を示す．ターゲット 2 が本来行われた合成であり，リボース (左辺第 1 項) の第一級アルコール部分を選択的にエステル化する反応である．ここでは，8 つの酵素製品に対する，生成物のアッセイ収率を調べるための，スクリーニングを行っている．最終的に Novozym435 の酵素製品が一番優れた結果となったが，これは BRENDA によると EC3.1.1.3 に分類される酵素とされている．特性値変化量が，酵素番号予測を行うために，十分な特徴を備えている場合，この反応式をターゲットとして他の EC 反応式とともに SOM によるクラスタリング行えば，EC3.1.1.3 の反応式がターゲット 2 の付近に位置すると考えられる．

一方で，ターゲット 2 の反応は，通常では起こりえない特殊な反応である．初めに別の反応を試したのち，生成物の収率を上げるために，等価体として類似の性質を持つ化合物に置き換えた，ターゲット 2 を用いたと考えられる．ターゲット 1 も EC3.1.1.3 の酵素を用いた場合に起こりうる反応であり，初めに行った別の反応としてターゲット 1 を仮定する．EC 反応式とそれぞれのターゲット反応式の類似性を SOM のクラスタリングで可視化し，基準として EC3.1.1.3 反応式がターゲットに対してどの場所に位置するかで提案手法を評価する．

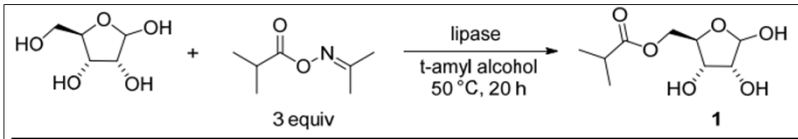
比較対象となる EC 反応式

今回の予測は比較する EC 反応式をあらかじめ絞ったうえで行う．ターゲットの反応はエステル加水分解の逆反応となるエステル化反応のため，用いる酵素として，EC3.1.1 の加

ターゲット1



ターゲット2



Enzyme Name	5-isobutyryl ribose assay yield%
Novozym 435 (Candida antarctica lipase B)	65
IMMTLL-T2-150 (Thermomyces lanuginosus lipase)	40
IMMRES-T2-150 (Resinase HT lipase)	38
IMMLIPX-T2-150 (Lipex 100 L lipase)	56
IMML51-T2-150 (Novozymes 51032)	61
IMMP6-T2-250 (protease from Bacillus licheniformis)	11
Lipozyme RM IM	10
CDX IMB-103	33

図 5.1: ターゲット反応式

水分解酵素が適当であると考えられる．これは，加水分解が一般的には可逆的な反応であり，加水分解酵素でエステル化も可能であるためである．したがって，EC3.1.1 反応式もエステル化する方向 (右辺を反応物，左辺を生成物とする) でターゲットと比較を行う．一方で，全ての EC 反応式に対して比較することも重要であると考えられる．しかし，EC3.1.1 以外の番号で類似していると認識される可能性を考慮し，今回は，EC3.1.1 と認識された場合を仮定し，そのうえで提案した記述子変化量とその記述子選択によって，適切な酵素を予測できるか検証する．

データの対応表取得と整理

まず，KEGG の ID や反応式を取得するソースコードを用いて [?]，EC3.1.1 に属する EC 番号，対応する R 番号，C 番号で構成された R 番号の反応式を取得し，それぞれの対応表を作成した．同様に PubChem からは C 番号と CID，SID の対応表を取得した．次に C 番号と CID または SID を参照し，PubChemPy によって，各反応式の反応物と生成物の SMILES を取得することを試みた．しかし，C 番号に対する CID がまだ登録されていない化合物や，SID を引数にして，PubChemPy から SMILES を取得できないなどの問題が発生した．そこで，PubChem の SID で検索したリンク内で SDF ファイルを入手し，それを RDKit で読み込み SMILES に変換した．また，ターゲットの SMILES は SciFinder^n で入手した MOL ファイルを RDKit で変換することで取得した．それらの SMILES をまとめて，ターゲットおよび EC 番号に対する，各反応物・生成物の SMILES 対応表を作成した．以下，表??，表??, および表??にそれぞれの対応表を示す．

対応表の前処理 1

得られた SMILES 対応表には KEGGC COMPOUND に登録されていない (番号が新しい) 化合物，あるいは登録されているが，構造式が記載されていない化合物が存在する．そ

