修　士　論　文

ゲノムデータベースから外延的オントロジーを自動生成する手法に関する研究

指導教官　　佐藤 賢二　助教授

北陸先端科学技術大学院大学
知識科学研究科知識システム基礎学専攻

050087　柳生 拓也

審査委員：　佐藤　賢二　助教授（主査）
小長谷　明彦　教授
中森　義輝　教授
本多　卓也　教授

2002 年 2 月

目　次

図　目　次

表　目　次

第　1　章
背景と目的

1.1　研究の背景と目的

近年の研究の成果により様々な生物のゲノム解析が進み、生物ごとにデータベースが整備されてきた。また、遺伝子のデータベースや酵素に関するデータベースなど、様々な機能ごとのゲノムデータベースも出来上がっていった。今後は別々に発展してきたゲノムデータベースを再統合し、ゲノムの世界全体を再構築することで新たな知識の発見を目指す研究が行われていくと考えられる。そのためにはデータベース間で共通の語彙が必要であり、統合のために機能を定義するオントロジーが必要であると言われている。

オントロジーとは、言葉の意味を明示的に記述し、言葉の間の関係を階層的に表現したものである。暗黙の内に共有している概念を定義することで、言葉がもつ多義性を排除するのが大きな目的である。生命科学の分野でもオントロジーの研究が進められた結果、TaO[1]、Gene Ontology[2]、EcoCyc[3]といったオントロジーやGKB-Editor[4]といったオントロジー構築ツールが開発された。しかし、これらは概念の内包的意味についてのオントロジーであるといえる。概念の内包的意味とは、その概念が持つべき条件ということもできる。このような概念の内包的意味について厳密に定義し、矛盾が無いように概念同士を関係付けて行くという作業は大変な労力を要する。また、専門家が知識を総動員し、合意を得ながら作成する作業には大変な時間がかかる。

一方、言葉の意味はその言葉が存在している文章や文書の集合体であると言うこともできる。これは、前述の内包的な定義とは対照的な、外延的・集合論的な定義である。このような外延的な言葉の意味に関するオントロジーであれば、専門知識が無くとも、ゲノムデータベース中から言葉を切り出し、グループ化・階層化することで自動的に作成することができる可能性がある。また、この方法で構築された

外延的オントロジーには、膨大な数の専門用語を擁し、データベースの更新にあわせて容易に更新できるなどの利点があると考えられる。

以上の理由から、本研究ではゲノムデータベースを題材に、外延的オントロジーの自動構築に関する研究を行う。

1.2 本論文の構成

本論文は本章を含め 7 章から構成される。第 2 章ではオントロジーについて説明した後、生命科学の分野で行われたオントロジー研究について述べる。第 3 章では、関連研究について述べた後、本研究における外延的オントロジーの作成方法、ならびに作成されたオントロジーの特徴について述べる。第 4 章では、作成した外延的オントロジーの出現情報を元に行った、専門用語のクラスタリングについて述べる。第 5 章では、作成したオントロジーとクラスタリングの結果を利用するために作成した CGI について述べる。第 6 章では、ステミング技術、および切り出した用語に適用した結果について述べる。第 7 章では、本論文で述べた研究についてまとめ、今後の課題を述べる。

第　2　章
オントロジー

本章ではオントロジーとはどのようなものであるかを説明した後、生命科学の分野で行われたオントロジー研究について述べる。

2.1　オントロジーの定義

オントロジーは哲学や人工知能、自然言語処理といった様々な分野で研究が進められている。オントロジーの定義はいまだ研究者によって議論が続いており全員が合意する定義はない[5]と言ってよいが、情報科学においては「対象とする領域における概念の属性と包含関係について首尾一貫した定義を与え、それを人間と計算機の両方が理解できる形式で記述したもの」と定義されている[6]。

以下では、オントロジーがどのように構成され、どのような特徴を持ち、どのような機能を持つかについて述べる。

2.1.1　オントロジーの構成

オントロジーは以下のような手順で記述される[5]。

(1) 概念の切り出し
(2) 概念分類、階層的表示
(3) 概念間の関係記述
(4) 形式的定義

まず、対象とする世界に存在する概念を集める。集められた概念は様々な関係に基づいて階層的に組織化される。多くのオントロジーでは、このような概念と概念の関

係が述語論理などを用いて、公理の形で形式的に記述されている。

例として、図 2.1 の世界に関するオントロジーを作成してみる。この世界はテーブルの上にいくつかの積み木が載っている世界で、積み木は移動できるがテーブルは移動できないとする。以下この世界のことを積み木の世界と呼ぶ。

まずは積み木の世界を記述するために必要な概念を選ぶ。この例の場合は A、B、C、TABLE の 4 つである。

次に概念の分類を行う。移動できる物体と移動できない物体というものに分類した場合、図 2.2 のように分類され階層的に表示される。積み木やテーブル、物体という

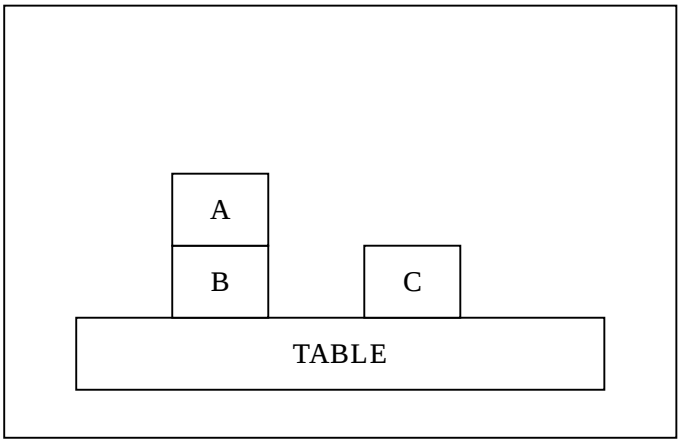


図 2.1　積み木の世界

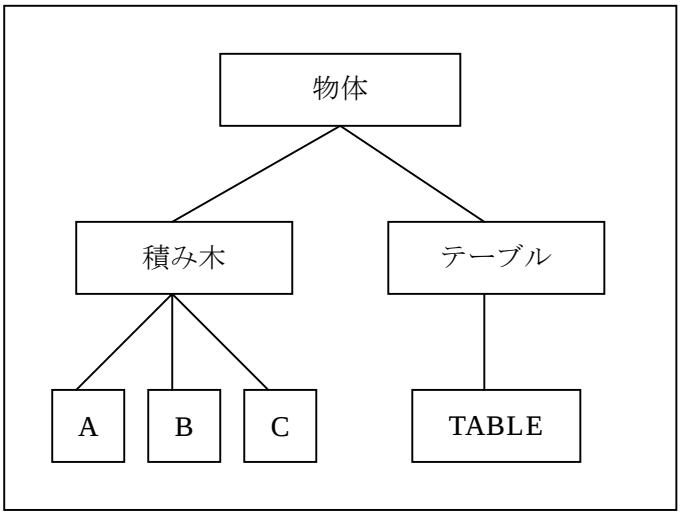


図 2.2　概念の分類

概念もこの世界を記述するために必要なものである。

次に概念間の関係を記述する。上下関係に注目した場合、“A は B の上にある。”、“B は TABLE の上にある”、“C は TABLE の上にある”という関係が記述できる。また、“A の上には何も無い”、“C の上には何も無い”という記述もできるであろう。“上にある”や“何も無い”も積み木の世界を記述するために必要な概念である。

そして、物体 x が物体 y の上にある状態を ON(x,y)、物体 x の上に何も無い状態を CLEAR(x)と形式的に表現した場合、概念の間の関係は ON(A,B)、ON(B,TABLE)、ON(C,TABLE)、CLEAR(A)、CLEAR(B)と記述できる。

これまでに記述してきた概念全てが積み木の世界を記述するためのオントロジーである。このオントロジーを使うことで、積み木の世界を以下のように形式的に表現することができる。

$$ON(A,B) \wedge ON(B,TABLE) \wedge ON(C,TABLE) \wedge CLEAR(A) \wedge CLEAR(B)$$

2.1.2 オントロジーが持つ性質

オントロジーがもつ性質としては以下のようなものがあるといわれている[5]。

(1) 目的依存性
(2) 一般性
(3) 共通性・合意
(4) 安定性
(5) 形式性（コンピュータ理解可能性）
(6) 部分性
(7) 一貫性
(8) 明示性
(9) 部品性

オントロジーをどのような目的で利用するかによって、出来上がるオントロジーは変わってくる。例えば、自然言語処理に携わるものにとっては“語彙”や“辞書”といった観点が重要視され、知識ベース研究者にとっては“実行可能性”や知識ベース構築の“部品”という考えに基づいたオントロジーが作成される。さらに、対象とす

る世界が変われば、オントロジーを構成する概念も変わってくる。こういったことから、オントロジーは目的依存性を持つと言われる。

また、オントロジーは一般性を持っていなければならない。万人が興味を持つものしかオントロジー構築の対象にしてはいけないというのは強すぎる制限であるが、世界に一つしかない対象、あるいは一人の人しか興味を持たない領域などに関するオントロジーは意味が無いことは明らかであろう。そして、そのオントロジーは多くの人の合意を得ているべきである。

2.1.3 オントロジーの機能

オントロジーはその利用目的に合致した様々な機能を持っている。具体的には以下のような機能があるといわれている[5]。

(1) 語彙の提供
(2) メタモデル
(3) 暗黙情報を明示化
(4) 概念定義
(5) データ構造
(6) 知識の体系化
(7) 標準化
(8) 設計意図
(9) 内容の理論

自然言語処理を目的とする人はオントロジーに語彙の提供という機能を期待するだろう。オントロジーは対象とする世界を記述する際には厳密に定義された標準的な語彙を提供する。

また、計算機上でのモデル化を行う目的でオントロジーを扱いたい者は、オントロジーがもつメタモデル的機能に期待するだろう。オントロジーは概念とそれらの間に成立する関係を明示的に規定したものであり、これらの概念と制約の元で構築されたモデルによって様々な問題解決を試みることができる。

2.2 生命科学におけるオントロジー

以下、生命科学の分野で行われたオントロジーに関する研究について幾つか紹介する。

2.2.1 TaO と MBO

TaO と MBO（Molecular Biology Ontology）は統合データベースを目的として最初に開発されたオントロジーである。これらのオントロジーは、生命科学全体という広い領域を対象としているので、必然的に抽象的な概念のみが記述されているという特徴がある。TaO や MBO は、生命科学の普遍的な概念について、それらの関係の種類と構造を詳細に考察し、生命科学の概念の性質を明らかにした[7]。

図 2.3 は TaO のオントロジー空間をブラウズするためのアプリケーションのスクリーンショットである。四角で囲まれた文字が概念を示している。注目する概念（この例では protein）の周りに関係のある概念が配置され、注目する概念に対する関係がその上に表示されている。概念に[+]の印がついているものは、さらに別の関連がある概念があることを表す印である。

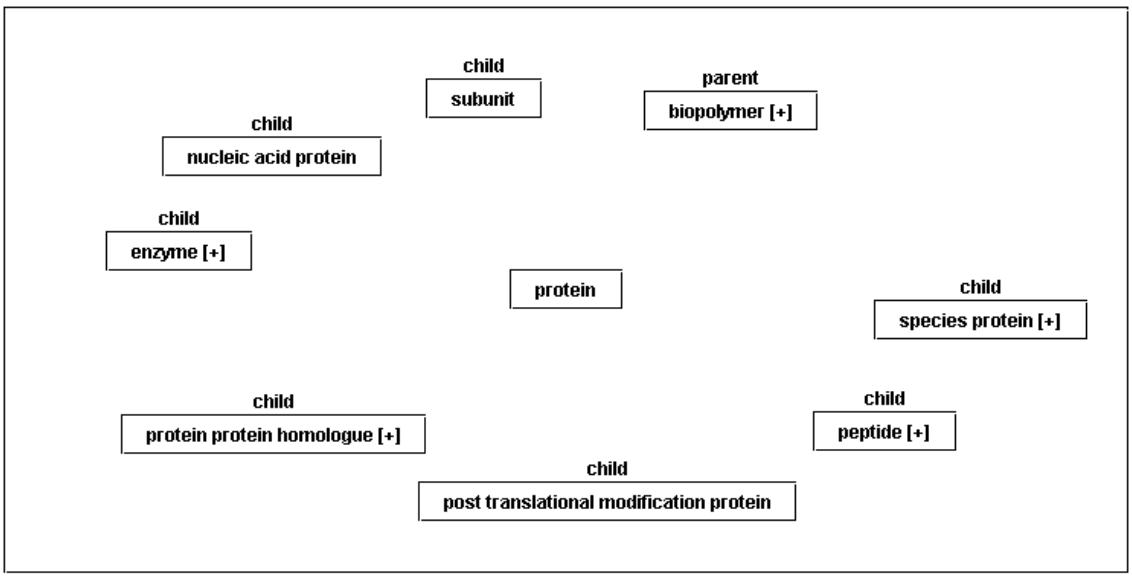


図 2.3　TaO における概念と関係をグラフで図示した例

2.2.2 Gene Ontology

実際に比較ゲノム生物学に利用できる大規模なオントロジーをつくろうとする研究は Gene Ontology 共同体によって始められた。Gene Ontology は TaO や MBO と異なり、分子の詳細な機能を対象としている。

