단백질 상호작용 네트워크에서의 개념 기반 기능 모듈 탐색 기법

(Concept-based Detection of Functional Modules in Protein Interaction Networks)

박 종 민 † 최 재 훈 † 박 수 준 †† 양 재 동 †††

(Jong-Min Park) (Jae-Hun Choi) (Soo-Jun Park) (Jae-Dong Yang)

요 약 단백질 상호작용 네트워크는 생체 내에서 특정 역할을 담당하는 패스웨이나 복합체와 같은 중요한 의미의 많은 기능 모듈들을 포함하고 있다. 본 논문에서는 이 기능 모듈들과 정합될 수 있는 개념 모듈을 정의하고 이를 기반으로 원하는 기능 모듈들을 개념적으로 표현하고 효율적으로 탐색할 수 있는 새로운 방법을 제안한다. 개념 모듈은 트리플들과 이들 사이의 연산자로 이루어진 표현 규칙에 의해 정의되며 탐색하고자 하는 기능 모듈들의 구조를 개념적으로 표현한다. 이 표현 규칙에서의 트리플은 한 기능 모듈을 구성하는 단백질들 사이의 구체적인 상호작용 관계를, 연산자는 트리플들 사이의 구조적인 연관 관계를 각각 개념적으로 정의한다. 또한, 사용자는 사전에 표현 규칙에 의해 잘 정의된 개념들을 조합하여 새로운 의미의 복합 개념 모듈을 정의할 수도 있다. 복합 개념 모듈은 복잡한 기능 모듈들의 개념적 구조를 보다 정교하게 표현할 수 있기 때문에, 사용자 탐색 질의의 의미적 표현력을 획기적으로 높일 수 있다. 정의된 규칙들은 XML로 관리될 수 있어 다른 종류의 단백질 상호작용 네트워크에서 사용자가 유사한 모듈들을 탐색하기 위해 쉽게 적용 가능하다. 본 논문에서는 또한, 구조적으로 복잡한 규칙들을 직관적으로 표현하고 효율적으로 탐색하기 위한 시각화된 질의 환경도 구현하였다.

키워드 : 개념 기반 탐색, 트리플 기반 규칙 정의, 기능 모듈, 단백질 상호작용 네트워크

Abstract In the protein interaction network, there are many meaningful functional modules, each involving several protein interactions to perform discrete functions. Pathways and protein complexes are the examples of the functional modules. In this paper, we propose a new method for detecting the functional modules based on concept. A conceptual functional module, briefly concept module is introduced to match the modules taking them as its instances. It is defined by the corresponding rule composed of triples and operators between the triples. The triples represent conceptual relations reifying the protein interactions of a module, and the operators specify the structure of the module with the relations. Furthermore, users can define a composite concept module by the counterpart rule which, in turn, is defined in terms of the predefined rules. The concept module makes it possible to detect functional modules that are conceptually similar as well as structurally identical to users' queries. The rules are managed in the XML format so that they can be easily applied to other networks of different species. In this paper, we also provide a visualized environment for intuitionally describing complexly structured rules.

Key words : concept-based detection, triple-based rule definition, functional module, protein interaction network

· 본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력핵심기술개발사업의 일환으로 수행하였음.[2006-S-007-02, 유비쿼터스 건강관리용 모듈/시스템]

† 정 회 원 : 한국전자통신연구원 라이프인포매틱스팀 연구원
jmpark93@gmail.com
jhchoi@etri.re.kr
†† 정 회 원 : 한국전자통신연구원 라이프인포매틱스팀 팀장
psj@etri.re.kr
††† 정 회 원 : 전북대학교 전자정보공학부 교수
jdyang2000@paran.com

논문접수 : 2007년 3월 16일
심사완료 : 2007년 6월 11일

1. 서 론

단백질 상호작용 네트워크는 생체 내에서 일어나는 주요한 현상들을 통합한 시스템으로서 생물학적 메커니즘을 전체적으로 이해하기 위한 수단으로 사용된다[1]. 따라서, 이 네트워크에는 여러 단백질 상호작용들이 서로 연관되어 하나의 특정한 기능을 수행하는 많은 기능 모듈들이 존재한다[2]. 기능 모듈들은 세포 구조를 구성

하는 기본적인 단위로 볼 수 있으며 대표적인 예로 다양한 패스웨이 또는 복합체 등이 있다. 패스웨이는 "Apoptosis," "Parkinson's disease" 등과 같이 세포 대사를 조절하거나 질병을 유발하는 여러 단백질 사이의 논리적인 신호 전달 경로이며, 복합체는 "Hemoglobin," "Ribosome" 등과 같이 특정 기능을 수행하기 위해 여러 단백질이 물리적으로 결합한 구조이다. 특히, 질병과 관련 있는 특정한 기능 모듈들은 신약 개발의 주요 표적이 될 수 있으며, 이 모듈들에 포함된 다른 단백질들과의 생물학적인 반응들을 분석하여 긍정적이거나 부정적인 효과들을 미리 예측할 수 있다.

현재, 단백질 상호작용은 "Yeast Two-Hybrid"[3]와 같은 생물학적 실험 방법을 통해 빠르게 추출되고 있으며, 단백질 복합체는 "TAP-MS"[4], "HMS-PCI"[5]와 같은 실험 방법에 의해 추출되고 있다. 그러나, 이 실험 데이타로부터 사용자가 관심 있는 특정 기능을 수행하는 모듈을 추출하기 위해서는 많은 시행 착오를 수반하는 생물학적 실험을 반복해야 하고, 이 작업은 필연적으로 많은 시간과 비용을 필요로 하게 된다. 실험을 수행하기 전에 복잡하고 방대한 단백질 상호작용 네트워크에서 목표로 하는 기능 모듈을 미리 예측하거나, 기존에 이미 알려진 기능 모듈을 분석하기 위한 다양한 탐색 방법들이 시도되고 있다[6-17]. 대표적으로 키워드 기반 탐색[6-10]은 먼저 사용자가 입력한 특정 키워드와 일치하는 단백질들을 검색하고, 이 단백질들을 중심으로 서로 상호작용하는 다른 단백질들로 구성된 부분 네트워크를 탐색한다. 이 방법은 매우 단순하지만 사용자가 관심 있는 기능 모듈을 탐색하기 위해서는 많은 시간과 노력이 수반된다. 클러스터링 기반 탐색[11-15]은 복합체에 포함된 단백질들이 서로 많은 상호작용을 한다는 특성을 이용하여 네트워크로부터 밀집된 부분 네트워크를 탐색한다. 그러나, 이 방법 역시 단백질들이 서로 다른 기능을 수행하지만 개념적으로 연관성을 가지고 있는 다양한 구조의 기능 모듈들을 탐색할 수 없다는 문제점을 가지고 있다.