表 5.1: EC 番号と KCID

	ENZYME	left1	left2	right1	right2	right3
0	3.1.1.22	C04546	C00001	C01089	None	None
1	3.1.1.20	C01572	C00001	C01424	None	None
2	3.1.1.40	C02868	C00001	C01839	None	None
3	3.1.1.33	C02655	C00001	C00031	C00033	None
4	3.1.1.6	C01883	C00001	C00069	C00033	None
...	...	...	...	...	...	...
173	3.1.1.111	C18125	C00001	C22237	C00162	None
174	3.1.1.115	C22218	C00001	C22219	None	None
175	3.1.1.117	C22373	C00001	C22374	C00069	None
176	3.1.1.118	C01194	C00001	C22400	C00162	None
177	3.1.1.118	C00416	C00001	C03974	C00162	None

表 5.2: 各化合物 ID 対応表

	cid	pubchem_SID	pubchem_CID
1	C00001	3303	962
2	C00002	3304	5957
3	C00003	3305	5893
4	C00004	3306	439153
5	C00005	3307	5884
...	...	...	...
18594	C22269	405226444	6365572
18595	C22272	405226445	11788398
18596	C22273	405226446	11411510
18597	C22274	405226447	135567131
18598	C22275	405226448	44468216

表 5.3: EC 番号と SMILES の対応表

	ENZYME	left1	left2	right1	right2	right3
0	3.1.1.22	C[C@@H](O)CC(=O)O[C@H](C)CC(=O)O	[H]O[H]	C[C@@H](O)CC(=O)O	N	N
1	3.1.1.20	O=C(O)c1cc(O)c(O)c(OC(=O)c2cc(O)c(O)c(O)c2)c1	[H]O[H]	O=C(O)c1cc(O)c(O)c(O)c1	N	N
2	3.1.1.40	Cc1cc(OC(=O)c2c(C)cc(O)cc2O)cc(O)c1C(=O)O	[H]O[H]	Cc1cc(O)cc(O)c1C(=O)O	N	N
3	3.1.1.33	CC(=O)OC[C@H]1O[C@@H](O)[C@H](O)[C@@H](O)[C@@H]1O	[H]O[H]	OC[C@H]1OC(O)[C@H](O)[C@@H](O)[C@@H]1O	CC(=O)O	N
4	3.1.1.6	*OC(C)=O	[H]O[H]	*O	CC(=O)O	N
...	...	...	...	...	...	...
173	3.1.1.111	*C(=O)OCC(O)COP(=O)(O)OC[C@H](N)C(=O)O	[H]O[H]	N[C@@H](COP(=O)(O)OCC(O)CO)C(=O)O	*C(=O)O	N
174	3.1.1.115	O=C1OC[C@](O)(CO)[C@H]1O	[H]O[H]	O=C(O)[C@H](O)C(O)(CO)CO	N	N
175	3.1.1.117	N	[H]O[H]	N	*O	N
176	3.1.1.118	*C(=O)OC[C@H](COP(=O)(O)O[C@H]1[C@H](O)[C@@H](...	[H]O[H]	N	*C(=O)O	N
177	3.1.1.118	*C(=O)OCC(COP(=O)(O)O)OC(*)=O	[H]O[H]	*C(=O)O[C@H](CO)COP(=O)(O)O	*C(=O)O	N

のため，対応表内の SMILES の項に空白となる部分が発生するため，その項を含む反応式は除外した．また，ターゲットは反応物 2 個，生成物 1 個の組み合わせであるため，EC 反応式はターゲット同様の組み合わせにする．これは，提案手法の特性値変化量の計算に用いられる化合物の数が増加または減少することで，構造変化とは別の要因による変化が影響すると考えられるためである．つまり，ターゲットの物理・化学特性値の純粋な変化と比較して，化合物の多寡による特性値の変化も追加されると推測されるためである．以上の理由から，ほとんどの反応式に含まれている H_2O を除外した場合の，反応物が 2 個，生成物が 1 個の組み合わせとなる EC3.1.1 反応式 113 種のみを採用した．

物理・化学特性値および記述子変化量の計算

ターゲット+113 種の SMILES 対応表を元に，RDKit の rdkit.chem.descriptor から 208 種の記述子名を取得し，反応式 1，反応式 2，生成物それぞれの場合で特性値を計算した．その後，特性値変化量を求め，図**??**のような 208 次元ベクトルを持つ反応式の表を作成した．

対応表の前処理 2

図**??**の表から，nan 値や発散している要素を持つ記述子を除外した．また，全ての反応式において等しい特性値を持つ記述子を除外し，最終的に 128 種類 (ターゲット 1)，または 129 種類 (ターゲット 2) の記述子で次元削減を行う．