Gene Ontology はショウジョウバエ、出芽酵母、マウスのデータベースの共同プロジェクトとして始まり、現在はシロイズナズナ、線虫、分裂酵母が加わり、これらのモデル生物における遺伝子の共通な機能を定義づけている。ショウジョウバエ、線虫、出芽酵母のゲノム比較解析から、真核生物においては、遺伝子の配列と機能が生物種間で高く保存されていることを明らかにするといった成果をあげている[7]。

図 2.4 は Gene Ontology Browser のスクリーンショットである。Gene Ontology Browser は Gene Ontology の空間をブラウズするためのアプリケーションであり、この例では enzyme（酵素）という概念に注目している。図上部の Definition には enzyme の定義が記述されており、下部にはオントロジーの階層構造が表示されている。

2.2.3 Interaction Ontology

分子の機能を分子相互作用の側面から定義したオントロジーとして Interaction Ontology[8]がある。Gene Ontology が機能全般を対象としているのに対し、Interaction Ontology は複数分子の間の相互作用による機能に対象を限定している。Interaction Ontology は特に概念(機能)の属性について詳細な考察を行っているが、概念の階層化は行っていない[7]。

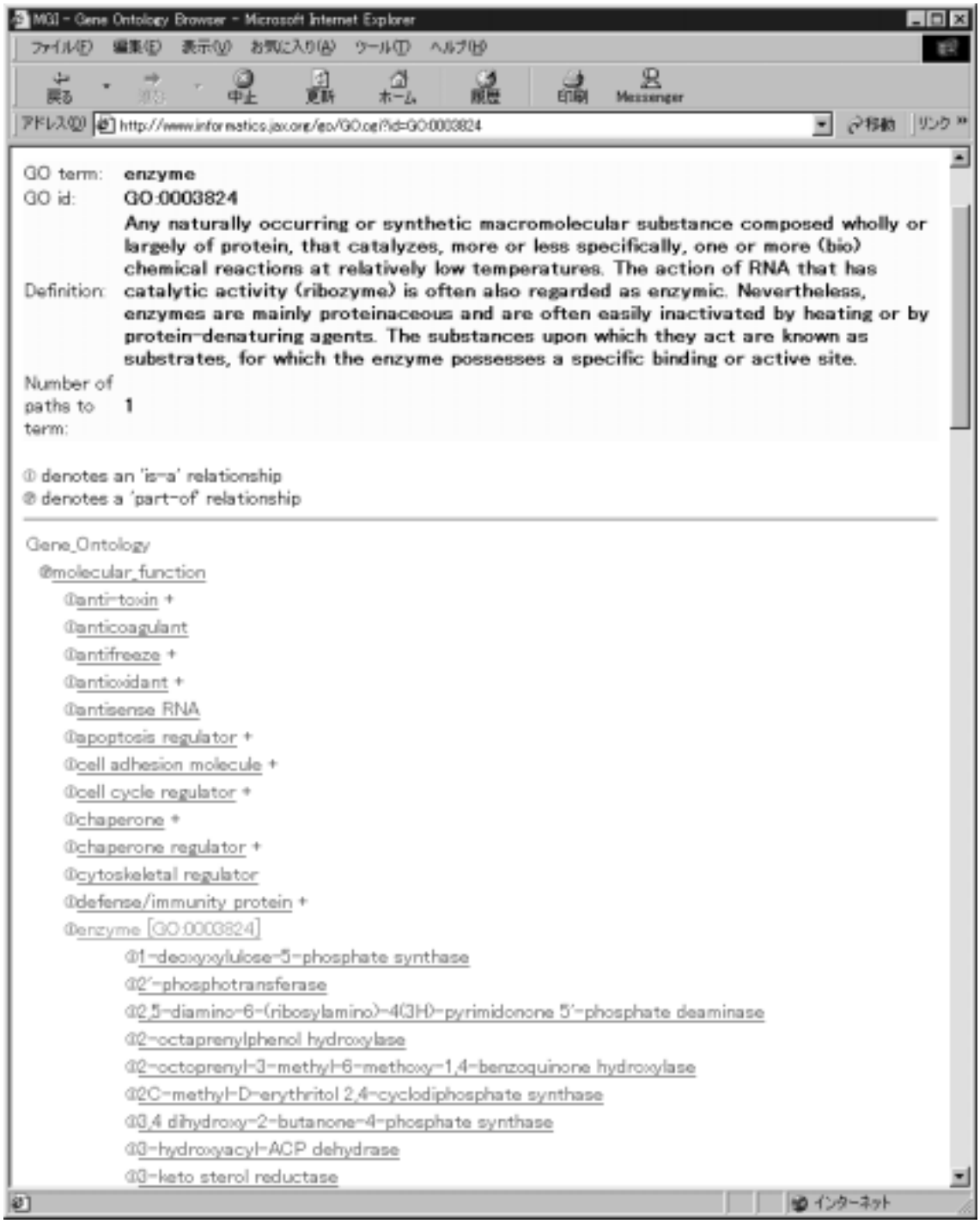


図 2.4　Gene Ontology Browser のスクリーンショット

第　3　章
外延的オントロジーの作成

本章では、関連研究について紹介した後、本研究における外延的オントロジーの作成方法や特徴について述べる。

3.1　関連研究

大阪大学細胞生体工学センターの大久保らがおこなった BOB プロジェクトは、専門用語の出現情報を元に概念関係の構造化を行うという点で本研究と類似している。BOB プロジェクトは、医学で使われている教科書の索引に載っている用語を医学オブジェクトとし、教科書内の同じページで共起する回数をオブジェクト関係として測定することで医学知識を表現しようというものである。同一ページには同じ内容に関することが書かれているという仮定に基づき、オブジェクトの共起関係をページごとに調べていくことで、用語の関係を定量的に扱うことを目的としている。そして、多くのページに出現しているオブジェクトや、共起しているオブジェクトが多いオブジェクトはより抽象的、あるいは重要な意味をもつと考えることができるとしている[9]。

関連研究として BOB プロジェクトを紹介したが、用語の出現情報を元に知識の構造化を行うという試みは生命科学では他に例がないようである。

3.2　外延的オントロジーの作成方法

本研究では、専門用語がどこにどのようにして存在しているかという情報を元に専門用語間の関係付けを行うことでオントロジーを作成するというアプローチをとる。専門知識を人間がモノやコトに分類し階層的に厳密に分類する手間を省き、分類者の個性による影響や、絶え間ない更新の努力から解放することを目的としている。

外延的オントロジーの作成過程は大きく次の三つに分けることができる。

・専門用語の収集
・用語の正規化
・概念階層の構築

以下では、これらについて順番に述べていく。

3.2.1 ゲノムデータベースからの用語の切り出し

オントロジーを作成するためには、まずはオントロジーを構成する概念（用語）を準備する必要がある。そこで、ゲノム分野の知識の集合ともいえるゲノムデータベース中から専門用語を収集することを考えた。学内にゲノムネット[10]のミラーサーバーがあったので、ゲノムネットで利用できるデータベースから、専門用語の切り出しを行うことにした。ゲノムネットは複数のデータベースをネットワークで繋ぐことで多種多様なデータベースを統合的に扱うためのサービスを提供している。図 3.1 はゲノムネットで利用可能なデータベースと、これらのデータベースが DBGET/LinkDB 統合データベースシステムによってネットワーク化されている様子を示すものである。

ゲノムデータベースはエントリーと呼ばれる単位の情報がたくさん集まって出来ている（図 3.2）。エントリーはいくつかのフィールドに分類されており、フィールドにはそれぞれ記述すべき内容や記述するための書式が決められている。

表 3.1 は酵素に関するデータベースである enzyme[1] 中のエントリーを一部抜粋したものである。enzyme の場合、NAME フィールドにはこのエントリーで情報を扱う酵素の名前が改行区切りで記述される。このフィールドから一行づつ切り出しを行えば、酵素の名前が大量に収集できそうである。また、REACTION フィールドには化学式が、COMMENT フィールドにはコメント文が記述されるが、どちらも用語とはいえないであろう。このように各データベースの記述を見ていくことで、専門用語を抽出しやすいフィールドを選択した。表 3.2 と表 3.3 は、実際に用語を切り出す対象として選んだデータベースとフィールドの一覧である。これらのデータベース毎に専門用語を切り出すためのプログラムを作成し用語の収集を行った。各エントリー中のフ

[1] 正式なデータベース名は ENZYME であるが、ゲノムネットでは内部処理の関係でどのデータベースも全て小文字で表記されているため、本論文ではそれに習い全てのデータベース名を小文字で表記するものとする。

ィールドから切り出された用語の数を全エントリー分合計したものを、フィールドから用語が切り出された回数と表現する（図 3.3）。また用語を切り出して収集する際に、切り出した用語が出現しているデータベースとエントリーについての情報も同時に収集した。このエントリー情報から用語の外延的意味を知ることができる。収集した用語と出現エントリーの情報は表 3.4 のようにタブ区切りの形で保存することとした。この段階で 1,911,950 種の用語を集めることが出来た。

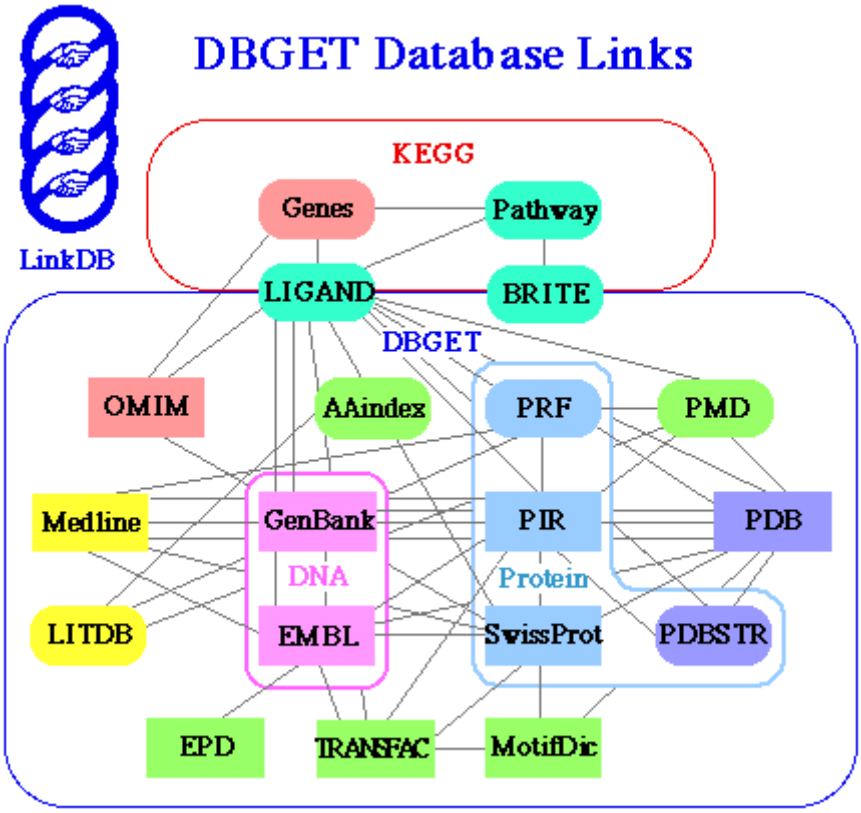


図 3.1　ゲノムネットで利用できるデータベースとネットワーク

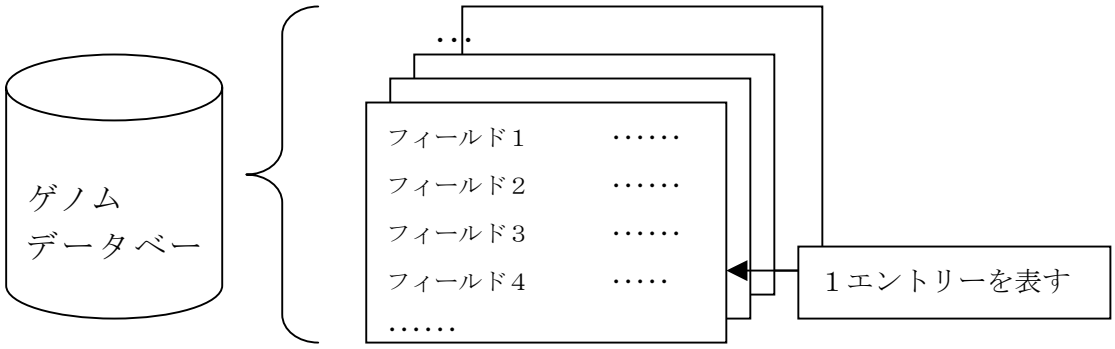


図 3.2　ゲノムデータベース概略図

ENTRY	EC 2.7.1.37
NAME	Protein kinase
	Phospharylase b kinase kinase
	Glycogen synthase a kinase
	Hydroxyalkyl-protein kinase
	Serine(threonine) protein kinase
CLASS	Transferases
	Transferring phosphorus-containing groups
	Phosphotransferases with an alcohol group as acceptor
SYSNAME	ATP:protein phosphotransferase
REACTION	ATP + a Protein = ADP + a Phosphoprotein
SUBSTRATE	ATP
	Protein
PRODUCT	ADP
	Phosphoprotein
INHIBITOR	Debromoaplysiatoxin
	Bryostatins
	Teleocidin B-1
	12-O-Tetradecanoylphorbol 13-acetate
COMMENT	A group of enzymes which are under review by NC-IUBMB.
	Other present entries are EC 2.7.1.38, 2.7.1.70, 2.7.1.99,
	2.7.1.109-112, 2.7.1.115-6, 2.7.1.123-6, 2.7.1.135 and 2.7.1.141.
	Some enzymes are activated by cyclic GMP, but not by cyclic AMP,
	and some enzymes by neither.
PATHWAY	PATH: MAP04320　Dorso-Ventral axis formation
GENES	BSU: BG13391(prkC)
	BHA: BH2504
	SAU: SA1063
…（以下略）…	