본 논문에서는 이러한 문제들을 해결하기 위해서 기능 모듈들과 정합될 수 있는 개념 모듈을 정의하고 이를 기반으로 원하는 기능 모듈들을 개념적으로 표현하고 효율적으로 탐색할 수 있는 새로운 방법을 제안한다. 개념 모듈은 트리플들과 이들 사이의 연산자로 이루어진 표현 규칙에 의해 정의되며 탐색하고자 하는 기능 모듈들의 구조를 개념적으로 표현한다. 이 표현 규칙에서의 트리플은 기능 모듈들을 구성하는 단백질들 사이의 구체적인 상호작용 관계를, 연산자는 트리플들 사이의 구조적인 연관 관계를 각각 개념적으로 정의한다. 또한, 사용자는 사전에 표현 규칙에 의해 잘 정의된 개념들을 조합하여 새로운 의미의 복합 개념 모듈을 정의할 수도 있다. 복합 개념 모듈은 복잡한 기능 모듈들의 개념적 구조를 보다 정교하게 표현할 수 있기 때문에, 사용자 탐색 질의의 의미적 표현력을 획기적으로 높일 수 있다. 정의된 규칙들은 XML로 관리될 수 있어 다른 종류의 단백질 상호작용 네트워크에서 사용자가 유사한 모듈들을 탐색하기 위해 쉽게 적용 가능하다. 본 논문에서는 또한, 구조적으로 복잡한 규칙들을 직관적으로 표현하고 효율적으로 탐색하기 위한 시각화된 질의 환경도 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 기존의 연구들에 대해 살펴본다. 3장은 개념 모듈들을 정의하기 위한 개념 모듈 표현 규칙을 기술하고, 개념 모듈과 구조적으로 정합되는 기능 모듈들을 탐색하는 방법 및 유사도를 고려한 질의 처리에 대해 설명한다. 4장에서는 개념 기반 탐색 방법에 대한 설계 및 구현에 관한 내용을 기술한다. 마지막으로 5장에서는 결론 및 향후 연구과제를 제시한다.

2. 관련 연구

현재, BIND[6], DIP[7], MINT[8] 등에서는 많은 단백질 상호작용 데이타를 관리하고 있으며 키워드 기반 단백질 검색을 통해 기능 모듈을 점진적으로 탐색할 수 있도록 지원한다. 사용자는 먼저 자신이 요구하는 기능 모듈에서 중요한 역할을 하는 단백질들을 키워드를 통해 검색한 뒤, 검색된 단백질을 중심으로 상호작용 관계를 가지고 있는 이웃 단백질들을 점진적으로 포함시키면서 모듈을 탐색한다. 이 방법은 사용자가 직접 단백질의 정보를 참조하면서 모듈의 범위를 점진적으로 확장해야 하기 때문에 많은 시간과 노력이 요구된다. 이를 개선하기 위해, Reactome[9], PATIKAweb[10]에서는 시작과 종료 단백질만을 키워드 방식으로 검색한 다음, 이 두 단백질 사이 여러 경로들을 탐색함으로써 기능 모듈을 탐색한다. 그러나 이 방법 역시 사용자가 원하는 기능 모듈을 탐색하기 위해서는 경로 주변에 있는 단백질들을 사용자가 직접 참조하면서 모듈의 범위를 확장해야 하는 단점을 가지고 있다.

이 단점을 보완하기 위해 단백질 상호작용 네트워크에서 기능 모듈들을 자동으로 탐색할 수 있는 많은 방법들이 연구되고 있다[11-15]. 이 연구들은 대부분 단백질 상호작용 네트워크를 단순한 그래프로 모델링하고 기존의 그래프 이론들을 적용하여 기능 모듈들을 탐색한다. Li et al.[11]과 Zhang et al.[12]은 실험을 통해 밝혀진 단백질 복합체들의 상당부분이 물리적으로 밀집된 연결 관계를 가지고 있다는 구조적인 특징을 이용하고 있다. 즉, 네트워크에서 모든 단백질들이 서로 완전

한 연결을 가지는 클릭(Clique) 구조의 부분 네트워크를 기능 모듈로 탐색한다. 그러나, 이 클릭 탐색은 네트워크 데이타에 생략된 상호작용 관계가 존재하거나, 클릭 구조가 아닌 다른 형태의 기능 모듈을 탐색할 수 없다는 문제점을 가진다. Bader and Hogue[13]와 Lubovac et al.[14]는 완전한 클릭 구조는 아니지만 어느 정도 밀집된 구조를 가지면서 구성 단백질들이 서로 개념적으로 유사한 기능을 수행하는 부분 네트워크를 탐색하는 방법을 제안하였다. 특히, Bader and Hogue[13]는 각각의 단백질에 이웃 단백질들과의 물리적인 연결 정도에 따라 가중치를 부여한 다음, 가중치가 높은 단백질을 중심으로 가중치가 일정 정도 이상인 이웃하는 단백질들을 순차적으로 모듈에 포함시키면서 확장한다. 이들 중에서 부분 네트워크에 포함된 구성 단백질들의 기능이 서로 유사한 것들만을 기능 모듈로 제시한다. 유사한 방법으로, Koyuturk et al.[15]는 다양한 생물학적인 네트워크에서 상호작용 관계의 출현 빈도에 따라 모듈을 확장하면서 부분 네트워크를 탐색하는 방법을 이용하고 있다. 즉, 여러 네트워크들에서 각각의 단백질 상호작용 관계의 출현 빈도를 계산한 다음, 가장 빈번하게 나오는 관계를 중심으로 주변에 출현 빈도가 높은 상호작용 관계를 점진적으로 확장하여 기능 모듈을 탐색한다. 하지만, 이 두 방법 역시 패스웨이 형태와 같은 복잡한 구조를 가지거나, 구성 단백질의 기능은 서로 다르지만 개념적으로 연관성을 가지고 있는 기능 모듈들을 탐색할 수 없다는 단점을 가지고 있다.

최근에, 생물학적인 네트워크에서 사용자가 원하는 구조를 가지는 기능 모듈을 탐색하기 위해 Leser[16]는 PQL(Pathway Query Language)을 정의하였다. PQL은 SQL과 마찬가지로 "SELECT," "FROM," "WHERE" 세 부분으로 구성되며, "WHERE" 절에 단백질의 속성을 지정하거나 단백질들 사이의 존재하는 경로에 대한 제약 조건을 명시함으로써 비교적 다양한 구조를 가지는 기능 모듈을 탐색한다. 비슷한 형태의 질의 언어로서, Baitaluk et al.[17]은 생물학적인 상호작용 데이타들을 그래프 형태로 통합하기 위한 PathSys 시스템에서 부분 네트워크를 탐색하기 위한 질의 언어로 BioNetSQL를 정의하였다. 그러나, PQL과 BioNetSQL은 모두 질의에 정의된 단백질 속성과 정확히 일치하는 단백질들만을 대상으로 하며, 여러 모듈들이 조합되어 하나의 복합 구조를 가지는 기능 모듈에 대한 개념적인 질의 표현이 불가능하다는 단점을 가지고 있다. 또한, 사용자가 직접 질의 언어를 기술해야 하기 때문에 복잡한 구조를 가지는 기능 모듈들을 탐색하기 어렵다.

다음 절에서는 지금까지 언급한 문제점들을 본 논문에서 제안한 규칙 기반 개념 모듈 탐색 기법이 어떻게 해결하고 있는 가를 구체적으로 기술하도록 한다.

3. 개념 기반 기능 모듈 탐색 기법

본 절에서는 사용자가 의도하는 개념 모듈을 트리플 규칙으로 표현하고, 정의된 규칙과 개념적으로 그리고 구조적으로 일치하는 기능 모듈들을 탐색할 수 있는 방법을 설명한다. 이를 위해 먼저 단백질 상호작용 네트워크를 정의한다.

3.1 단백질 상호작용 네트워크

단백질 상호작용 네트워크는 N=<P, R>로 표현되며, P는 단백질들의 집합, R은 단백질들 사이의 생물학적 상호작용 관계들의 집합을 나타낸다. N에 존재하는 단백질들의 집합과 단백질들 사이의 상호작용 관계들의 집합은 P(N)과 R(N)으로 각각 표현된다. 본 논문에서 사용되는 각 단백질 $p \in P(N)$에 대한 정보는 SWISS-PROT[18] 데이타베이스를 사용하였으며 데이타베이스 스키마는 다음과 같이 표현된다.