表 5.4: 各反応式の特性値変化量

	MaxEStateIndex	MinEStateIndex	MaxAbsEStateIndex	MinAbsEStateIndex	qed	MolWt	HeavyAtomMolWt	ExactMolWt	NumValenceElectrons
Target	-8.378152	0.949632	-8.378152	-0.144028	-0.330982	-18.015	-15.999	-18.010565	-8
3.1.1.33	-7.632875	0.794822	-7.632875	-1.064815	-0.343138	-18.015	-15.999	-18.010565	-8
3.1.1.6	-6.597222	0.946759	-6.597222	-0.949074	-0.409219	-18.015	-15.999	-18.010565	-8
3.1.1.1	-6.486111	0.972222	-6.486111	-0.675926	-0.331106	-17.007	-15.999	-17.003288	-8
3.1.1.7 3.1.1.8	-7.085822	0.351574	-7.085822	-0.914074	-0.484689	-18.015	-15.999	-18.010565	-8
...	...	...	...	...	...	...	...	...	...
3.1.1.106	-8.896201	0.794521	-8.896201	-0.839784	-0.462056	-18.015	-15.999	-18.010565	-8
3.1.1.113	-6.747917	0.372685	-6.747917	-0.872685	-0.398840	-18.015	-15.999	-18.010565	-8
3.1.1.112	-7.033650	0.317731	-7.033650	-0.979769	-0.421762	-18.015	-15.999	-18.010565	-8
3.1.1.111	-8.902683	0.535378	-8.902683	-0.657129	-0.360318	-18.015	-15.999	-18.010565	-8
3.1.1.118	-8.839073	0.535378	-8.839073	-0.575822	-0.317022	-18.015	-15.999	-18.010565	-8

表 5.5: 記述子間の相関係数に基づくクラスタリング結果 (ターゲット 1)

0	0	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2
Kappa2	Kappa3	fr_Al_COO	fr_COO	NumValeI	Chi0n	Chi0v	Chi1n	Chi1v	Chi2n	Chi2v	Kappa1	LabuteAS	SMR_VSA	SlogP_VS	NumRotaI	MolMR
3	3	4	4	5	5	6	6	6	7	7	8	8	8	9	9	9
NumAliph	RingCoun	FpDensit	FpDensit	SMR_VSA	SlogP_VS	SMR_VSA	VSA_ESta	fr_C_O	NumAliph	NumSatu	fr_Ar_OH	fr_phenol	fr_phenol	fr_alkyl_h	fr_ketone	fr_lactone
10	11	12	12	13	14	14	15	15	16	16	16	16	17	17	17	
fr_ester	fr_ether	MaxEStat	MaxAbsE	NumSatu	fr_NH1	fr_NH2	fr_amide	VSA_ESta	fr_allylic_	VSA_ESta	NumAron	NumAron	fr_benzen	MolWt	HeavyAtc	ExactMolWt
17	17	17	17	17	17	17	18	18	18	19	20	21	22	23	24	25
Chi0	Chi1	Chi3n	Chi3v	Chi4n	Chi4v	HeavyAtc	SMR_VSA	TPSA	NOCount	NHOHCo	EState_V	EState_V	NumHete	fr_COO2	FractionC	VSA_EState4
26	27	28	29	30	31	32	33	34	35	36	36	36	36	36	37	38
VSA_ESta	VSA_ESta	NumHDo	fr_bicyclic	fr_C_O_n	fr_metho	fr_Ar_CO	VSA_ESta	SlogP_VS	fr_unbrch	SlogP_VS	NumAliph	NumSatu	fr_NH0	fr_piperd	EState_V	fr_Al_OH
38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
fr_Al_OH	VSA_ESta	Ipc	PEOE_VS	SlogP_VS	PEOE_VS	HallKierA	PEOE_VS	EState_V	PEOE_VS	fr_ArN	PEOE_VS	EState_V	SMR_VSA	EState_V	PEOE_VS	EState_VSA10
55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
EState_V	PEOE_VS	qed	VSA_ESta	PEOE_VS	fr_aldehy	EState_V	FpDensit	MinAbsE	SlogP_VS	VSA_ESta	EState_V	BalabanJ	PEOE_VS	PEOE_VS	PEOE_VS	SMR_VSA6
72	73	74	75	76	77	78	79									
SlogP_VS	PEOE_VS	BertzCT	PEOE_VS	SMR_VSA	MolLogP	NumHAc	MinEStateIndex									