用語が切り出しやすそうなフィールド

表 3.1　ゲノムデータベースの記述例（enzyme:EC 2.7.1.37）

データベース名	フィールド名	用語が切り出された回数	抽出された用語の種類
brite	FUNCTION	154	41
	MNEMONIC	215	30
	NAME	608	182
	ORGANISM	204	7
compound	NAME	10,093	10,067
enzyme	COFACTOR	808	94
	EFFECTOR	46	35
	INHIBITOR	210	169
	NAME	6,553	6,146
	PRODUCT	7,835	2,878
	SUBSTRATE	8,623	3,103
epd	KW	5,494	763
	OS	2,580	216
genbank	bound_moiety	5,540	1,825
	cell_line	170,867	9,366
	cell_type	709,129	4,426
	clone_lib	10,346,792	24,395
	dev_stage	4,326,504	5,835
	function	43,031	18,564
	gene	62,172	264,117
	lab_host	6,515,235	896
	organelle	93,256	8
	organism	11,547,050	86,992
	phenotype	473	385
	plasmid	16,248	2,039
	product	800,175	215,133
	rpt_family	167,815	5,106
	sex	3,978,791	198
	specidic_host	23,442	4,552
	standard_name	25,208	16,007
	strain	3,689,963	80,204
	sub_clone	1,513	1,233
	sub_species	22,784	3,257
	tissue_type	4,753,884	6,303
	transposon	4,283	1,485
	variety	20,305	1,157

表 3.2　切り出し対象データベースとフィールド一覧(1)

データベース名	フィールド名	用語が切り出された回数	抽出された用語の種類
genome	DEFINITION	53	53
	LINEAGE	275	115
	MORPHOLOGY	114	20
	NAME	53	53
	PHYSIOLOGY	133	64
litdb	KEYWORD	2,783,594	1,018,313
pdb	KEYWDS	41,432	10,982
pir	KEYWORDS	206,088	1,067
pmd	EXPRESSION-SY	8,755	1,244
	STEM	38,529	18,297
	PROTEIN	24,417	2,677
prf	SOURCE	147,954	82,491
refseq	NAME	106	89
	bound_moiety	3,055	1,199
refseq	cell_line	3,381	911
	cell_type	5,978	1,535
	clone_lib	4,485	618
	dev_stage	2,415	1,842
	function	61,710	50,591
	gene	469	37
	lab_host	224	6
	organelle	24,176	927
	organism	22	22
	phenotype	246	240
	plasmid	56,176	41,101
	product	319	182
	rpt_family	1,851	18
	sex	97	75
	specidic_host	466	445
	standard_name	4,579	583
	strain	27	26
	sub_clone	313	40
	sub_species	9,165	995
	tissue_type	10	10
	transposon	7	6
swissprot	variety	290,205	836
	KW	716,875	4,482
	OC	141,000	9,097
transfac	OS	13,331	2,431
	DE	84,140	243
	OC	21,791	319
	OS		

表 3.3　切り出し対象データベースとフィールド一覧(2)

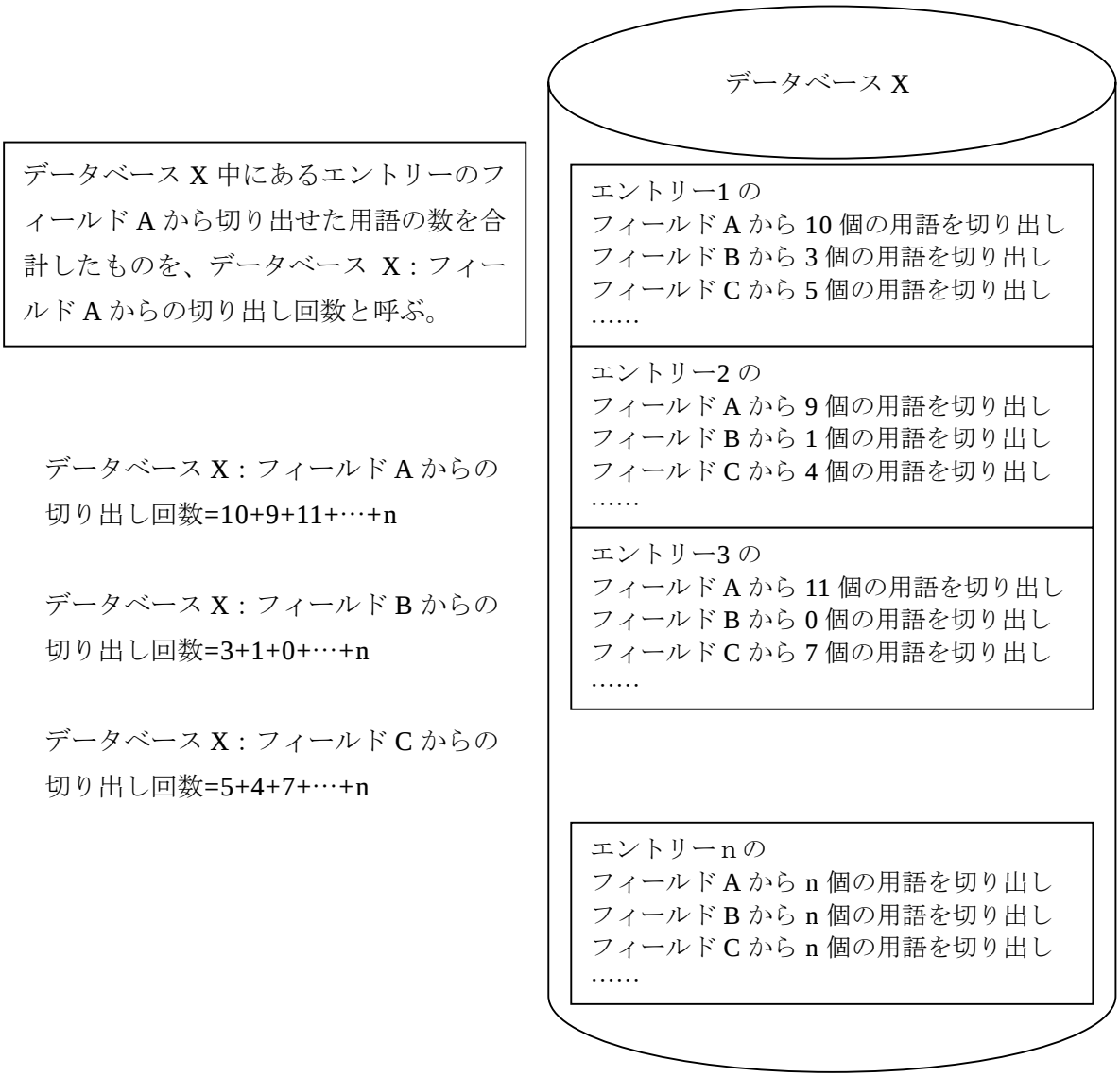


図 3.3　切り出し回数の説明

網掛けされている部分は切り出された用語。データベース名：フィールド名：エントリーID の形で記述されているものが出現個所の情報

表 3.4　切り出された用語と出現個所の情報を格納したデータ

3.2.2　用語の正規化

この段階で集められた用語の中には、本来は同じ物を指しているが、表記のされかたが違っているために別の概念として扱われている用語も含まれている。例えば以下のような場合が考えられる。

(1) 大文字小文字の違い
(2) 特殊記号の使われ方による違い
(3) 語順による違い

(1)の例としては、Superoxide と superoxide の違いが挙げられる。頭文字が大文字か小文字かの違いだけで、人間がみれば明らかに同じ物を指すのではないかと察しがつくが、計算機はこの二つを別の概念を表す用語であると判断してしまう。

(2)の例としては、*beta 1,3 glucanase と beta 1,3-glucanase の違いが挙げられる。データベースに情報を入力する人により特殊記号の使い方が違うために、このような違いが発生していると考えられる。

(4)の例としては、

Cu/Zn Superoxide dismutase
Superoxide dismutase (Cu-Zn)
*Superoxide Dismutase Cu/Zn

といった違いが挙げられる。特殊記号の使い方の違いに加え、語順の違いによって、やはり別の用語であると判断されてしまうことになる。

これらの問題を解消するために、収集した用語に対して以下の処理を行った。これらの処理をまとめて用語の正規化と呼ぶことにする。

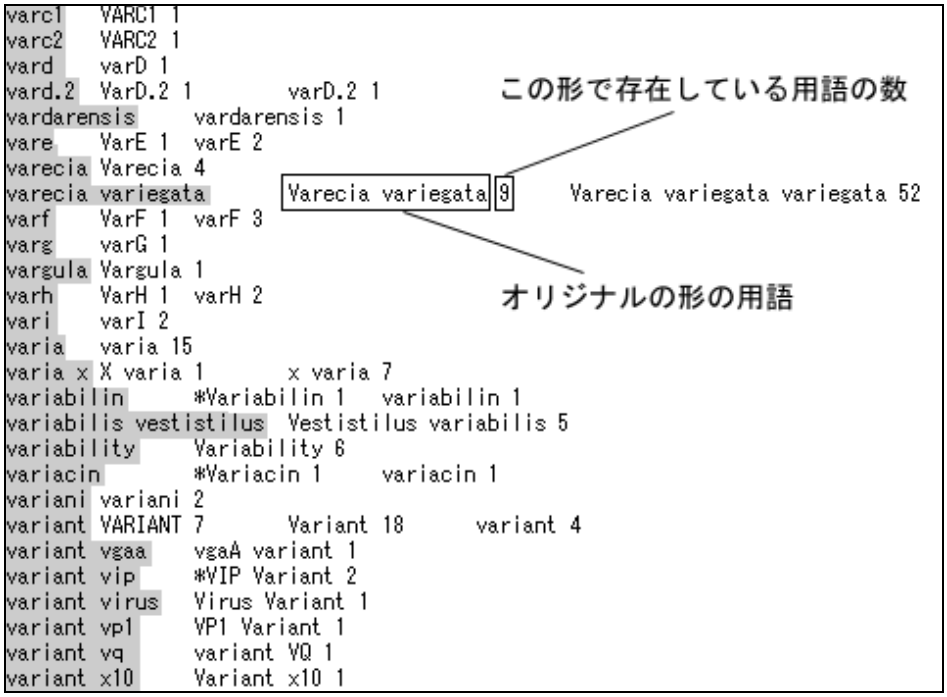
①大文字を全て小文字に変換。
②特殊記号の除去
特殊記号（!"#$%&'()*+,-/:;<=>?@[¥]^_`{|}~の 31 種）を全て空白に変換後、二つ以上連続している空白を一つの空白に置き換える。
③用語を構成している単語をソート
重複している単語はマージする。用語を単語の集合であると考えることを意味する。

単純に切り出した状態では 1,911,950 個だった用語が、①の処理により 1,850,933 個、②までの処理により 1,770,562 個、③までの処理を行うことにより 1,745,877 個にまでまとめることができた。①までの処理、②までの処理、③までの処理の結果をそれぞれ表 3.5 のようなタブ区切りの形式で保存している[2]。

以上により、オントロジーを構成する用語とその出現個所についての情報を収集することができた。

---

[2] ③の処理によって用語の意味が大きく変わってしまうという問題があるため、次に説明する親子関係や次章で説明するクラスタリングは、②までの処理を行った用語を対象に行っている。



網掛けの部分が正規化後の用語。その後に切り出されたままの形の用語と切り出された回数がタブ区切りで記述されている。

表 3.5　正規化後の用語と元の形の用語を格納したデータ

3.2.3 親子関係の記述

用語とその外延的意味の収集の次に、各用語の間にある関係について記述していく必要がある。本研究では、単語の包含関係を親子関係と呼び、これらを概念間の関係として記述することとした。例えば、apple と green apple では green apple の方がより狭い概念であろうと考えられる。このように単語の包含関係を見ていくことで、単純に概念の大きさの関係を比べることができ、計算機を使って関係を網羅することが簡単にできると考えた。

親子関係の定義は以下のとおりである。

ある用語 t_1 を構成する単語 $w_1, w_2 \ldots w_n$ 全てが、ある用語 t_2 に含まれている時、t_1 は t_2 の親であるとする。逆に t_2 は t_1 の子供であるとする。

図 3.4 の例では dna binding protein という用語には、binding protein を構成する

単語である、binding および protein を全て含んでいる。よって binding protein は dna binding protein の親であり、dna binding protein は binding protein の子供というということになる。