정의 1. P(N)을 단백질들의 집합이라고 할 때, 단백질 데이타베이스 스키마, *schema*(P(N))는 다음과 같이 정의된다.

$$schema(P(N)) = <ID, NAME, GO>$$

여기서, ID는 단백질에 대한 대표 식별자이며, NAME은 유전자 이름을 포함한 단백질에 대한 명칭, GO[20]는 단백질의 특성을 표현한 온톨로지 용어이다.

예 1. 표 1은 N를 구성하는 단백질 $p \in P(N)$에 대한 예이다. 특정 단백질에 정의된 속성 정보들은 "."를 통해 참조할 수 있는데, 예를 들어, p_8.ID="Q99179"에 대한 명칭들은 p_8.NAME={"CDCrel1," "SEPT5," "Septin-5," …}으로, 관련 온톨로지 용어들은 p_8.GO={"cytokinesis," "regulation of exocytosis," …}으로 참조할 수 있다.

정의 2. $r \in R(N)$을 단백질들 사이의 상호작용 관계 또는 간단히 관계라고 할 때, r은 다음과 같이 정의된다.

$$r = < p_i, p_j, type_{ij} >.$$

여기서, $type_{ij} \in$TYPE은 두 단백질 $p_i, p_j \in P(N)$ 사이의 상호작용 형태를 나타낸다. $type_{ij}$는 "bind," "activate," "regulate," "decrease," "increase" ∈TYPE 등과 같이 두 단백질 사이에 존재하는 다양한 상호작용 형태들 중 하나로 표현된다.

기능 모듈들은 세포에서 생물학적 또는 화학적으로 독립된 기능을 수행하며, 단백질 상호작용 네트워크의 부분 네트워크로 정의될 수 있다. 즉, 단백질 상호작용

표 1 단백질 상호작용 네트워크 N을 구성하는 단백질 데이타베이스

	ID	NAME	GO
...	...	...	...
p3	P68036	UBCH7, E2, ···	0019899; enzyme binding 0006464; protein modification ···
p4	O14933	UBCH8, E2, ···	0004840; protein binding 0006464; protein modification ···
...	...	...	...
p7	O60260	PARK2, Parkin, ···	0006511; ubiquitin-dependent protein catabolism 0007417; central nervous system development ···
p8	Q99719	CDCrel1, SEPT5, Septin-5, ···	0000910; cytokinesis 0017157; regulation of exocytosis ···
...	...	...	...

네트워크 N에 대해 기능 모듈 m은 다음과 같이 정의된다.

정의 3. N=<P, R>을 단백질 상호 작용 네트워크라고 하고, N내의 모든 기능 모듈들의 집합을 M이라고 할 때, 한 기능 모듈 $m \in M$은 다음과 같이 정의된다.

m = <P, R>, 여기서 $P(m) \subseteq P(N)$이고 $R(m) \subseteq R(N)$.

예 2. 그림 1은 KEGG[19]에서 참조한 "Parkinson's disease" 패스웨이에서 유전자 "PARK2"와 관련된 한 기능 모듈 m을 위에서 정의한 단백질과 상호작용 관계를 이용하여 표현하였다. m에서 유전자들은 단백질들로 변환되고, 이들 사이의 관계 형태는 그대로 유지되며, 중복되는 유전자들은 하나의 단백질로 통합된다. 예를 들어, p_8.NAME={"CDCrel1," "SEPT5," "Septin-5,"···} 이기 때문에 이 패스웨이에서 "CDCrel1"으로 표현된 노드는 식별자가 "Q99719"인 단백질 p_8이며 [Q99719, CDCrel1]로 표현하였다. p_7과 p_8사이의 관계 형태는 "inactivation"으로 "PARK2"와 "CDCrel1"사이의 관계 형태를 그대로 사용하며, 중복된 노드 "PARK2"는 하나의 단백질 p_7으로 통합된다. 따라서, 단백질 p_7, $p_8 \in P(m)$이고, 관계 r=<p_7, p_8, "inactivation">은 $r \in R(m)$이다.

"Parkinson's disease" 패스웨이 역시 하나의 기능 모듈임을 유념하자. 즉, "Parkinson's disease"는 예 2에서의 m과 같은 단순한 기능 모듈들이 복합되어 구성되는 한 복잡한 기능 모듈 에 개념적으로 의미를 부여한 것으로 볼 수 있다. 따라서, 한 기능 모듈과 그 기능 모듈을 개념적으로 명명한 "Parkinson's disease"는 구별될 필요가 있기 때문에, 앞으로는 기능 모듈의 개념적 이름은 개념 기능 모듈, 또는 간단히 개념 모듈이라고 하고, 개념 모듈로 명명된 기능 모듈은 인스턴스 기능 모듈, 또는 간단히 인스턴스 모듈이라고 부르기로 한다. 다음 절에서는 개념 모듈의 복합 구조를 표현하기 위해 필요한 표현 규칙에 대해 기술하도록 한다.

3.2 개념 모듈 표현 및 평가

개념 모듈의 개념적 구조는 개념 모듈 표현 규칙에 의해 정의된다. 개념 모듈 표현 규칙, 또는 간단히, 표현 규칙은 하나의 트리플로 구성되거나, 트리플들과 이들 사이의 연산자들로 구성된 단일 규칙, 그리고 다른 규칙들의 조합으로 구성된 복합 규칙으로 구성된다. 다음 절에서는 식별자는 다르지만 동종인 단백질들이 포함되는 단백질 상호작용 관계들을 개념적으로 표현하기 위해 필요한 자료 구조인 트리플을 정의하고 이를 기반으로 표현 규칙을 이용한 개념 모듈의 표현 방식에 대해 기술하기로 한다.

3.2.1 개념 모듈 표현

개념 모듈을 표현하기 위한 표현 규칙들은 기본적으로 트리플과 이들 사이의 연산자로 구성된다. 다음에 소개되는 단백질 개념 객체는 트리플을 정의하기 위해 필요한 개념이다.

정의 4. 단백질 개념 객체 집합 $O=2^P$는 단백질 집합 P의 멱집합으로 정의되며, 단백질 스키마 *schema*(P(N)) 의 3가지 속성 값을 다음의 매핑 함수 F_O로 명시함으로써 설정된다.

$$F_O : \times_{i=1}^{3} \mathrm{dom}(A) \cup \{*\} \rightarrow O.$$

여기서, dom(A), $A \in$ {ID, NAME, GO}는 각 속성 A의 도메인을 나타내고 *는 어떤 값도 무방함을 나타낸다.