§ 5.2　実験結果と考察

特徴ベクトルの次元削減

相関係数の逆数である距離行列を入力とした，最遠距離法の凝集型クラスタリングによる次元削減を行った．各ターゲットの場合でそれぞれ 18 個のクラスタが形成され，ターゲット 1 では 80 次元，ターゲット 2 では 82 次元の特徴ベクトルとなった．表??に，ターゲット 1 の場合のクラスタ番号と，そのクラスタに含まれる記述子名の対応表を示す．12 番のクラスタに所属する記述子「MaxEStateIndex」と「MaxAbsEStateIndex」は相関係数が 1 であるが，ともにマージされていることが分かる．また，記述子名が類似している記述子が，同じクラスタに属している傾向があることが分かる．表??にはターゲット 1 の場合に，クラスタリングでマージされた記述子，および次元削減の結果を示す．合成された記述子は，クラスタ番号 X を後ろにつけた「clusterX」で表示されている．また，ターゲットを T とし，EC 番号の下 1 桁のみ表示している，ピリオド以下の番号は，EC 番号の代表の反応式が，複数ある場合の区別に用いられている．アンダーバーで区切られているものは，その反応式が複数の EC 番号間で重複している場合の区別となっている．

SOM による反応式のクラスタリング結果

SOM のプログラムによって，反応式をクラスタリングするとターゲット 1，ターゲット 2 でそれぞれ図??のようになった．E は EC3.1.1 以外の反応式を表す．ターゲット 1 では青色のクラスタが離れているが，これは，SOM が本来は 2 次元平面を円柱状に丸め，さらに

表 5.6: 次元削減後におけるターゲット 1 と EC 反応式の特徴ベクトル

	cluster0	cluster1	cluster2	cluster3	cluster4	cluster5	cluster6	cluster7	cluster8	cluster9	...	PEOE_VSA10	SMR_VSA6	SlogP_VSA10
T	-0.358029	-0.842985	0.204247	4.571328	0.622585	0.207469	-1.696273	2.341995	0.173683	-0.133043	...	0.796801	0.043121	-0.1269
33	-0.223667	-0.842985	0.219299	-0.103504	0.568775	0.207469	0.045521	-0.141173	0.173683	-0.133043	...	0.796801	0.043121	-0.1269
6	3.495807	-0.842985	0.161480	-0.103504	-0.360529	0.207469	0.004312	-0.141173	0.173683	-0.133043	...	0.046341	0.043121	-0.1269
1	3.518113	1.079686	0.207055	-0.103504	-0.497256	0.207469	0.009803	-0.141173	0.173683	-0.133043	...	0.046341	0.043121	-0.1269
7_8	-0.214983	-0.842985	0.219101	-0.103504	0.596503	0.207469	0.036318	-0.141173	0.173683	-0.133043	...	0.796801	0.043121	-0.1269
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
106.1	-0.233937	-0.842985	0.213692	-0.103504	0.310413	0.207469	0.083963	-0.141173	0.173683	-0.133043	...	-0.646994	0.043121	-0.1269
113	-0.251825	-0.842985	0.217973	-0.103504	0.596561	0.207469	0.012651	-0.141173	0.173683	-0.133043	...	0.046341	0.043121	-0.1269
112	-0.182730	-0.842985	0.219101	-0.103504	0.501004	0.207469	0.031504	-0.141173	0.173683	-0.133043	...	0.046341	0.043121	-0.1269
111	-0.280504	1.079686	0.215345	-0.103504	-0.970931	0.207469	0.055048	-0.141173	0.173683	-0.133043	...	-1.333272	0.043121	-0.1269
118	-0.266600	1.079686	0.223487	-0.103504	-1.377476	0.207469	0.075928	-0.141173	0.173683	-0.133043	...	0.046341	0.043121	-0.1269