この様に集合の包含関係を用いて用語の間に親子関係をつけていく場合、ある用語の親集合の中でさらに親子関係が成立することが考えられる。dna binding protein は、binding、protein、binding protein の三つの親を持つが、binding protein はさらに binding と protein を親に持っている。そこで、直接の親と、間接の親という関係を新たに作ることとした。直接の親と、間接の親の定義は以下のとおりである。

*ある用語 **t** の親集合 **P** の中で、さらに親子関係があったとき、**P** の中で親となる用語は **t** の間接の親であるとする。親集合 **P** から間接の親を除き、残ったものを **t** の直接の親とする。*

これにより dna binding protein は三つの親を持っているが、直接の親は binding protein だけという事になる。そして、ある用語の直接の親と間接の親をまとめてその用語の先祖と呼ぶことにし、直接の子供と間接の子供をまとめて子孫と呼ぶこととした。ある用語の先祖（子孫）はその用語と関連がある用語であるが、その中でも直接の親（子供）と特に関連が深いといえるだろう。

以上の手順で用語の親子関係を調べ、結果を表 3.6 と表 3.7 のようなタブ区切りの形で保存した。

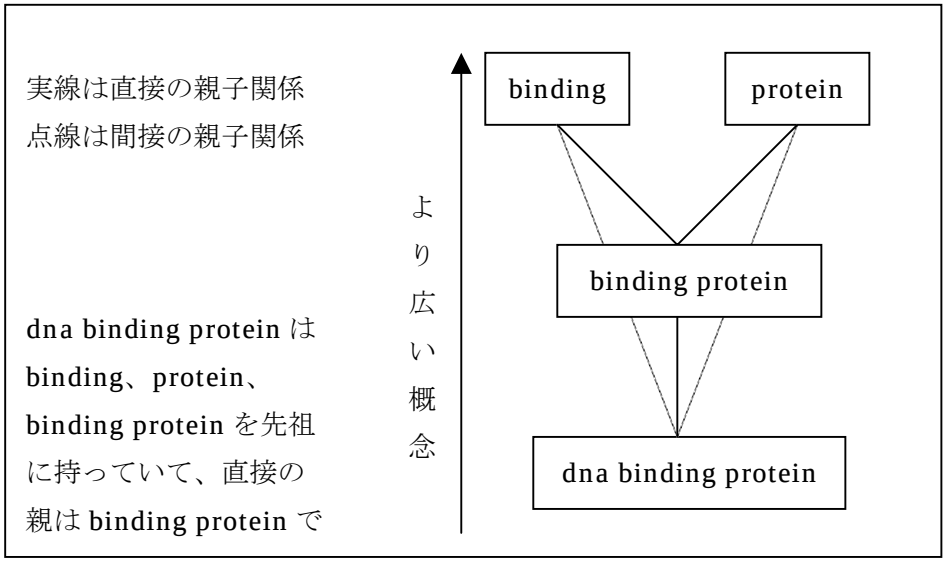


図 3.4　親子関係図

zyxin binding protein	binding	binding protein	protein	zyxin			
zyxin related lim protein		lim	lim protein	protein	zyxin	zyxin relate	
d protein							
zyxin related protein	protein	zyxin					
zyxin related protein 1	1	protein	protein 1		zyxin	zyxin related protei	
n							
zyxin related protein isotype trip6		isotype	protein	trip6	zyxin	zyxin relate	
d protein	zyxin related protein trip6						
zyxin related protein isotype zrp 1		1	isotype	protein	protein 1		zrp
zrp 1	zyxin	zyxin related protein	zyxin related protein 1				
zyxin related protein trip6		protein	trip6	zyxin	zyxin related protein		
zz beta gal igg binding fusion protein		beta	beta gal		beta protein		binc
ing	binding protein	fusion	fusion protein	gal	gal protein	igg	igg
binding protein	protein	zz					
zz domain protein		domain	protein	zz			
zz igf 1	1	igf	igf 1	zz			
zz igf 57 70	57	70	igf	zz			
zz lacz lacz	zz						
zz sequence	sequence		zz				
zzv 1050	1050						
zzz4 gene	gene						

下線が引かれた用語の後に、先祖である用語がタブ区切りで記述されている。

表 3.6　用語と先祖の関係を格納したデータ

zyxin binding protein	binding protein	zyxin			
zyxin related lim protein		lim protein	zyxin related protein		
zyxin related protein	protein	zyxin			
zyxin related protein 1	protein 1		zyxin related protein		
zyxin related protein isotype trip6		isotype	zyxin related protein trip6		
zyxin related protein isotype zrp 1		isotype	zrp 1	zyxin related protein 1	
zyxin related protein trip6	trip6	zyxin related protein			
zz beta gal igg binding fusion protein	beta gal		beta protein	fusion prote	
in	gal protein	igg binding protein	zz		
zz domain protein	domain	protein	zz		
zz igf 1	igf 1	zz			
zz igf 57 70	57	70	igf	zz	
zz lacz lacz	zz				
zz sequence	sequence		zz		
zzv 1050	1050				
zzz4 gene	gene				

下線が引かれた用語の後に、直接の親である用語がタブ区切りで記述されている。

表 3.7　用語と直接の親の関係を格納したデータ

3.3　本研究で作成したオントロジーの特徴

本研究では、専門知識をできるかぎり用いずに大量の専門用語からなるオントロジーを作成するのが目的であるため、2 章で挙げた一般的なオントロジーとは異なる性質や機能を持っている。それらについてまとめてみたいと思う。

(1)概念の記述

内包的なオントロジーは、用語が指す概念が備えていなければならない特徴や機能、条件に着目して作成される。本研究で作成したオントロジーでは、用語がどんな文章に現われ、どのように使われているかという外延をエントリー情報という形で記述している。

また、本研究では用語を正規化された単語集合に変換し、集合の包含関係から親子関係を作っているが、これは用語の内包的意味を考慮した関係付けではなく、この点でも内包的オントロジーと違っている。

(2)共通性・合意

本研究で作成したオントロジーは、実際に存在し多くの研究者によって利用されているゲノムデータベースからの情報を元に作ったものであり、ゲノムデータベースと同じレベルでの共通性や合意を得られるものであると考える。

(3)語彙の提供

本研究で作成したオントロジーを構成する用語は全て、生物分野の専門知識の集まりであるゲノムデータベース中に存在している。これらの用語は生物分野の専門用語としての合意を得ているものであり、専門用語辞書としての役割を十分に果たすと考えられる。

第 4 章
クラスタリングによる用語の分類

ゲノムデータベース中のエントリーは、様々なフィールド名とそれぞれに対応した内容の記述が集まって構成されている。よって、フィールドの名前からそれに対応する記述の特徴をある程度予想することができる。例えば、PROTEIN というフィールドには何かしらのタンパク質が記述されているはずである。逆に、PROTEIN というフィールドに多く存在している用語はタンパク質と関係が深いはずである。つまり、ある用語がある場所に出現することにはそれなりの意味がある。

このように考えると、用語がどのフィールドにどの程度出現しているかという外延に着目することは、その用語の意味を考えるには重要なことであり、外延が似ている用語の集合には、厳密ではなくとも、なにかしら共通性があるのではないかと考えられる。そこで、用語の出現個所と出現頻度からベクトルを作成し、クラスタリングを行ってみた。

本章では、本研究で用いた LBG アルゴリズムによるクラスタリングについて簡単に説明した後、実際に行った用語のクラスタリング結果について述べる。

4.1 LBG アルゴリズム

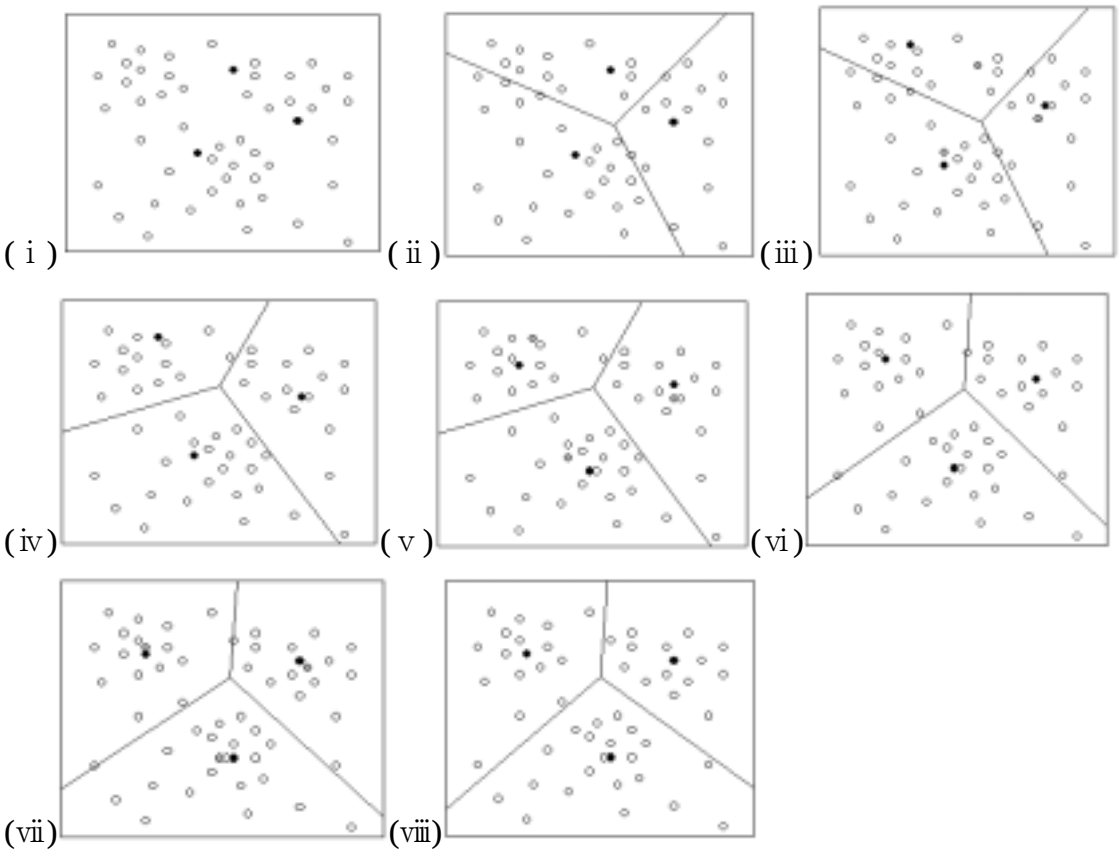
本研究で収集した用語の数は約 200 万と大変大きいので、計算複雑度が高いアルゴリズムを用いて厳密なクラスタリングを行うのは難しい。そのため、単純で比較的計算時間を抑えることができる LBG アルゴリズム[11]によってクラスタリングを行った。

LBG アルゴリズムは Linde、Buzo、Gray の三人が考案したクラスタリング手法であり、Lloyd algorithm とも呼ばれている。具体的なアルゴリズムは以下のとおりである。

1. データセット D の中から基準ベクトルとして要素 c をランダムに n 個選ぶ。この n 個の基準ベクトルを $w_k(1<k<n)$とする。
2. 全ての要素ごとに基準ベクトルとの距離を求め、最も近い基準ベクトルが同じであった要素の集合を R_k（$1<k<n$）とする。
3. 集合 R_k に含まれる要素の重心を計算し、新たに w_k とする。
4. ステップ３において一つでも基準ベクトル w_k が変更された場合、ステップ２へ戻る。
5. 基準ベクトルの変更が行われなければ終了。

全ての要素間の距離を厳密に計算する手法では、要素数が n 倍になると計算時間が n^2 倍になってしまうが、LBG アルゴリズムでは基準ベクトルとの距離しか計算しないので、計算時間は要素数の n 倍でおさまることになる。

図 4.1 は LBG アルゴリズムによってクラスタが作られていく様子を表している。○がユニットひとつでありベクトル空間での位置を表している。



（ⅰ）●は初めに基準ベクトルとして選ばれた要素。

（ⅱ）基準ベクトルを結ぶ線分の垂直二等分線で分割。ユニットが 3 つの集合に分かれる。

（ⅲ）集合ごとに要素の重心を新たな基準ベクトルとする。グレーの丸は一段階前の基準ベクトル。

（ⅳ）新たに計算により求めた基準ベクトルで、再び 3 つの集合に分ける。

（ⅴ）～（ⅷ）基準ベクトルの更新がなくなるまで繰り返し。

図 4.1　LBG アルゴリズムによるクラスタリング概略図

4.2　ベクトル作成及びクラスタリング結果

4.2.1　ベクトル作成方法

用語の意味を表すために、用語がもつ特徴を定量しベクトルの形にしたものを意味ベクトルと呼ぶ。そして、各用語の意味ベクトルを比べることで、用語が似ているかどうかを判断することができる。

本研究では、出現しているフィールドや出現頻度といった情報から作成したベクトルを意味ベクトルとして用いた。その作成方法は以下のとおりである。

(1) 用語 t がフィールド f に出現している回数 f_t を求める。
(2) フィールド f から用語が切り出された回数 f_e を求める（表 3.2、表 3.3 参照）。
(3) 用語 t がフィールド f に占める割合 f_t/f_e をベクトルの要素とする。
(4) すべてのフィールドについて(1)から(3)の手順で要素を求める。
(5) ベクトルの正規化を行う