예 3. 그림 1에서, p_3.GO={"enzyme binding," "protein modification,"···}, p_4.NAME={"protein binding," "protein modification,"···}이므로 F_O(*, *, "protein modification"/GO)={p_3, p_4}=$o_1 \in O$이다. 즉, 단백질 개념 객체 $o1$은 단백질의 기능이 "protein modification"인 모든

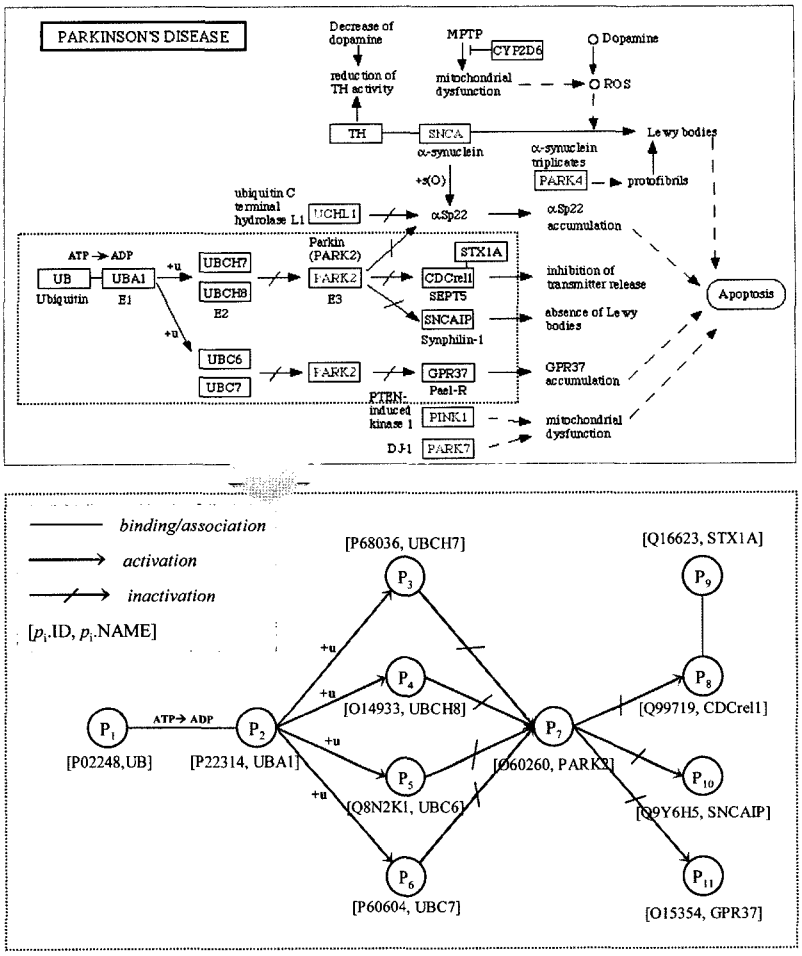


그림 1 "parkinson's disease" 패스웨이에서 한 기능 모듈 m

단백질들의 집합으로 정의된다. o_1은 단백질의 기능이 "protein modification"로 같은 단백질 p_3와 p_4를 내용으로 포함하는 하나의 개념 객체이므로, 각 두 단백질들 중 하나와 정합될 수 있는 템플릿 또는 변수로 볼 수 있다. 계속해서, p_7.ID="O60260"이므로 F_O("O60260"/ID, *, *)={p_7}=o_2이고, p_3.NAME={"UBCH7," "E2,"…}, p_4.NAME={"UBCH8," "E2,"…}이므로 F_O(*, "E2"/NAME, *)={p_3, p_4}=o_3이다. 편의상, F_O의 인자들 중 일부만 명시할 필요가 있을 경우에는 [ ] 표기를 사용하기로 한다. 예를 들어, F_O(*, "E2"/NAME, *)는 F_O["E2"/NAME]로 표기한다.

이제, 단백질 개념 객체들간의 상호작용、관계들을 개념적으로 표현하는 트리플 t는 다음과 같이 정의할 수 있다.

정의 5. o_i, $o_j \in O$는 단백질 개념 객체이고, $type_{ij} \in$ TYPE일 때 트리플 t는 다음과 같이 표현된다.

$$t = < o_i, o_j, type_{ij} >$$

예 4. 그림 1에서, 트리플 t=<o_1, o_2, "inactivation"> 와 대응되는, 또는 정합될 수 있는 관계들은 r_1=<p_3, p_7, "inactivation">, r_2=<p_4, p_7, "inactivation">이 된다.

예제 4에서, m_1=<{p_3, p_7}, {r_1}>, m_2=<{p_4, p_7}, {r_2}>는 각각 트리플 t에 정합될 수 있는 가장 작은 단위의 기능 모듈로 볼 수 있기 때문에, 트리플 t는 가장 작은 단위의 개념 모듈로 볼 수 있다. 다음 정의에서는 트리플 t에 정합될 수 있는 인스턴스 모듈을 정형화한 것이다.

정의 6. t=<o_1, o_2, $type$>와 m을 각각 하나의 트리플과 단위 기능 모듈이라고 하자. m이 다음 조건을 만족

할 때, m은 t에 정합 가능하다. 또는 t를 만족한다고 정의하고 $m \in \| t \|$로 표기한다.

$m \in \| t \| \Leftrightarrow \forall r=<p_1, p_2, type'> \in R(m)$에 대해,
$p_1 \in o_1$이고 $p_2 \in o_2$이며 $type' = type$이다.

$m \in \| t \|$일 때, m은 앞으로 m_t로 표기하고 t의 인스턴스 모듈이라고 부르기로 한다.

정의 6에서 트리플 t에 대한 단위 모듈 각각의 정합 정도에 대한 속성들에 대한 평가 함수는 3.2.3절에서 정의하는 함수 sim_{ID}, sim_{NAME}, sim_{GO}, sim_{TYPE} 들에 의해 평가된다.

예 5. 그림 2에서는 단백질 $p_{i=1,\cdots,6} \in P(N)$이고, GO (Gene Ontology) 용어들로 정의되는 단백질의 기능 $n_{j=1,\cdots,6} \in GO$으로 구성된 단백질 상호작용 네트워크 N를 보여주고 있다. 설명을 단순화하기 위해서 예제에 사용되는 단백질 개념 객체들을 $o_i = F_O[n_i/GO]$라고 가정한다.

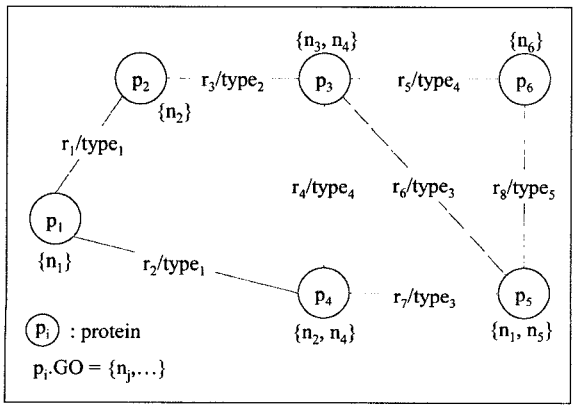
그림 2에서, 단백질 개념 객체 $o_1 = F_O[n_1/GO] = \{p_1, p_5\}$, $o_2 = F_O[n_2/GO] = \{p_2, p_4\}$일 때, 트리플 $t = <o_1, o_2, type_1>$는 인스턴스 모듈 $m^1{}_t$, $m^2{}_t \in \| t \|$으로 정합되며, 이들의 관계 집합은 각각 $R(m^1{}_t) = \{r_1\}$과 $R(m^2{}_t) = \{r_2\}$이다. 여기서, $r_7 = <p_5, p_4, type_3>$는 t의 타입 $type_1$과 다르므로 결과에서 제외된다.

개념 모듈은 하나의 트리플 t로 정의되거나 기존에 정의된 개념 모듈 사이의 연산자로 정의되므로, 모든 개념 모듈들은 트리플들과 이들 사이의 연산자들로 표현될 수 있다. 연산자는 트리플 사이의 개념적인 관계들을 이용하여 개념 모듈의 구조를 정의하며, 크게 연결 연산자와 일반화 연산자로 구분할 수 있다. 연결 연산자와 일반화 연산자는 각각 두 개념 모듈 사이의 구조적인 연결과 일반화 관계를 표현한다. 연결 연산자는 · (Arbitrariness) 와 *(Association) 그리고 일반화 연산자는 ¦ 이 있다. 연결 연산자 · 은 서로 조건 없이 여러 기능 모듈들을 포함하는 개념 모듈을 표현하기 위해 사용되며, 연결 연산자 *은 기능 모듈들이 서로 물리적으로 연결되어 결합된 개념 모듈을 표현하기 위해 사용된다. 특히, 단백질 개념 객체 o_1과 o_2에 대해 RESTRICTION=[DISTANCE(o_1, o_2) < length]로 명시될 때, "*RESCTRICTION"은 * 연산에서 기능 모듈들이 서로 간접적으로 연결되어 결합된 개념 모듈들도 표현할 수 있도록 한다. 즉, 단백질 $p_i \in o_1$과 단백질 $p_j \in o_2$ 사이에 서로 연결된 관계들로 구성된 모든 기능 모듈들 중에서 간접 연결의 길이가 length 보다 작은 모든 기능 모듈들을 포함하는 개념 모듈을 명시할 때 사용한다.