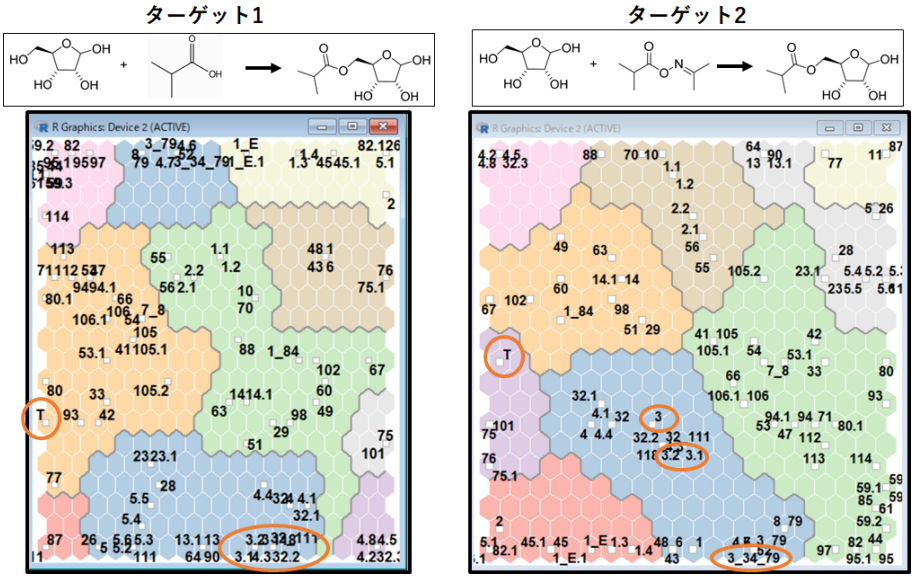


図 5.2: SOM による反応式のクラスタリング結果

円柱を曲げて切り口をつないだトーラス型の形状をしており，2 次元平面にした際に分裂したものと考えられる．ターゲット 1(T) と同じクラスタに属し，かつ付近に位置する EC 反応式として，EC3.1.1.80，EC3.1.1.93 という結果となった．これらのターゲット反応式と比較すると図??のようになった．EC 3.1.1.93 の右辺 (反応物) である C00191 は，リボースと同じ糖類に分類されるグルコースの構造が含まれている．また，ターゲット 2(T) において同じクラスタに属するのは，EC 3.1.1.75(2 つ)，EC 3.1.1.76，EC 3.1.1.101 となった．図??にこれらの反応式を示す．各反応式中には同様の構造が n 個連なる重合体が多く含まれる結果となった．一方で，EC3.1.1.3 の代表反応式は他 EC 番号の重複を含めて 4 種採用しているが，いずれの結果においても全てターゲットと異なるクラスタに属していた．

考察 1

ターゲット 2 を用いた反応式クラスタリングにおいて，特性値変化が類似しているものとして，本実験では，あらかじめ用いられた酵素が分かっているターゲットを使用した．提案手法によって優れた予測が行われているかを検証するため，その EC 番号反応式が最も類似しているものとして，認識されるかを確認した．結果として，目的としていた EC 3.1.1.3

反応式はSOMのマップ上において，ターゲット付近に位置せず，最も類似しているものとして提示されなかった．原因として，以下の4つが挙げられる．

1つ目は反応式中の係数を反映していなかったことが挙げられる．用いられる全ての化合物の比率を平等にした場合でも，特性値変化量は構造変化の特徴として機能すると考えられる．しかし，反応式中の1つの化合物が用いられる分子数だけ特性値を上乗せすることで，より化学的に構造変化を捉えられると考えられる．

2つ目は，ターゲットの反応において，提案した特徴変化量では，捉えきれていない要因が多く影響している点である．モルヌピラビルの論文のサポート資料 [?] では，tert-アミルアルコールを溶媒として用いており，50 ℃で 20 時間振とうを行うことでターゲットの生成物を生成している．また，用いた反応物の分量なども異なっている．一方で，EC 反応式は生物の体内等で起こる反応であり，基本的には有機溶媒等を用いない．実験に用いた試薬や，溶媒，実験環境，配合比率などの様々な要因によって，天然に起こる反応との差異を発生させていると考えられる．特性値変化量だけでなく，反応以外の要因も考慮した特徴作成が必要である．

3つ目は，相関係数に基づくクラスタリングで，同一クラスタに存在する記述子を合成した際に，構造変化に重要な特徴の影響力を弱めてしまった可能性が挙げられる．今回は，相関の高い記述子ペアに対し，片方を除外することで重要な記述子を誤って削除するのを避けるため，クラスタ内の記述子どうしで標準化，および平均化を行った．しかし，やはり重要な記述子と重要でない記述子が混ざったクラスタが存在し，平均化によって，重要な記述子の説明力を薄めてしまった可能性がある．改善策としては，相関係数に応じて形成されたクラスタ内で，記述子の重要度に基づいて，重みづけをする手法を提案できれば，適切な記述子選択と次元削減ができると考えられる．