このベクトルは、用語が指す事物の内包的意味は全く考えず、用語がどこにどのように出現しているかという外延に着目して作成したベクトルである。切り出し対象となっているフィールドは全部で 77 種類であるので、作成するベクトルは 77 次元のベクトルとなる。

例として”dna”という用語のベクトル化を行う。この用語が各フィールドに出現した回数は表 4.1 のとおりである。単純な出現回数だけを見てみると、litdb の KEYWORD フィールドが 120 と最も多い。しかし 3 章の表 3.3 によると、このフィールドから用語が切り出された数は 2,783,594 と他のフィールドから切り出された回数と比べてかなり大きい。一方 enzyme の SUBSTRATE フィールドに出現している回数は 23 であり、litdb の KEYWORD フィールドから切り出された回数である 120 と比べて少ない。しかし、enzyme の SUBSTRATE フィールドから用語の切り出しが行われた回数が 8,623 と、元々少なかったことを考慮すると、enzyme の SUBSTRATE に”dna”が出現している割合は、litdb の KEYWORD フィールドに”dna”が出現している割合よりも大きくなる。このように、単純に出現回数だけからベクトルを作成するのは適切ではない。よって、各データベースから用語の切り出しを行った回数で”dna”の出現回数を割り、各フィールドに出現する割合を考えることにした（表 4.2）。出現していないフィールドに対応する要素は当然 0 となる。ここまでで、用語”dna”の特徴を示すベクトルができたわけだが、他の用語の特徴を表すベクトルと比較できるように単位ベクトルに変換した。

以上の操作を全ての用語に対して行い、用語の外延的意味を表すベクトルを作成した。

データベース：フィールド	用語“dna”の出現回数(f_t)
compound:NAME	1
enzyme:PRODUCT	7
enzyme:SUBSTRATE	23
genbank:bound_moiety	51
genbank:rpt_family	95
litdb:KEYWORD	120
pdb:KEYWDS	102
refseq:bound_moiety	3

表 4.1　用語“dna”が出現するフィールドと出現回数

データベース：フィールド	用語”dna”の出現割合(f_t/f_e)
compound:NAME	0.009206
enzyme:PRODUCT	0.000893
enzyme:SUBSTRATE	0.002667
genbank:bound_moiety	0.009206
genbank:rpt_family	0.000666
litdb:KEYWORD	0.000043
pdb:KEYWDS	0.002462
refseq:bound_moiety	0.028302

表 4.2　用語”dna”が出現するフィールドと出現割合

4.2.2 クラスタリング結果

前述の LBG アルゴリズムを用いて、収集した用語のクラスタリングを行った。クラスタリングには、小文字化および特殊記号の削除までの正規化を行った用語を用いた。100 のクラスタを作成した結果、ある程度内包的にも関連が深いと思われる用語を集める事ができた。表 4.3 はクラスタリングによって集められた出現パターンが似ている用語の例である。これらの用語には、生物を意味する用語であるという共通点がある。

一方、あまり関連が無さそうな用語が集まった大きなクラスタも存在した。表 4.4 はこのクラスタに分類されていた用語をいくつか例示したものである。“nematode”は線虫という生物種名である。生物種名が記述される genbank の lab_host に出現したという特徴が、litdb の KEYWORD に出現したという特徴のせいで埋もれてしまっている。”mesenteric artery”は腸間膜動脈という器官を表す用語であり、”piglet”は子豚という発達段階を示す用語として使われているようであるが、やはりこれらも litdb の KEYWORD に出現したという特徴が強いせいで、このクラスタに分類されている。そして、litdb の KEYWORD にはあらゆるカテゴリーに属する用語が記述されているため、このフィールドに出現した用語の集合は内包的意味をもたなかったと考えられる。

pseudomonas fluorescens	蛍光菌
shigella dysenteriae	志賀菌
streptococcus pneumoniae	肺炎連鎖球菌
salmonella enteritidis	サルモネラ腸炎菌
poplar	ポプラ
neisseria gonorrhoeae	淋菌
dermatophagoides pteronyssinus	ヤケヒョウダニ
bacillus megaterium	巨大菌
honeybee	ミツバチ
listeria monocytogenes	リステリア菌
herpes simples virus	ヘルペス・ウィルス
vaccinia virus	ワクシニア・ウィルス
garden pea	エンドウ豆
vibrio cholerae	コレラ菌
cmv	サイトメガロ・ウィルス
klebsiella pneumoniae	肺炎桿菌
grape	ブドウ
slime mold	粘菌
spinach	ホウレンソウ
lupinus luteus	ルピナス
（以下略）	

表 4.3　出現パターンが近い用語の例

用語	出現するデータベース：フィールド：出現する割合
nematode（線虫）	genbank:lab_host:1.53486405325364e-07 litdb:KEYWORD:0.000008
mesenteric artery（腸間膜動脈）	genbank:tissue_type:4.20708624779233e-07 litdb:KEYWORD:0.000011
piglet（子豚）	genbank:dev_stage:2.16145927897177e-07 litdb:KEYWORD:0.000010

表 4.4　分類がうまくいかなかった用語の例

4.2.3　フィールドのカテゴリー化

出現パターンが似ている用語を集めることで、ある程度内包的な共通性を持つ用語を集めることができることがわかった。しかし、単純に切り出し対象フィールド全てから特徴ベクトルを作成し、クラスタリングを行っても、特徴付けに適さないフィールドがあるためにうまくいかないことがわかった。そこで、特徴付けに適さないフィールドを削除した上で、同じ特徴を持つものが記述されているフィールドを一つのカテゴリーにまとめた。例えば、genbank の standard_name フィールドには遺伝子やアミノ酸などの複数のカテゴリーに分類されるべき用語が記述されている。このようなフィールドは特徴としては不適切であるので削除対象とした。また、brite の ORGANISM フィールドや epd の OS フィールドなどはどれも生物種に関する用語が記述されるフィールドである。このような同じ特徴をもつものが記述されているフィールドはまとめて一つのカテゴリーにすることとした。表 4.5 と表 4.6 はカテゴリーとフィールドの対応表である。

カテゴリー	データベース：フィールド
organism	brite:ORGANISM
	epd:OS
	genbank:organism
	refseq:organism
	genome:NAME
	genome:DEFINITION
	pmd:EXPRESSION-SYSTEM
	pmd:SOURCE
	refseq:organism
	swissprot:OS
	transfac:OS
	genbank:specific_host
	refseq:specific_host
	genbank:lab_host
	refseq:lab_host
	genbank:sub_species
	refseq:sub_species
	genbank:variety
	refseq:variety
organism_class	genome:LINEAGE
	swissprot:OC
	transfac:OC
protein	pmd:PROTEIN
	prf:NAME
	transfac:DE
	genbank:product
	refseq:product
	enzyme:NAME
compound	compound:NAME
morphorogy	genome:MORPHOLOGY
physiology	genome:PHYSIOLOGY
sex	genbank:sex
	refseq:sex
gene	genbank:gene
	refseq:gene
mnemonic	brite:MNEMONIC
strain	genbank:strain
	refseq:strai
phenotype	genbank:phenotype
	refseq:phenotype

表 4.5　カテゴリーにまとめたフィールド一覧(1)

カテゴリー	データベース：フィールド
plasmid	genbank:plasmid refseq:plasmid
organelle	genbank:organelle refseq:organelle
tissue_type	genbank:tissue_type refseq:tissue_type
cell_type	genbank:cell_type refseq:cell_type
cell_line	genbank:cell_line refseq:cell_line
enzyme_product	enzyme:PRODUCT
enzyme_cofactor	enzyme:COFACTOR
enzyme_effector	enzyme:EFFECTOR
enzyme_inhibitor	enzyme:INHIBITOR
enzyme_substrate	enzyme:SUBSTRATE
transposon	genbank:transposon refseq:transposon
function	brite:FUNCTION genbank:function refseq:function
dev_stage	genbank:dev_stage refseq:dev_stage
bound_moiety	enbank:bound_moiety refseq:bound_moiety
clone_lib	enbank:clone_lib refseq:clone_lib
rpt_family	genbank:rpt_family refseq:rpt_family
sub_clone	genbank:sub_clone refseq:sub_clone
削除対象	epd:KW litdb:KEYWORD pdb:KEYWDS pir:KEYWORDS swissprot:KW brite:NAME genbank:standard_name refseq:standard_name

表 4.6　カテゴリーにまとめたフィールド一覧(2)

4.2.4 カテゴリー化後のクラスタリング結果

フィールドを 28 のカテゴリーに分類した上で、改めて用語の意味ベクトルを作成した。作成方法は 4.2.1 で述べた方法とほぼ同じである。

(1) 用語 t がカテゴリーc に出現している回数 c_t を求める。
(2) カテゴリーc から用語が切り出された回数 c_e を求める。
(3) 用語 t がカテゴリーc に占める割合 c_t/c_e をベクトルの要素とする。
(4) すべてのカテゴリーについて(1)から(3)の手順で要素を求める。
(5) ベクトルの正規化を行う

(1)のカテゴリー c に出現している回数とは、カテゴリー c に含まれる各フィールドに用語 t が出現した回数を合計したものであり、(2)のカテゴリーから用語が切り出された回数とは、このカテゴリーc に含まれる各フィールドから用語が切り出された回数を合計したものである。

どのカテゴリー化にも属していないフィールド（litdb:KEYWORD 等の削除対象となったフィールド）のみに存在していた用語は、ベクトルを作成することが出来ないのでクラスタリングの対象から外すことにした。その結果 956,777 個の用語が対象から外れたので、残りの 813,785 個のベクトルをクラスタリングした。その結果、概ね期待通りのクラスタリング結果を得ることができた。表 4.4 の例で同じクラスタに入ってしまった 3 つの用語は違うクラスタに分かれ、それぞれが共通性のある用語と同じクラスタに分類された。表 4.7 はこれらの用語と同じクラスタに入った用語の一例である。上から順に、生物種名、組織名、発達段階といった特徴に分類されていることがわかる。

nematode	線虫
mouse	マウス
rat	ラット
xenopus laevis	アフリカツメガエル
mulberry	桑の実
bacteriophage lambda	バクテリオファージλ
orangutan	オランウータン
chick	ヒヨコ
adrenal medulla	副腎髄質
neurula	神経胚
fetus	胎児
brain	脳
liver	肝臓
synovial membrane	滑膜
pollen	花粉
etiolated shoot	暗中退色した苗条
piglet	子ブタ
calf	子ウシ
plantlet	小植物
young mycelia	若い菌糸体
8.5 day embryo	8.5 日目の胚
7 day old seedling	7 日目の実生
pupa	サナギ
larval	幼生の

表 4.7　フィールドのカテゴリー化後のクラスタリング結果

第　５　章
オントロジーブラウザ

実際に作成した外延的オントロジーの情報を用いて、エントリー中の専門知識を見つけ出すためのシステムを作成した。図 5.1 がその概要である。今回作成したオントロジーブラウザは CGI を利用したもので、大きく以下の三つの CGI で構成されている。

(1) 入力テキスト中にある専門用語を見つける CGI
(2) 入力された専門用語と親子関係を持つ用語を表示する CGI
(3) 入力された専門用語と関連する用語を表示する CGI