개념 모듈은 트리플 t로 정의되는 단순 개념 모듈과 여러 개념 모듈들이 연산자들에 의해 복합 되어 정의되는 복합 개념 모듈을 모두 포함하는 개념이지만, 특별히, 구별할 필요가 있는 경우를 제외하고는 앞으로 개념 모듈로 통칭하기로 한다.

정의 7. OP$\in${ · , *}가 연결 연산자일 때, 개념 모듈과 개념 모듈을 표현하는 표현 규칙은 다음과 같이 정의된다.

1. t가 트리플이면, $c \rightarrow t$는 표현 규칙이고, c는 개념 모듈이다.
2. c_1과 c_2가 개념 모듈이라면,
   $c \rightarrow c_1$ OP c_2는 표현 규칙이고, c는 개념 모듈이다.
3. c_1과 c_2가 개념 모듈이라면 *RESTRICTION에 대해,
   $c \rightarrow c_1$ *RESTRICTION c_2는 표현 규칙이고, c는 개념 모듈이다.
4. c_1과 c_2가 개념 모듈이라면 ¦ 에 대해,
   $c \rightarrow c_1$ ¦ c_2는 표현 규칙이고, c는 개념 모듈이다.
5. 모든 표현 규칙과 개념 모듈은 (1), (2), (3), (4)에 의해서만 정의된다.



P(N)	ID	NAME	GO
p_1	id_1	$name_1$, $gene_1$,...	n_1
p_2	id_2	$name_2$, $gene_2$,...	n_2
p_3	id_3	$name_3$, $gene_3$,...	n_3, n_4
p_4	id_4	$name_4$, $gene_4$,...	n_2, n_4
...	...	...	...

R(N)	Source	Dest.	TYPE
r_1	p_1	p_2	$type_1$
r_2	p_1	p_4	$type_1$
r_3	p_2	p_3	$type_2$
...	...	...	...

그림 2 단백질 상호작용 네트워크 N

정의 7의 예로, t_i, i=1,…,5가 트리플들이고, c_j, j=1,…,3이 개념 모듈들일 경우, 다음은 개념 모듈 c를 정의하는 한 표현 규칙이다.

$c \rightarrow c_1 \mid c_2$
$c_1 \rightarrow (t_1 * t_2) \cdot c_3$
$c_2 \rightarrow t_3$ *RESTRICTION t_4
$c_3 \rightarrow t_5$

표현 규칙은 기존의 개념 모듈을 표현하기 위해 사용되었던 개념 모듈들을 다시 개념적으로 조합하여 새로운 개념 모듈을 정의할 수 있으며, 분리 가능한 모듈들로 이루어진 복잡한 개념 모듈을 표현할 때 유용하다.

정의 7에서 ¦ 를 다른 연결 연산자 집합 OP와 구별하여 정의한 것을 유념해 볼 필요가 있다. ¦ 는 OP와 의미(semantics)가 다르기 때문인데, 예를 들어, $c \rightarrow c_1 \mid c_2$는 c_1 또는 c_2 둘 중 하나를 일반화하여 하나의 개념 모듈 c로 볼 수 있다는 뜻인 반면, $c \rightarrow c_1$ OP c_2는 c_1과 c_2가 OP 내 연결 연산자들에 의해 합성되어 c를 구성한다는 의미이다.

3.2.2 개념 모듈 평가

표현 규칙은 개념 모듈들의 개념적 구조를 정의하는 규칙이며, 이 규칙에 따라 정의된 개념 모듈들을 평가한 결과는 정의 8에서 보이는 바와 같이 인스턴스 모듈들의 집합으로 정의된다.

정의 8. $m \in M$이 한 인스턴스 모듈일 때, 개념 모듈 c 의 평가 결과 $\|c\|$는 다음과 같이 정의된다.

1. 표현 규칙 $c \rightarrow t = <o_1, o_2, type>$일 때,
   $m \in \|c\| \Leftrightarrow m \in \|t\|$.
2. c_1, c_2가 개념 모듈이고, 표현 규칙 $c \rightarrow c_1$ OP c_2 일 때,
   $m \in \|c\| \Leftrightarrow \|c\| = \|c_1\|$ OP $\|c_2\|$, OP$\in\{\cdot, *\}$.
3. c_1, c_2가 개념 모듈이고, 표현 규칙 $c \rightarrow c_1$ *RESTRICTION c_2일 때,
   $m \in \|c\| \Leftrightarrow \|c\| = \|c_1\|$ *RESTRICTION $\|c_2\|$.

c가 개념 모듈일 때, $m \in \|c\|$인 m은 m_c로 표기하기로 한다.

직관적으로, 개념 모듈 c 의 평가 결과 $\|c\|$는 c로 볼 수 있는 모든 인스턴스 모듈들의 집합이다. 따라서, 만약 c가 c_1과 c_2 사이에 OP, *RESTRICTION, ¦ 등을 이용해 정의된 개념 모듈이라면, c로 볼 수 있는 인스턴스 모듈 집합을 구하기 위해서는 c_1과 c_2의 평가 결과 인 $\|c_1\|$와 $\|c_2\|$사이의 각 평가 연산 방식을 정의해야 할 필요가 있다. 먼저, $c \rightarrow c_1 \cdot c_2$이 표현 규칙일 때, $\|c_1\|$과 $\|c_2\|$로 평가된 인스턴스 모듈들의 각 집합에 대한 연산 $\|c\| = \|c_1\| \cdot \|c_2\|$은 다음과 같다.

정의 9. 개념 모듈 c_1, c_2에 대해, $m_{c1} \in \|c_1\|$, $m_{c2} \in \|c_2\|$일 때, $c \rightarrow c1 \cdot c_2$에 의해 정의된 개념 모듈 c의 평가결과 $\|c\| = \|c_1\| \cdot \|c_2\|$는 다음과 같이 정의된다.

$m_c \in \|c\| \Leftrightarrow R(m_c) = \{r \mid r \in R(m_{c1})$ 또는 $r \in R(m_{c2})\}$.

예 6. 예 5에서, $t_1 = <o_1, o_2, type_1>$, $t_2 = <o_2, o_3, type_2>$이고, 표현 규칙 $c \rightarrow t_1 \cdot t_2$일 때, 개념 모듈 c는 그림 3과 같이 평가된다.