4つ目として，次元削減後に用いられた記述子は 80 種とまだまだ多く，不必要な記述子によって構造変化を上手く説明できなかったことが考えられる．今回は主に多重共線性の対策としての手法を提案したが，少数かつ構造変化を十分に説明できるような，記述子の組み合わせを提案する手法を検討していきたい．

考察 2

ターゲット 2 と同クラスタの反応式において，多数の重合体が含まれていた点について，ターゲット化合物が重合体の特性と類似点が多い可能性がある．また，EC3.1.1.3 よりも

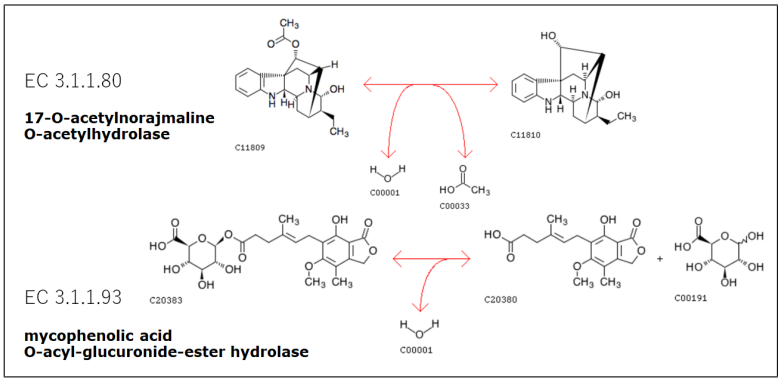


図 5.3: ターゲット 1 の近くに位置している反応式

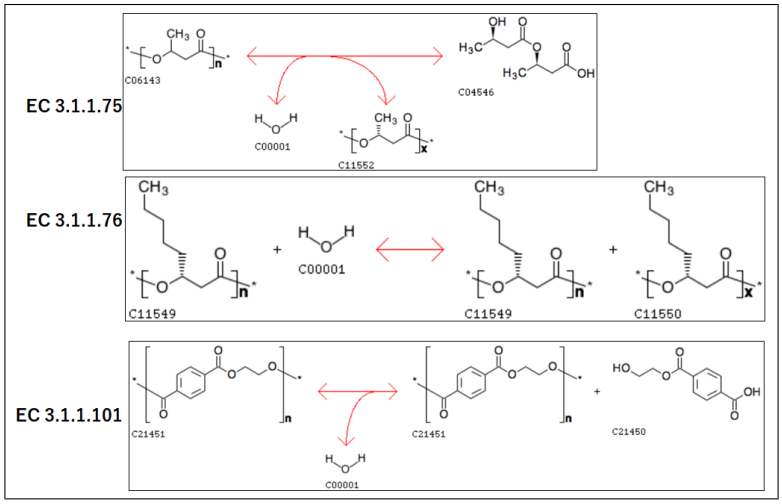


図 5.4: ターゲット 2 と同クラスタに属する反応式

ターゲットの近くに位置していた EC 番号について，実験に関する文献数が少なく，あまり知られていない酵素の場合，今後の検証実験等で，EC3.1.1.3 よりも優れた酵素であることが示される可能性があるため，これらについてさらなる分析が必要であると考えられる．

第6章

おわりに

近年，新型コロナウイルスになどの影響で，新薬開発の需要が高まり，化学反応の設計や予測を行う研究が発展を続けている．一方で，反応の効率化と環境面から，酵素の生体触媒を用いて合成が行われる機会が増えており，目的の反応に対して最適な酵素を予測することが重要視されている．しかし，基質特異性などの酵素の性質は生物分野にかかわるため，有機合成の知識のみでは解決が難しく，酵素研究の専門家協力する，または，酵素データベースを参照するなどして最適な酵素候補は探索されていた．

目的とする反応に対して，酵素候補を予測するシステムがあれば，次のステップである1つの酵素に絞るスクリーニングまでスムーズに進めることができる．また，酵素はEC番号と，生体内で自身が使用されて起こる代表的な反応の反応式で管理されている．

これらのことから，本研究では，ターゲットとなる反応式を与えた際に，EC番号の代表的な反応式と比較し，最も類似する反応式のEC番号を最適な酵素として予測する．予測の方法として，化合物の物理・化学的な特性値を計算し，反応物から生成物の差分を取った特性値変化量をもとに，反応式どうしの類似性を比較することを提案した．

KEGGやPubChemなどで必要とするデータを取得し，RDKitを用いて各反応に対して208種の特性値変化量を計算し，特徴ベクトルを作成した．その後，凝集型クラスタリングによって特徴ベクトルの次元を80次元まで削減し，SOMによって反応式のクラスタリングを行った．