本章では、これらの機能について順番に説明していく。

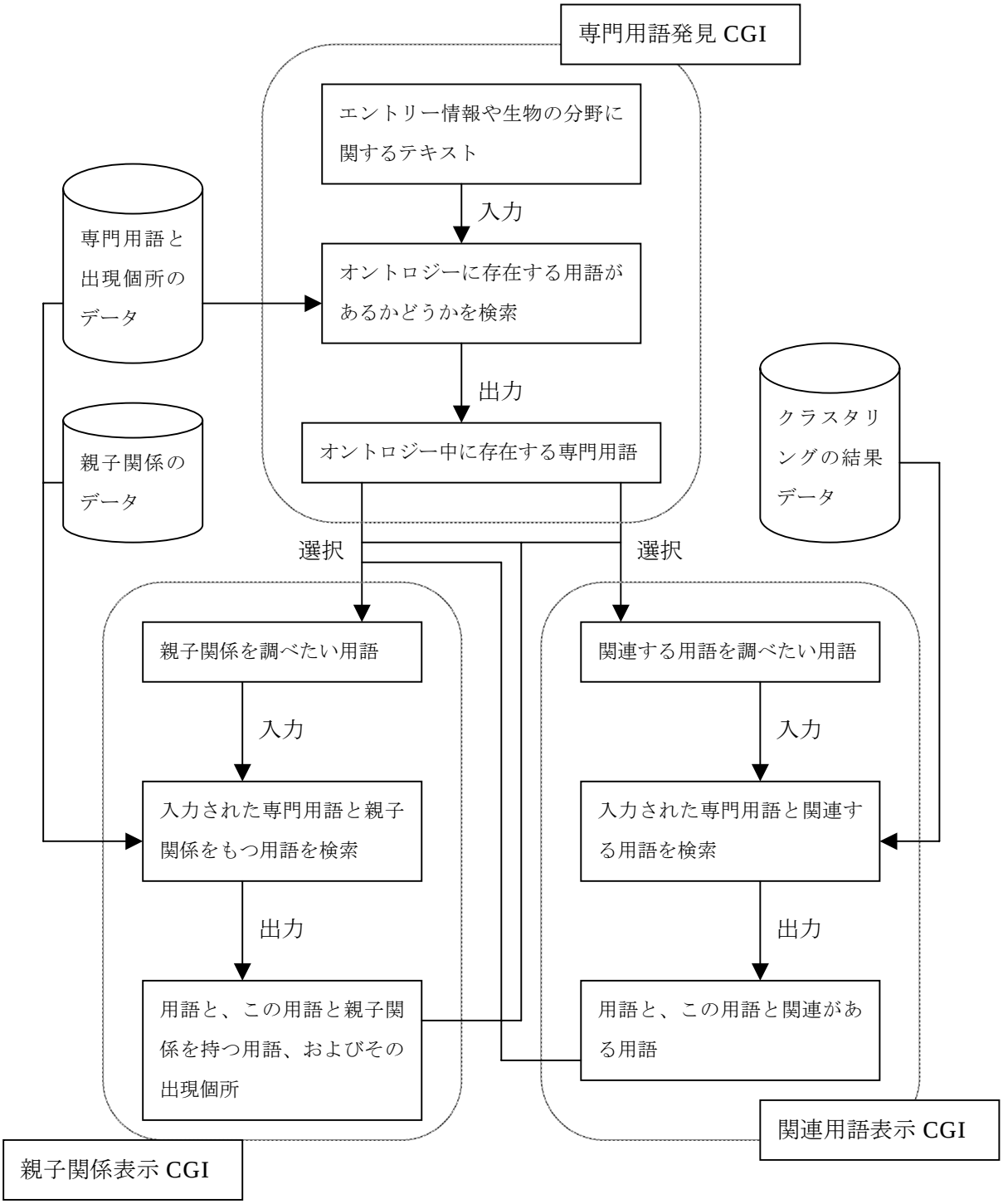


図 5.1　オントロジーブラウザ概略図

5.1　専門用語マッチング CGI

この CGI はテキスト中に存在する専門用語をすばやく見つけ出すためのものである。例として、OMIM データベース中のテキスト（表 5.1）を入力としてこの機能を説明していく。

図 5.2 がテキスト入力画面で、図 5.3 がその出力結果である。出力結果の上部には入力テキストが表示され、その後に専門用語を検索した結果が表示される。下線が引かれている単語は、この単語から始まる用語がオントロジー中に存在していた事を意味する。下線が引かれた単語をクリックすると、この単語から始まる用語の中で、このテキスト中に存在する用語の候補が表示される（図 5.4）。図の例では単語”peripheral”から始まる用語が三つ表示されている。用語の後に括弧で”tissue_type”や”cell_line”と表示されているのは、特に出現頻度が高かったカテゴリーから求めた特徴である。また、その後の”関”というマークは、同じクラスタに分類された用語（関連用語）があることを示している。表示される用語はラジオボタンで選択できるようになっており、選択した用語を入力として、次に説明する親子関係表示 CGI や関連用語を表示するための CGI と連携している。

Text:

Prostaglandins, thromboxanes, and leukotrienes are oxygenated metabolites of arachidonic acid that are thought to act through 7-transmembrane domain-type receptors, reviewed by Coleman et al. (1994). Prostaglandin E2 activates 4 receptors, designated prostaglandin E receptor subtypes EP1 (176802) through EP4. Several cDNAs, originally described as corresponding to an EP2 receptor (176804), were reported by An et al. (1993) and Bastien et al. (1994). The pharmacologically correct EP2 was subsequently isolated by Regan et al. (1994) and the older EP2 sequences were renamed EP4 (Coleman et al., 1994). See Foord et al. (1996) for a more complete discussion of nomenclature issues.

The EP4 receptor is expressed in a variety of tissues including lung, peripheral blood lymphocytes, and vasculature. Foord et al. (1996) isolated the EP4 receptor gene and showed that it consists of 3 exons and spans about 22 kb of genomic DNA. The first exon is noncoding. Exons 2 and 3 encode a predicted 499-amino acid protein. The gene structure
（以下略）

表 5.1　入力テキストの例（OMIM:601586 より抜粋）

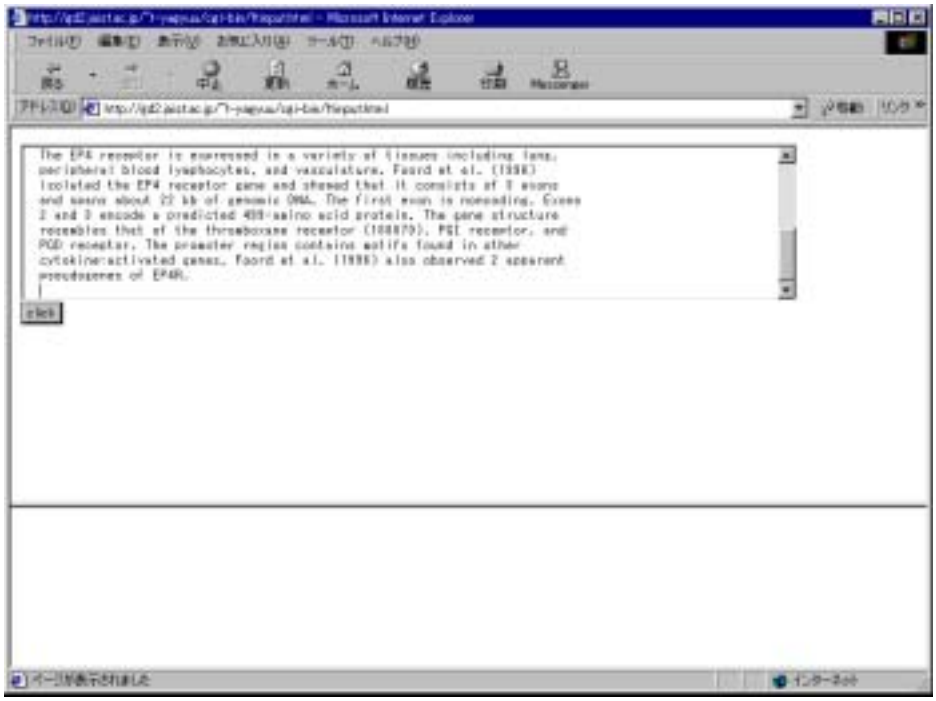


図 5.2　専門用語マッチング CGI の入力画面

図 5.3　専門用語マッチング CGI の出力画面(1)

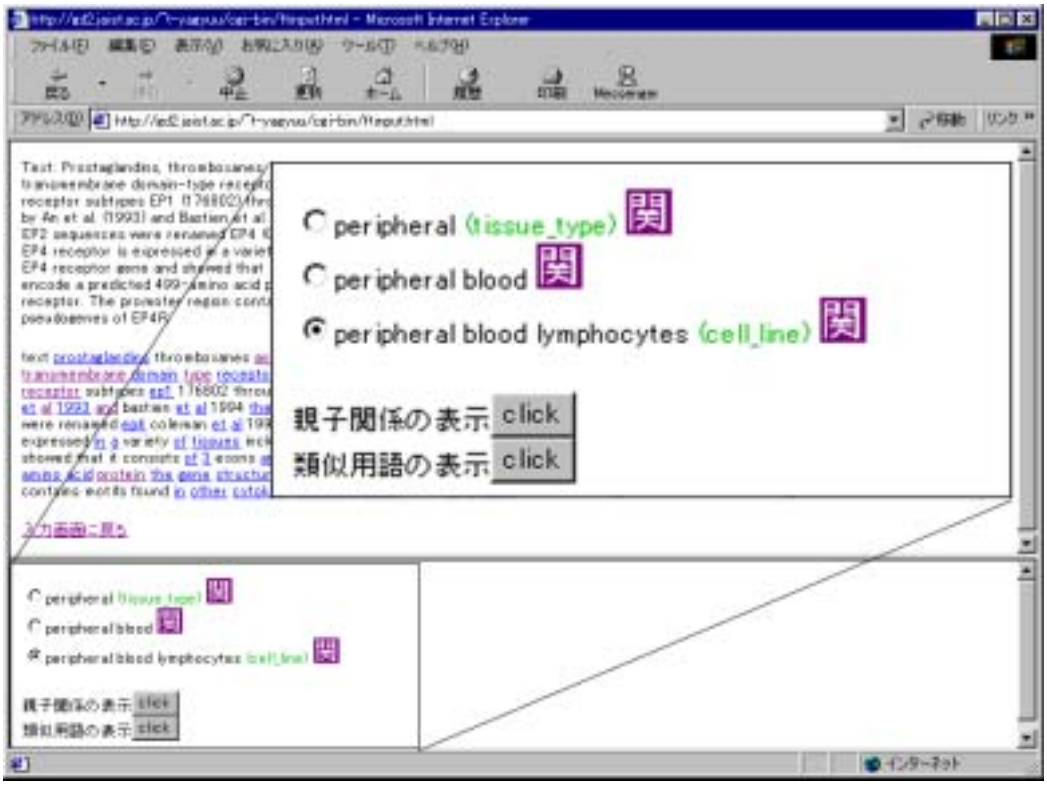


図 5.4　専門用語マッチング CGI の出力画面(2)

5.2　親子関係表示 CGI

この CGI は、オントロジー中の親子関係をたどっていくことで、オントロジー空間を探索するためのものである。

図 5.5 は入力画面で、出力結果が図 5.6 である。他の CGI からこの CGI に来た場合は、入力画面をスキップして出力結果のみを表示する。

入力された用語は一旦正規化された後、オントロジー中に存在しているかマッチングが行われる。その後、親子関係を持つ用語を検索し結果を表示するというのがこの CGI の基本的な機能である。ただし、正規化のうち、語順をソートする処理については行っていない。表示された用語はラジオボタンで選択できるので、この CGI を再帰的に使ってオントロジー空間を自由に移動できる。また、オプションを選択することで表示内容を変えることができる。用語の特殊記号の使われ方による違いを保存した状態での親子関係や、大文字小文字の違いを保存した状態での親子関係を見たい場合には、”正規化レベル”の項目を変更することで意図した親子関係を表示させること

ができる。また、正規化によってまとめられた用語が、もともとどのような形で表記されていたかを知りたければ、”オリジナルの形も表示”にチェックをいれることで、オリジナルの形もあわせて表示できる。さらに、”出現個所も表示”にチェックをいれることで、どのエントリーから切り出された用語であるかを知ることもでき、リンクをたどることで実際に出現したエントリーの情報を見ることもできる(図 5.7)。また、直接の親子関係だけではなく、先祖や子孫全てを表示したいときには、”先祖　子孫全てを表示”にチェックをいれる。これらのオプションを適宜選択することで、希望の情報を表示させることができる。