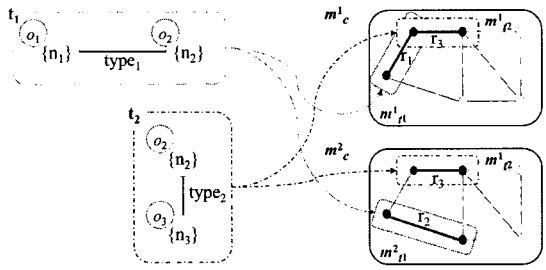


그림 3 표현 규칙 : $c \rightarrow t_1 \cdot t_2$

즉, t_1을 만족하는 인스턴스 모듈 m^1_{t1}, $m^2_{t1} \in \|t_1\|$의 관계 집합은 각각 $R(m^1_{t1})=\{r_1\}$, $R(m^2_{t1})=\{r_2\}$이고, t_2를 만족하는 모듈 $m^1_{t2} \in \|t_2\|$의 관계 집합은 $R(m^1_{t2})=\{r_3\}$이다. 따라서, 개념 모듈 c를 만족하는 인스턴스 모듈은 m^1_c, $m^2_c \in \|c\|$이며, 이들의 관계 집합은 정의 9에 따라 $R(m^1_c)=\{r_1, r_3\}$와 $R(m^2_c)=\{r_2, r_3\}$이다.

표현 규칙이 $c \rightarrow c_1 * c_2$일 때, $\|c_1\|$과 $\|c_2\|$로 평가된 인스턴스 모듈들의 집합에 대한 연산 $\|c\| = \|c_1\| * \|c_2\|$은 다음과 같다.

정의 10. 개념 모듈 c_1, c_2에 대해, $m_{c1} \in \|c_1\|$, $m_{c2} \in \|c_2\|$일 때, $c \rightarrow c_1 * c_2$에 의해 정의된 개념 모듈 c의 평가 결과 $\|c\| = \|c_1\| * \|c_2\|$는 다음과 같이 정의된다.

$m_c \in \|c\| \Leftrightarrow R(m_c)=\{r \mid r \in R(m_{c1})$ 또는 $r \in R(m_{c2})\}$, $\exists p \in P(m_c)$일 때,
여기서 $p \in P(m_{c1})$이고 $p \in P(m_{c2})$.
$= \emptyset$, 그 외.

예 7. 예 5에서, 표현 규칙 $c \rightarrow t_1 * t_2$일 때, 개념 모듈 c는 예 6의 m^1_c 만을 인스턴스 모듈로 선택한다. 그 이유는 m^1_c의 두 관계 r_1, $r_3 \in R(m^1_c)$는 단백질 p_2를 중심으로 직접 연결되어 제약 조건을 만족하지만, m^2_c의 두 관계 r_2, $r_3 \in R(m^2_c)$는 직접적으로 연결되어 있지 않

으므로 c를 만족시키지 못한다. 따라서, 그림 4와 같이 개념 모듈 c를 만족하는 인스턴스 모듈은 $m^1_c \in ||c||$이며, 이들의 관계 집합은 정의 10에 따라 $R(m^1_c)=\{r_1, r_3\}$이다.

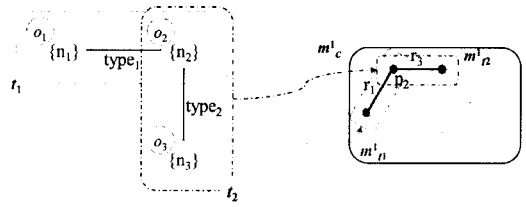


그림 4 표현 규칙 : $c \rightarrow t_1 * t_2$

다음 예는 연결 연산자, * 와 · 모두가 사용되어 정의되는 개념 모듈의 평가 과정을 설명하고 있다.

예 8. 표현 규칙 $c \rightarrow (t_1 * t_2) \cdot t_3$이고, $t_3=<o_4, o_5, type_3>$일 때, 개념 모듈 c는 그림 5와 같이 평가된다.

먼저, 예 7과 동일하게 표현 규칙 $c_1 \rightarrow t_1 * t_2$에서 개념 모듈 c_1은 $m^1_{c1} \in ||c_1||$으로 평가되며, 이들의 관계 집합은 $R(m^1_{c1})=\{r_1, r_3\}$이다. 다음으로, $c \rightarrow c_1 \cdot t_3$이고 $m^1_{c1} \in ||c_1||$과 m^1_{t3}, $m^2_{t3} \in ||t_3||$이므로, 정의 9에 따라 $||c|| = ||c_1|| \cdot ||t_3||$으로 평가된다. 즉, $R(m^1_{c1}) = \{r_1, r_3\}$, $R(m^1_{t3})=\{r_6\}$로부터 $R(m^1_c)=\{r_1, r_3, r_6\}$이고, $R(m^1_{c1})=\{r_1, r_3\}$, $R(m^2_{t3})=\{r_7\}$로부터 $R(m^2_c)=\{r_1, r_3, r_7\}$이므로, m^1_c, $m^2_c \in ||c||$이다. 그림 5는 개념 모듈 c의 전체적인 평가 과정으로, 관계들이 서로 연결된 인스턴스 모듈 m^1_c과 두 개의 분리된 부분 그래프로 구성된 모듈 m^2_c를 보여주고 있다.

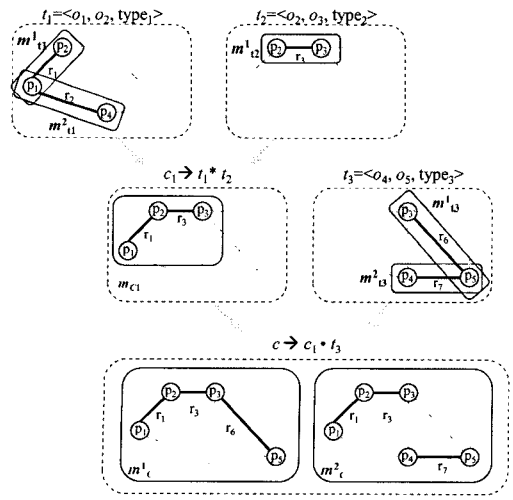


그림 5 표현 규칙 $c \rightarrow (t_1 * t_2) \cdot t_3$의 평가 과정

개념 모듈들이 여러 단계를 거쳐 간접적으로 연결되어 이루어지는 개념 모듈을 표현하고자 할 때 *RESCTRICTION이 사용된다. 예를 들어, $c \rightarrow c_1$ *RESCTRICTION c_2이 개념 모듈 c를 정의하는 표현 규칙일 때, RESCTRICTION은 c를 이루는 c_1과 c_2 사이 간접적으로 연결된 거리를 제약조건으로 명시하고 있다. 정의 12에서는 $||c_1||$과 $||c_2||$로 평가된 인스턴스 모듈들의 집합을 가지고 $||c||$를 구하는 연산 $||c|| = ||c_1||$ *RESCTRICTION $||c_2||$을 다음 정의 11을 이용해 보이도록 한다.

정의 11. $c_1, c_2, \cdots, c_{k+1}$, $k \geq 1$가 개념 모듈일 때, $c \rightarrow c_1 * c_2 * \cdots * c_{k+1}$에 의해 정의되는 개념 모듈 c는 c^*_k로 표기한다. $c \rightarrow c_1$*RESTRICTION c_2일 때, c_1과 c_2 사이 RESTRICTION을 만족하면서 *로 연결되어 이루어질 수 있는 모든 가능한 무명 개념 모듈은 c^*RESTRICTION으로 표기한다.

정의 12. 개념 모듈 c_1, c_2에 대해, $m_{c1} \in ||c_1||$, $m_{c2} \in ||c_2||$일 때, 표현 규칙 $c \rightarrow c1$ *RESCTRICTION c_2에 의해 정의된 개념 모듈 c의 평가결과 $||c||=||c_1||$ *RESCTRICTION $||c_2||$는 다음과 같이 정의된다.