2パターンでの検証を行い，1つ目では，ターゲットの反応式に対して，EC3.1.1.80，EC3.1.1.93 の反応式が，2つ目では，EC 3.1.1.75，EC 3.1.1.76，EC 3.1.1.101 が類似していると判定された．本来予測されるはずのEC 3.1.1.3を予測することはできなかったが，ターゲット反応式・EC反応式，または各反応式間で共通する特徴を確認することができた．

今後の課題として，化学反応時の構造変化を特徴としてより詳細に捉えるため，重要な記述子を残しつつ，さらに次元を削減していく手法を開発することが挙げられる．また，EC番号の反応式のクラス分類に着目し，最も精度よく分類できる記述子の組み合わせを特定したのち，ターゲットの反応式の酵素予測において検証していくことが考えられる．

謝辞

本研究を遂行するにあたり，多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学工学部電子・情報工学科情報基盤工学講座の奥原浩之教授，António Oliveira Nzinga René 講師に深甚な謝意を表します．そして，有機化学・酵素化学に関して貴重なご意見をいただいた工学部生物工学科酵素化学工学講座の浅野泰久教授，くすりのシリコンバレー TOYAMA 研究拠点化プロジェクトディレクター補佐の岩﨑源司博士 (薬学) に感謝申し上げます．最後になりましたが，多大な協力をしていただいた研究室の同輩諸氏に感謝致します．

2022 年 2 月

武藤 克弥

参考文献

[1] “ケモインフォマティクス市場、2021 年から 2026 年の間に CAGR13 %で成長見込み”, https://prtimes.jp/main/html/rd/p/000002048.000071640.html, 閲覧日 2022.1.6.

[2] Tamas Benkovics, John A. McIntosh, Steven M. Silverman, Jongrock Kong, Peter Maligres, Tetsuji Itoh, Hao Yang, Mark A. Huffman, Deeptak Verma, Weilan Pan, Hsing-I Ho, Jonathan Vroom, Anders Knight, Jessica Hurtak, William Morris, Neil A. Strotman, Grant Murphy, Kevin M. Maloney, and Patrick S. Fier1,, “Evolving to an Ideal Synthesis of Molnupiravir, an Investigational Treatment for COVID ‐ 19”, *ChemRxiv*, 2020.

[3] 北川勲，磯部稔, ”天然物化学・生物有機化学 I”. 朝倉書店, 2008. 3-4 ページ

[4] 西村淳, 樋口弘行, 大和武彦, ”有機合成化学入門 -基礎を理解して実践に備える” 丸善株式会社, 2010. 1 ページ

[5] “日本化学会・ケモインフォマティクス部会”, https://cicsj.csj.jp/, 閲覧日 2022.1.23.

[6] 中野裕太, 瀧川一学, “化学反応ネットワークにおける最適反応経路候補の列挙”, 情報処理学会研究報告, Vol. 122, No. 16, 2019.

[7] 佐藤寛子, “化学情報学 - 化学反応の系図と反応予測 -” 国立情報学研究所, 2003

[8] 藤波 美起登, 清野 淳司, “量子化学計算情報を記述子とした機械学習に基づく反応予測手法の開発”, *Journal of Computer Chemistry, Japan*, Vol. 15, No. 3, pp. 63-65, 2016.

[9] “酵素の化学”, http://www.sc.fukuoka-u.ac.jp/~bc1/Biochem/biochem5.htm, 閲覧日 2022.1.31.

[10] “酵素基質とは”, https://bizcomjapan.co.jp/iris-biotech/knowledge/substrate/, 閲覧日 2022.1.31.

[11] “新設された酵素分類 EC7 の和名提案について”, https://www.jbsoc.or.jp/notice/ec_translocase.html, 閲覧日 2022.1.15.

[12] 白兼孝雄, “酵素の分類と命名法”, JAS 情報, 2017

[13] “Enzyme Nomenclature”, https://iubmb.qmul.ac.uk/enzyme/, 閲覧日 2022.1.15.

[14] ”KEGG: Kyoto Encyclopedia of Genes and Genomes”, https://www.genome.jp/kegg/kegg_ja.html, 閲覧日 2022.1.17.

[15] ”CAS SciFinder^n” https://scifinder-n.cas.org/ 閲覧日 2022.1.23.

[16] "CAS SciFinder^n 検索ガイド" https://www.jaici.or.jp/scifinder-n/ref/sfn.pdf 閲覧日 2022.2.3.