　また、ここで選択した用語を入力として、関連用語を表示するための CGI を利用することができる。

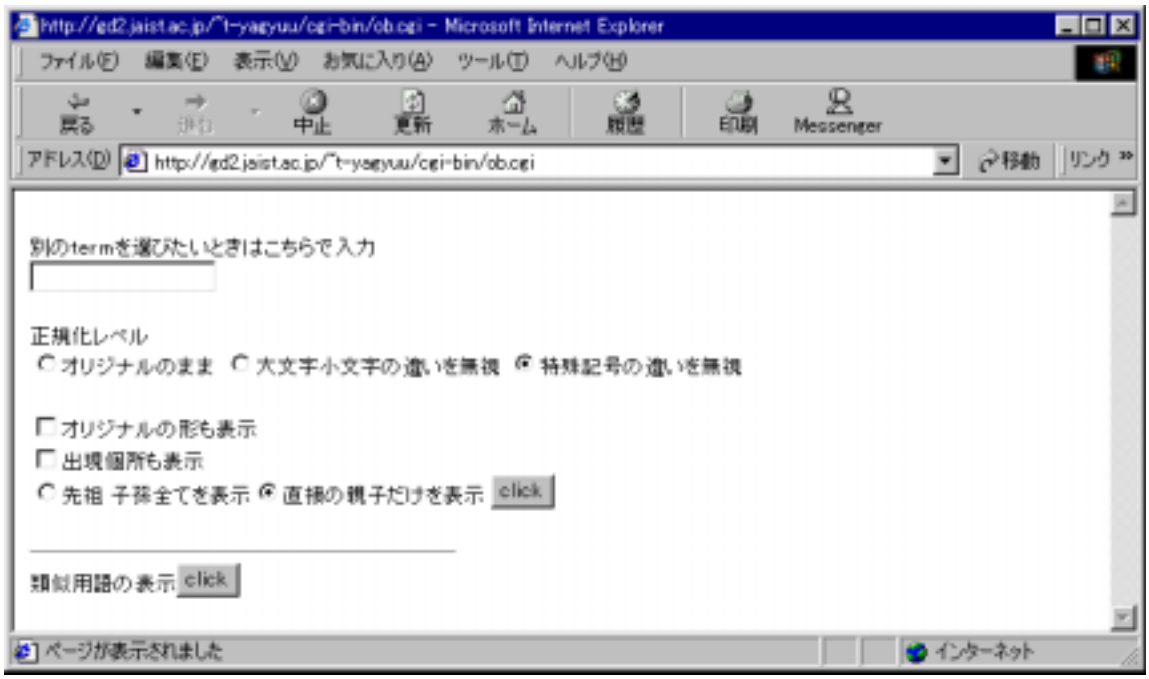


図 5.5　親子関係表示 CGI の入力画面

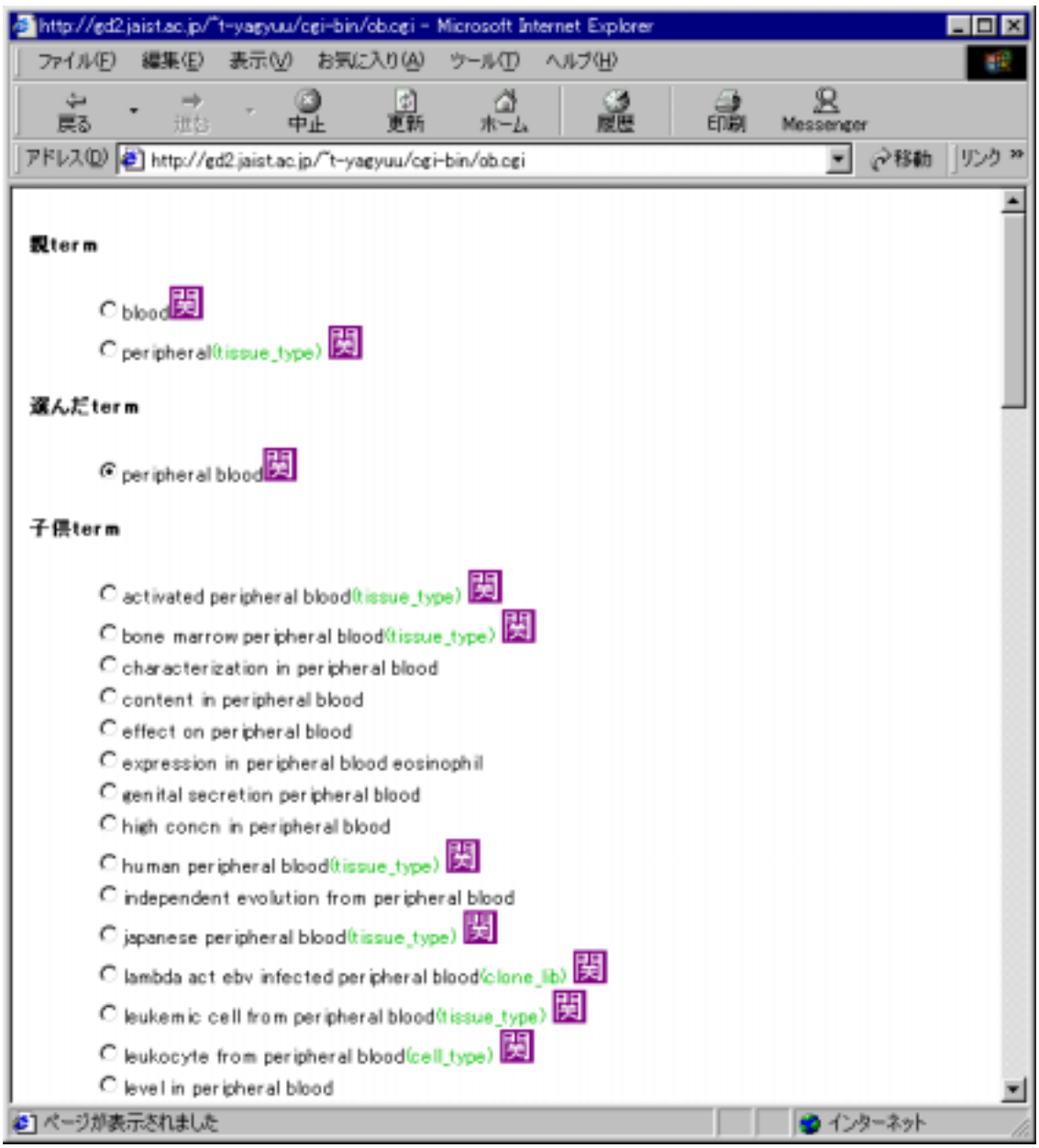


図 5.6　親子関係表示 CGI の出力画面

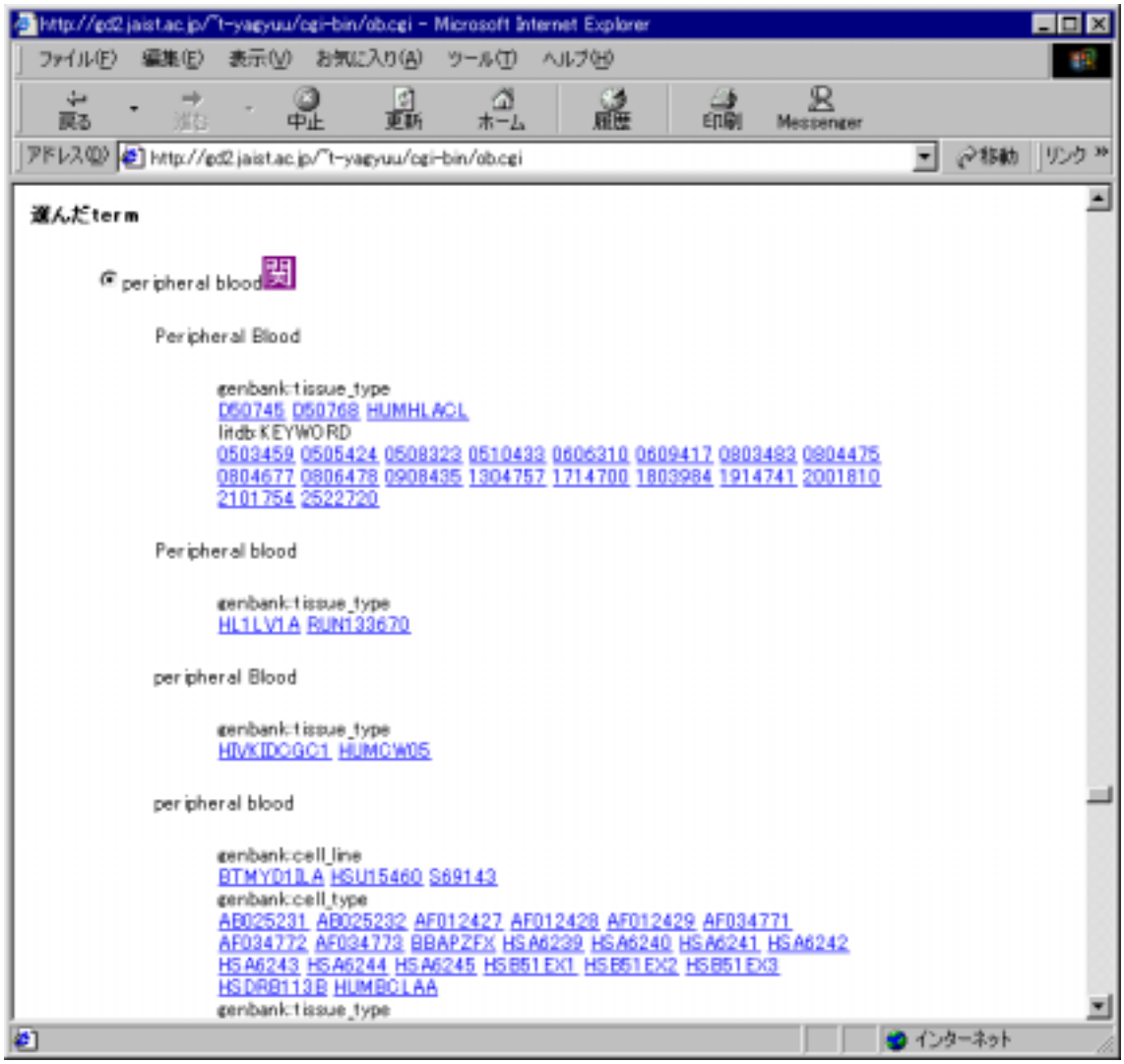


図 5.7　オリジナルの形および出現個所情報の表示例

5.3　関連用語表示 CGI

この CGI は、クラスタリングによって同じクラスタに分類された用語を表示するためのものである。クラスタリングに用いた用語は、特定の正規化（小文字化および特殊記号の削除）を行ったものだけなので、この CGI では正規化レベルの選択オプションはつけていない。図 5.8 は入力画面で図 5.9 は出力画面である。他の CGI からこの CGI に来た場合は、入力画面をスキップして出力結果のみを表示する。出力画面には同じクラスタに存在している用語の数と、その中から１００の用語が表示される。表示される用語はラジオボタンで選択できるようになっており、ここで選択した用語を入力として親子関係表示 CGI を使うことができる。