$m_c \in ||c|| \Leftrightarrow R(m_c)=\{r | r \in R(m_{c1})$ 또는 $r \in R$ (m_c*RESTRICTION)
또는 $r \in R(m_{c2})\}$,
$\exists p_1, p_2 \in P(m_c$*RESTRICTION$)$일 때, 여기서 $p_1 \in P(m_{c1})$
이고 $p_2 \in P(m_{c2})$.
$R(m_c) = \emptyset$, 그 외.

예 9. 예 5에서, $t_4=<o_1, o_4, type_1>$, $t_5=<o_5, o_6, type_5>$이고, 표현 규칙 $c \rightarrow t_4$ *RESCTRICTION t_5, RESCTRICTION=[DISTANCE(o_4, o_5)<2]일 때, RESCTRICTION은 o_4과 o_5이 직접 또는 간접적으로 연결되어 있으며, 이들 사이의 거리가 2 미만임을 명시한다. 따라서, 개념 모듈 c는 정의 12에 따라 그림 6과 같이 t_4을 만족하는 인스턴스 모듈 $m^1_{t4} \in ||t_4||$의 관계 집합 $R(m^1_{t4})=\{r_2\}$, 그리고 t_5를 만족하는 모듈 $m_{t5} \in ||t_5||$의 관계 집합 $R(m^1_{t5})=\{r_8\}$을 포함하는 모듈 $m_c \in ||c||$, $R(m_c)=\{r_2, r_7, r_8\}$으로

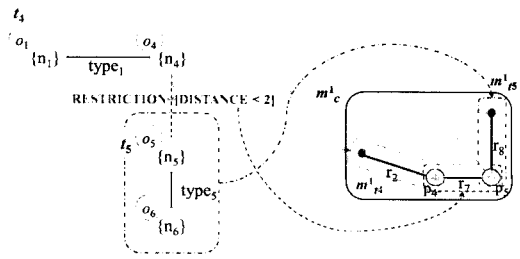


그림 6 표현 규칙 : $c \rightarrow t_4$ *RESCTRICTION t_5

평가된다. 그 이유는, 단백질 $p_4 \in o_4$와 단백질 $p_5 \in o_5$를 통해 간접적으로 연결되는, 다시 말하면, p_4, $p_5 \in P(m_c$ *RESTRICTION)인 인스턴스 모듈 m_c*RESTRICTION= m_c*_1이 중간에 존재하고 $R(m_c$*RESTRICTION)={r_7}이기 때문이다.

예 10. 예 5에서, 표현 규칙 $c \rightarrow t_1$ *RESCTRICTION t_2, RESCTRICTION=[DISTANCE(o_2, o_3) < 3]일 때, 개념 모듈 c에 의해 평가되는 인스턴스 모듈은 중간 기능 모듈들을 모두 포함한다. 이때, 중간 기능 모듈은 그림 7과 같이 t_1 또는 t_2를 각각 만족하는 두 기능 모듈들 사이에 여러 단계를 거쳐 간접적으로 연결하는 인스턴스 모듈들이다. 여기서, 제약조건은 o_2와 o_3사이에 중간에 경유할 수 있는 단백질 노드들의 수가 3미만이라는 것을 의미한다.

먼저, 기능 모듈 m^1_{t1}의 한 관계 r_1=<p_1, p_2, $type_1$>$\in R(m^1_{t1})$과 기능 모듈 m^1_{t2}의 한 관계 r_3=<p_2, p_3, $type_2$>$\in R(m^1_{t2})$는 단백질 p_2를 서로 공유하면서 직접 연결되어 있으므로 m_{c0}*RESTRICTION=$m_{c0}*_0$=ø이다. 따라서, $m^1_c \in ||c||$이고, $R(m^1_c)$={r_1, r_3}이다. 그러나, r_2=<p_1, p_4, $type_1$>$\in R(m^2_{t1})$와 r_3=<p_2, p_3, $type_2$>$\in R(m^1_{t2})$는 서로 연결되어 있지 않으므로 간접적으로 연결을 가능하게 하는 m_c*RESTRICTION를 추가적으로 찾아본다. 여기서, 단백질 개념 객체 o_2={p_2, p_4}으로, o_3={p_3}로 평가 되므로 o_2에서는 단백질 p_4만을 선택하고, o_3에서는 단백질 p_3를 사용하여 m_c*RESTRICTION를 탐색한다. 즉, 정의된 제약조건을 만족하는 기능 모듈 m_c*RESTRICTION을 실제 탐색할 때는 RESCTRICTION=[DISTANCE(p_4, p_3) < 3]와 같이 실제 단백질들을 이용하여 처리된다. 그림 7과 같이 관계 r_2, r_3사이에서 정의된 제약조건을 만족하는 두 개의 인스턴스 모듈 m_{c1}*RESTRICTION=$m_{c1}*_1$과 m_{c2} *RESTRICTION = $m_{c2}*_2$가 존재해서 $R(m_{c1}$*RESTRICTION)={r_4}, $R(m_{c2}$*RESTRICTION)={r_7, r_6}가 된다. 즉 {r_4}를 경유하거나, {r_7, r_6}를 경유하여 두 모듈이 연결될 수 있다. 따라서, m^1_c, m^2_c, $m^3_c \in ||c||$이며, 이들의 관계 집합은 정의 12에 따라 $R(m^1_c)$={r_1, r_3}, $R(m^2_c)$={r_2, r_4, r_3}, $R(m^3_c)$={r_2, r_7, r_6, r_3}이다.

다음으로, 표현 규칙이 $c \rightarrow c_1 \mid c_2$일 때, 개념 모듈 c는 c_1 또는 c_2가 될 수 있는 기능 모듈들이 인스턴스 모듈들로 정합되기 때문에, $||c||$는 다음과 같이 ∪를 써서 정의할 수 있다.

정의 13. c_1, c_2가 개념 모듈이고, 표현 규칙이 $c \rightarrow c_1 \mid c_2$ 라고 하자. 만약, $m_{c1} \in ||c_1||$, $m_{c2} \in ||c_2||$라면, $||c|| = ||c_1|| \cup ||c_2||$로 정의한다.

예 11. 예 5에서, 표현 규칙 $c \rightarrow t_1 \mid t_2$, $c_1 \rightarrow t_1$, $c_2 \rightarrow t_2$에 대해서, m^1_{c1}, $m^2_{c1} \in ||c_1||$이고, $m^1_{c2} \in ||c_2||$라고 가정하자. 그러면, 개념 모듈 c를 만족하는 인스턴스 모듈 집합 $||c|| = ||c_1|| \cup ||c_2|| \{m^1_{c1}, m^2_{c1}, m^1_{c2}\}$이다.