[17] "PubChem", https://pubchem.ncbi.nlm.nih.gov/, 閲覧日 2022.1.17.

[18] "About PubChem", https://pubchemdocs.ncbi.nlm.nih.gov/about, 閲覧日 2022.2.6

[19] Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang, Stephen H. Bryant, "PubChem Substance and Compound databases", *Nucleic Acids Research*, Vol. 44, No. 1, pp. 1202-1213, 2016.

[20] "BRENDA The Comprehensive Enzyme Information System", https://www.brenda-enzymes.org/index.php, 閲覧日 2022.2.1

[21] "KEGG API", https://www.kegg.jp/kegg/rest/keggapi.html, 閲覧日 2022.2.1

[22] Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Stephen H. Bryant, "PUG-SOAP and PUG-REST: web services for programmatic access to chemical information in PubChem", *Nucleic Acids Research*, Vol. 43, No. 1, pp. 605-611, 2015.

[23] "SMILES 記法は化学構造の線形表記法" https://future-chem.com/smiles-smarts/, 閲覧日 2022.1.27

[24] Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse, "Reoptimization of MDL Keys for Use in Drug Discovery", *American Chemical Society*, Vol. 7, No. 12, 2012.

[25] "The RDKit Documentation", https://www.rdkit.org/docs/GettingStartedInPython.html#list-of-available-descriptors, 閲覧日 2022.2.6

[26] "PubChemPy documentation", https://pubchempy.readthedocs.io/en/latest/#, 閲覧日 2022.2.6

[27] Qian-Nam Hu, Hui Zhu, Xiaobing Li, Manman Zhang, Zhe Deng, Xiaoyan Yang, and Zixin Deng, "Assignment of EC Numbers to Enzymatic Reactions with Reaction Difference Fingerprints", *J. Chem. Inf. Comput. Sci.*, Vol. 42, No. 6, 2002.

[28] Yoshihiro Yamanishi, Masahiro Hattori, Masaaki Kotera, Susumu Goto, Minoru Kanehisa, "E-zyme: predicting potential EC numbers from the chemical transformation pattern of substrate-product pairs", *Bioinformatics.*, Vol. 25, pp. 179-186, 2009.

[29] "クラスタリング (クラスター分析)", https://www.kamishima.net/jp/clustering/#bib_cutting, 閲覧日 2022.2.3

[30] "クラスタリングとは — 概要・手順・活用事例を紹介", https://ledge.ai/clustering/, 閲覧日 2022.2.3

[31] “クラスタリング手法の列挙 (一部)”, https://qiita.com/sotoattanito/items/b885ef2dd3fe11cb817d, 閲覧日 2022.2.8

[32] Teuvo KOHONEN, ”Self-organized formation of topologically correct feature map”, *Biological Cybernetics*, Vol. 43, pp. 59–69, 1982.

[33] 亀岡瑶, 宗像昌平, 八木圭太, 山本儀郎, “自己組織化マップによる顧客の分類とその可視化”, 計算機統計学, Vol. 29, No. 2, pp. 181-188, 2016.

[34] 福嶋 瑞希,”環境認識ライフログからの行動パターン解析による類似性・イベント検出”, 富山県立大学学位論文 2018.

[35] ”KH Coder” "http://khcoder.net/" 閲覧日 2022.1.30

[36] Mark A. Johnson, Gerald M. Maggiora, “Concepts and Applications of Molecular Similarity”, *Wiley*, New York, 1990.

[37] “[Python コード付き] 相関係数で変数選択したり変数のクラスタリングをしたりしてみましょう”, https://datachemeng.com/variable_selection_and_clustering_based_on_r/, 閲覧日 2022.1.29

[38] “sklearn.cluster.AgglomerativeClustering ”, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html, 閲覧日 2022.2.3

[39] “Package ‘som’ ”, https://cran.r-project.org/web/packages/som/som.pdf, 閲覧日 2022.2.3

[40] ”KEGG API を用いてデータ取得”, https://rstudio-pubs-static.s3.amazonaws.com/472676_97a2c135b5704dc1b52f7759b73466e8.html#kegg-compound, 閲覧日 2022.12.28.

[41] “Supporting Information for: Evolving to an Ideal Synthesis of Molnupiravir, an Investigational Treatment for COVID - 19”, https://europepmc.org/api/fulltextRepo?pprId=PPR257265&type=FILE&fileName=EMS109513-supplement-Supporting_Information.pdf&mimeType=application/pdf, 閲覧日 2022.2.6