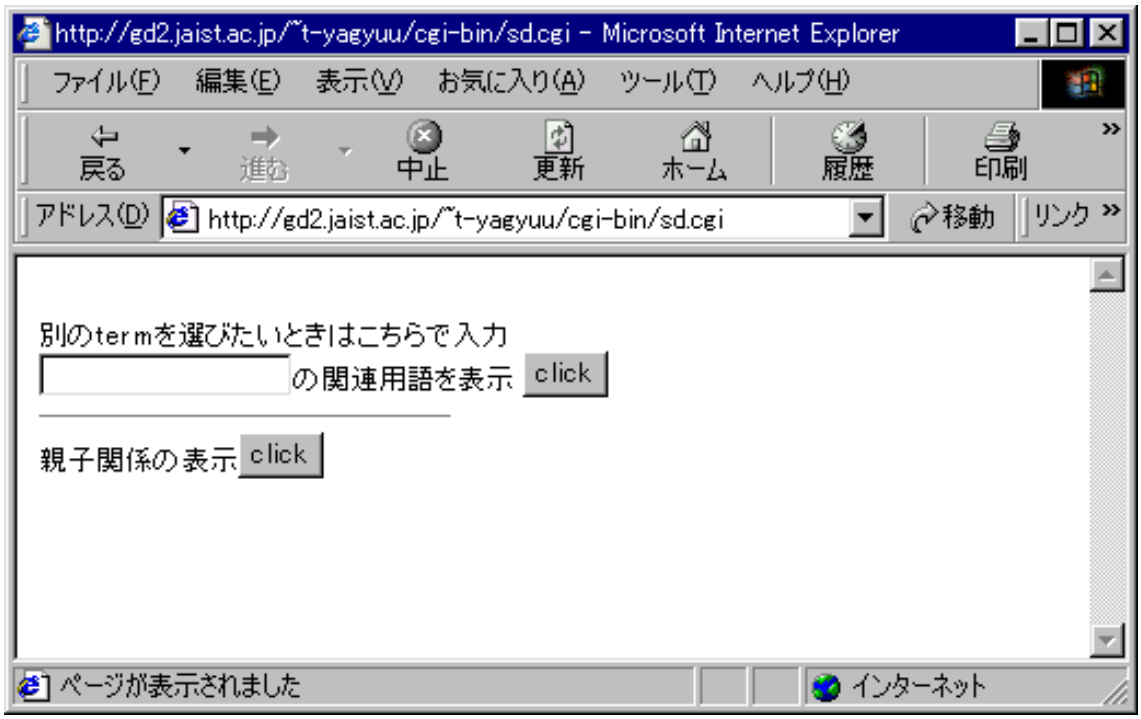


図 5.8　関連用語表示 CGI の入力画面

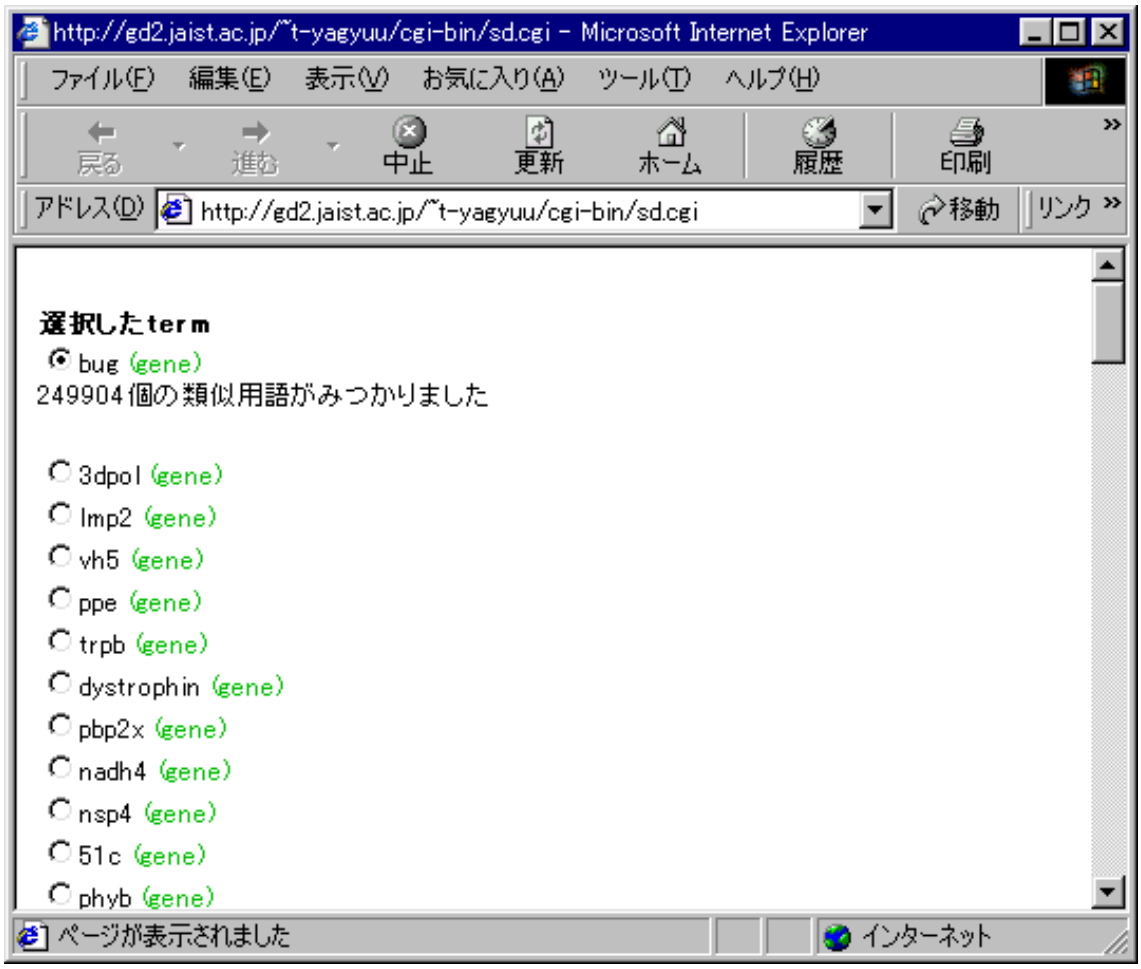


図 5.9　関連用語表示 CGI の出力画面

第 6 章
ステミングによる用語の曖昧さの除去

外延的オントロジーを作成する際に、記述による用語の違いについて 3 点挙げ、これらを軽減するために正規化という処理を行った。しかし、これらの問題以外にも記述の違いによって、本来同じ概念を指す用語が別の概念として扱われる場合がある。例えば、単数形と複数形の違いや、動詞の原型と過去形・進行形といったものの違いである。このような単語の変化を除去し、概念の根本となる形に変換する技術にステミングという手法がある。本研究では実際に外延的オントロジーに利用できなかったが、外延的オントロジーの作成に重要な役割を果たすと思われるため、本章ではステミング技術について簡単に説明した後、実際に行った処理と解決できなかった問題点について述べる。

6.1 ステミング

ステミングとは接頭辞や接尾辞を取り除くことで、同じ概念を指す索引語を作成する技術のことである。例えば、”formula”、”formulate”、”formulation”、そして”reformulate”という単語は、ステミングにより”formula”という元の語義を表す形に変換される[12]。最もよく知られ利用されているステミングアルゴリズムには、Lovins、Porter、simple S-removal といったものがある。これらのアルゴリズムは、語尾リストに対応するものがあった場合はその語尾を単語から取り除く、あるいは置き換えるという手順でステミングが行われる。例えば、simple S-removal アルゴリズムでは表 6.1 のようなルールと制約によりステミングを行っている[12,13]。Porter のアルゴリズムでは段階的な処理により約６０の接尾辞を取り除き、Lovins のアルゴリズムでは longest match と呼ばれるアルゴリズムと例外リストを用いて２６０以上の接尾辞を取り除いている[13]。多くのステミングアルゴリズムでは言葉の意味を無視し[14]、ルールにのっとった字面処理を行うことでステミングを行っている。そのため、ほぼ期待通りにステミングを行うことはできるが、誤りを完全に無くすことは出来ないというのが現状である。表 6.2 は Porter アルゴリズムによるステミング

誤りの例である。

ルール	語尾	変換	制約
1	ies	ies->y	aies、eies の場合を除く
2	es	es->e	aes、ees、oes の場合を除く
3	s	s->null	us、ss の場合を除く

表 6.1　simple S-Removal ステミングのルール

同じ形にステミングされる 別の意味を持つ単語		別の形にステミングされる 同じ概念からなる単語	
organization	organ	european	europe
doing	doe	analysis	analyzes
generalization	generic	cylinder	cylindrical
numerical	numerous	matrices	matrix
policy	police	urgency	urgent
university	universe	create	creation
easy	easily	decompose	decomposition
addition	additive	machine	machinery
negligible	negligent	useful	usefully
execute	executive	noise	noisy
define	definite	route	routed
past	paste	search	searcher
ignore	ignorant	sparse	sparsity
special	specialized	explain	explanation
arm	army	resolve	resolution
head	heading	triangle	triangluar

表 6.2　Porter アルゴリズムによるステミング誤りの例

6.2　切り出した専門用語のステミング結果

ステミングツール[12]を入手できたので、ステミングアルゴリズムの詳しい検証はせず、実際にゲノムデータベースから切り出した用語のステミングを試みた。このステミングツールでは、Lovins や Porter、S-removal といったステミングのルールを選択することができる。そこで、Porter アルゴリズムのルールを選択し用語のステミングを行ったが、ステミングによって用語がまとまるという例が少なかった。さらにその中でも、複数形の違いや活用の違いが除去された良い例よりも、好ましくないステミング結果が多かった。表 6.3 はステミング結果の例である。”grape berries”と”grape berry”は単数形と複数形の違いであるので、この二つが同じ形にステミングされることは全く問題ない。しかし、”Grape”と”Grap”のように全く違うものを指す言葉が同じ形にステミングされてしまうのは大変困る。ちなみに”GRAP”は”Grb2-related adaptor protein”の略であり、ブドウとは全く別のものである。さらに、”数字を全て空白に置き換えた後でステミングを行う”というこのツールの仕様のために、表 6.4 のようなステミング結果も見られた。数字の違いを無視されてステミングが行われた結果、本来区別されるべき遺伝子名やタンパク質名が同じ形になってしまっている。これらの問題を解決する時間的余裕が無かったため、結局ステミングを外延的オントロジーの作成に生かすことはできなかった。しかし、文字数の短い略語などに対する例外処理を追加し、数字を無視しない形でのステミングプログラムを作成し適用することで、よりよい外延的オントロジーの作成に役立てることができると考えられる。

ステミング後の形	＜－	切り出された専門用語
grap berri	＜－	grape berries
		grape berry
grap	＜－	GRAP 1
		GRAP2
		Grap
		Grape

表 6.3　ステミングツールを用いたステミングの結果(1)

ステミング後の形	＜－	切り出された専門用語
p	＜－	p35
		p7482
		pp59
		ps915
e	＜－	E1
		E13
		E-64
		e14.5

表 6.4　ステミングツールを用いたステミングの結果(2)

第　7　章
まとめと今後の課題

7.1　まとめ

本研究では、ゲノムデータベースから専門用語を切り出し、どのデータベースのどのフィールドに出現しているかという情報を基に外延的オントロジーを作成した。用語の正規化を行うことで記述の曖昧さを軽減し、単語の包含関係（親子関係）を調べることで、階層的な概念の関係を作成することができた。そして、切り出し対象となったフィールドをカテゴリー化し、用語が各カテゴリーに出現している割合から作成したベクトルをクラスタリングすることで、出現情報のみからでも、ある程度の共通性を持った用語に分類することができることがわかった。専門知識を用いた厳密な分類を必要とせず、計算機で自動的に処理させることができたため、比較的短期間で大量の専門用語から構成されるオントロジーを作成することができた。

また、外延的オントロジーを作成した後、このオントロジーを利用するためのオントロジーブラウザを作成した。オントロジーブラウザを利用することで、テキスト中にある専門用語を見つけ、外延的オントロジー空間の移動を行い、同じクラスタに分類された関連用語の情報を得ることができた。

7.2　今後の課題

(1)出現頻度からの用語の重要性・一般性の測定

重要な概念を表す用語は出現頻度が高く、一般的な概念を指す用語ならば様々なデータベースに出現していると考えられる。また、狭い概念を指す用語は逆に出現頻度が低いと考えられる。本研究では単語の包含関係から用語が示す概念の大きさを比べているが、用語がいくつのデータベースに出現しているか、何回出現しているかとい

う情報から、用語の重要性や一般性を測ることができると考えられる。

(2)親子関係からの用語の重要性の測定

重要な概念は出現頻度が高いだけではなく、他の概念との関わりも強いと考えられる。HEP(Heavy Edge Property)[15]というデータマイニング手法は、検索サービス”Google”[16]のページランキングにも採用されているもので、重要なノードと繋がるノードもやはり重要であるという考え方に基づき、ノードに得点を与えていくことでノードのランキングを行うというものである。この手法を親子関係のグラフに適用することで、専門用語の重要度ランキングができると考えられる。

(3)共起関係からの用語の関連度の測定

同一エントリーに出現している用語は関連が深いという仮説に基づき、用語の共起関係を調べることで、用語の関連度の測定を行うことができると考えられる。このような関連度から用語の関係を表すグラフを作成することができれば、親子関係のグラフから得られるものとは別の有用な知識を得ることができると考えられる。

(4)切り出し対象データベース及びフィールドの拡大

外延的オントロジーでは、用語の出現情報からその特徴や別の用語との関係を導き出している。よって、用語が出現している個所の情報が増えることで、より正確に用語の特徴や用語間の関係付けができると考えられる。本研究で専門用語の切り出し対象となっているデータベースとフィールドは 77 種類であったが、専門用語切り出しプログラムを工夫することで、今回切り出し対象にできなかったフィールド以外からも用語を切り出すことができるであろう。さらに、ゲノムネットで利用できるデータベース以外のゲノムデータベースからも用語の収集を行うことができればよりよいオントロジーが作成できると考えられる。

謝辞

本研究を進めるにあたり、出来の悪い生徒であった私に、優しく、厳しく、根気よく、そして適切な御指導、御助言を与えてくださった佐藤賢二助教授に深く感謝いたします。多くの心配とご迷惑をかけたこと以外には何も思い残すことはありません。良い先生に巡り合えたことを大変感謝しております。

遺伝子知識システム論講座の小長谷教授には、多くの助言を頂きました。また、研究室のメンバーへの様々な差し入れには大変助けられました。心より御礼を申し上げます。

助手のクサビエ先生と山本先生にも大変お世話になりました。同じ講座の先輩・同輩・後輩にも大変助けられました。討論の機会と助言、励まし、そして多くの笑いを提供してくださった講座のメンバー全員に、改めて御礼申し上げます。

最後に、長い間脛をかじらせてくれた両親と、私と関わってくださった全ての人達への感謝の気持ちを述べ、結びの言葉とさせていただきます。

参 考 文 献

[1] Baker, P. G. et al. "*Ontology for Bioinformatics Applications*", Bioinformatics, Vol. 15, No. 6, 510--520, 1999.

[2] The Gene Ontology Consortium. "*Gene Ontology: tool for the unification of biology*", nature genetics volume 25 may 2000, pp.25-29,2000

[3] P. D. Krap. et al. "*The Ecocyc Database*", Nucleic Acids Research, 30(1):56 2002

[4] "*Generic Knowledge-Base Editor*", http://www.ai.sri.com/~gkb/

[5] 財団法人 日本情報処理開発協会、"大規模知識ベースに関する調査研究―オントロジー工学に関する調査研究―報告書", 1999

[6] T. R. Gruber. "*A translation approach to portable ontologies*", Knowledge Acquisition, 5(2) pp.199-220, 1993

[7] 高井貴子,高木利久: "生命科学のためのオントロジー": 実験医学 Vol.19 No.11 (増刊) 2001 pp.47-53

[8] P. D. Karp. "*An Ontology for Biological Function Based on Molecular Interactions*", Bioinformatics 16(3):269-85.2000

[9] 大久保公策、 "医学知識を機械に伝えるーBOB プロジェクト": ポストシークエンスのゲノム科学(2) ゲノム機能 pp.134-143, 2000

[10] "*GenomeNet WWW server*", http://www.genome.ad.jp/

[11] Y. Linde, A. Buzo, and R. Gray. "*An algorithm for vector quantizer design*", IEEE Trans. Commun., vol. 28, pp. 84-95, Jan. 1980

[12] Brian Fox, Christopher J.Fox. "*Efficient Stemmer Generation*", Information Processing and Management, in press.

[13] Michael Fuller, Justin Zobel. "*Conflation-based Comparison of Stemming Algorithms*", Proceedings of ADCS'98 Third Australian Document Computing Symposium, 1998

[14] Robert Krovetz. "*Viewing Morphology as an Inference Process*", Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.191-202, 1993

[15] Brin S, Page L. "*Dynamic Data Mining: Exploring Large Rule Spaces by Sampling*", Stanford University, Paper number 261.

[16] "*Google*", http://www.google.com/

研究業績

Takuya Yagyu, Kenji Satou. “Toward Automatic Construction of Extensional Ontology from Genome Databases”, Genome Informatics 2000, UNIVERSAL ACADEMY PRESS,INC. TOKYO,JAPAN