마지막으로, "Parkinson's disease" 패스웨이를 표현하기 위한 개념 모듈 표현 규칙을 정의하였다. 그림 1에서 "Parkinson's disease"를 표현하기 위한 규칙은 "Inhibition of transmitter release," "Absence of lewy body," "GPR37 accumulation"등의 개념 모듈들 간 조

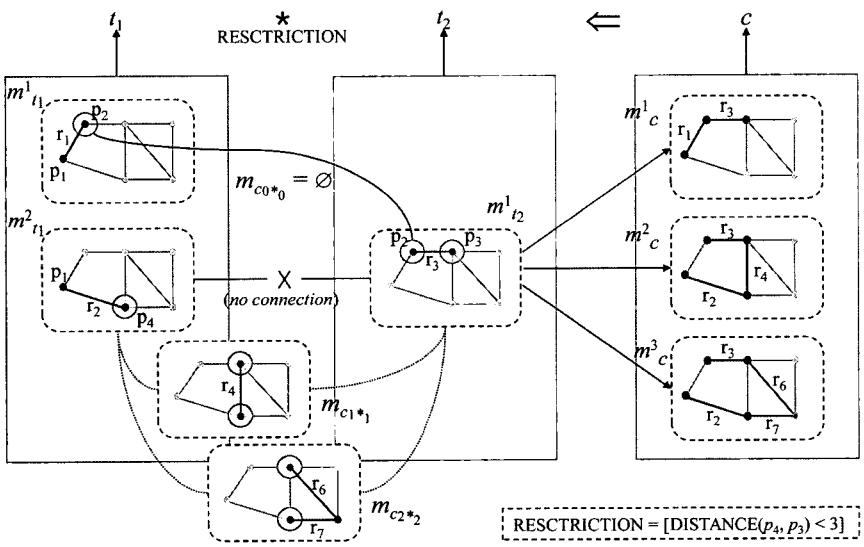


그림 7 t_1 *RESTRICTION t_2 처리 과정

합으로 구성될 수 있다. 따라서, "Parkinson's disease"을 정의하기 위한 표현 규칙은 다음과 같다. 편의상, "Parkinson's disease"의 표현 규칙에서 사용되는 단백질 개념 객체들은 $o_{p.NAME}$=Fo[p.NAME/NAME] 라고 표기한다. 예를 들어, t=<o_{UB}, o_{UBA1}, "binding">에서 o_{UB}=Fo["UB"/NAME], o_{UBA1}=Fo["UBA1"/NAME, n_2/GO]이다.

Inhibition of transmitter release →
<o_{UB}, o_{UBA1}, 'binding'> * <o_{UBA1}, $o_{\{UBCH7,\ UBCH8\}}$, 'activation'>
* <o=o_{UBCH7} ∪ o_{UBCH8}, o_{PARK2}, 'inactivation'>
* <o_{PARK2}, $o_{CDCrel1}$, 'inactivation'>
Absence of lewy body →
... * <$o_{\{UBCH7,\ UBCH8\}}$, o_{PARK2}, 'inactivation'> *
<o_{PARK2}, o_{SNCAIP}, 'inactivation'>
GPR37 accumulation →
<o_{UB}, o_{UBA1}, 'binding'> * <o_{UBA1}, $o_{\{UBCH6,\ UBCH7\}}$, 'activation'> * <o, o_{PARK2}, 'inactivation'>
* <o_{PARK2}, o_{GPR37}, 'inactivation'>
...
Parkinson's disease → *Inhibition of transmitter release* | *Absence of lewy body* |
GPR37 accumulation ...

개념적으로 정의된 유용한 표현 규칙들은 XML 형태로 지식베이스에 체계적으로 관리되며, 구축된 지식베이스를 다른 네트워크에 적용하면 정의된 규칙들과 개념적으로 유사하며 구조적으로 일치하는 개념 모듈을 탐색하여 예측할 수 있다.

3.2.3 유사도를 고려한 개념 모듈 평가

사전에 정의된 부 개념 모듈들의 조합으로 구성된 개념 모듈은 포함된 부 개념 모듈들을 확장하여 변환한 다음 평가된다. 즉, 모든 개념 모듈은 $c \rightarrow t$와 같이 트리플로 표현되거나 $c \rightarrow c_1$ OP c_2와 같이 부 개념 모듈들간의 연산자 조합으로 표현되기 때문에, 개념 모듈 c를 인스턴스 모듈 m_c가 만족하는 정도 SIM(c, m_c)는 다음 정의 14와 같이 평가된다.

정의 14. 개념 모듈 c의 부 개념 모듈 c_1, c_2에 대해, $m_{c1} \in ||c_1||$, $m_{c2} \in ||c_2||$일 때, c를 인스턴스 모듈 m_c가 만족하는 정도 $0 \le SIM(c, m_c) \le 1$은 다음과 같이 정의된다.

SIM(c, m_c) = $sim_T(t, m_t)$, $c \rightarrow t$ 일 때.
= min(SIM(c_1, m_{c1}), SIM(c_2, m_{c2})),
$c \rightarrow c_1$ OP c_2 일 때.
= max(SIM(c_1, m_{c1}), SIM(c_2, m_{c2})), $c \rightarrow c_1$ | c_2일 때.

정의 15. 개념 모듈 c, 인스턴스 모듈 m_c에 대해 SIM(c, m_c)가 주어져 있을 때, $||c||_a$, $0 \le a \le 1$는 다음과 같이 정의된다.

$$m_c \in ||c||_a,\ SIM(c, m_c) \ge a.$$

트리플로 표현된 각 개념 모듈은 모두 하나의 관계만을 가지기 때문에, 각각의 관계는 트리플과 개념적으로 유사한 의미를 가진다. 따라서, 트리플을 평가하기 위해서는 먼저 다음 정의에서와 같이 트리플과 관계 사이의 유사 정도를 계산해야 한다.

정의 16. 트리플 t=<o_1, o_2, $type_1$>과 관계 r=<p_1, p_2, $type_2$>의 사이의 유사 정도 $sim_T(t, r)$는 단백질 개념 객체 o_1, o_2와 단백질 p_1, p_2사이 각각의 유사 정도, 그리고 $type_1$과 $type_2$의 유사 정도에 의해 다음과 같이 평가된다.

$$sim_T(t, r) = w \times min(sim_P(o_1, p_1), sim_P(o_2, p_2)) + (1-w) \times sim_{TYPE}(type_1, type_2).$$

여기서, $sim_P(o_1, p_1)$, $sim_P(o_2, p_2)$는 단백질 개념 객체와 단백질 사이의 유사 정도, $sim_{TYPE}(type_1, type_2)$는 트리플과 관계의 형태에 대한 유사 정도를 평가하는 함수이다. 또한, w와 $(1-w)$는 두 유사 정도가 $sim_T(t, r)$에 영향을 주는 가중치이다.

정의 17. 단백질 개념 객체 o와 단백질 p 사이의 유사 정도 $sim_P(o, p)$는 다음과 같이 평가된다.

$$sim_P(o,p) = min(sim_{ID}(o,p),\ sim_{NAME}(o,p),\ sim_{GO}(o,p)).$$
$$sim_{ID}(o,p) = 1,\ p \in Fo[p.ID/ID] = o \text{이거나 ID} = \text{'*'일 때}$$
$$sim_{NAME}(o,p) = 1,\ p \in Fo[p.NAME/NAME] = o \text{이거나 NAME} = \text{'*'일 때}$$
$$sim_{GO}(o,p) = min(max(GO(n_i, n_j))),\ n_i \in p.GO \text{와 } n_j \in p.GO \text{에 대해서}$$
$$p \in Fo[p.GO/GO] = o \text{이거나 GO} = \text{'*'일 때.}$$

여기서, 단백질 개념 객체와 단백질의 유사 정도는 이들의 식별자(ID) 또는 이름(NAME)이 같으면 동일한 것으로 평가되며, 그렇지 않은 경우 각각의 GO 용어들 사이의 개념적 거리에 따라 평가된다. 즉, 온톨로지 계층에서 두 용어 n_i와 n_j 사이의 거리가 $DIST(n_i, n_j)$일 때 두 용어의 개념적 거리는 $GO(n_i, n_j)$=e-0.3*$DIST$(n_i, n_j)와 같이 계산될 수 있다. 따라서, 평가할 개념 모듈 객체가 GO 용어들로 정의된 경우에는 정확하게 일치하는 GO 용어들뿐만 아니라 개념적으로 유사한 GO 용어를 가지는 하나 이상의 단백질들로 평가될 수 있다.

예 12. 다음의 단백질 개념 객체 o_i, i=1...,3에 대해, 단백질들과의 유사 정도 sim_P를 구해보자.

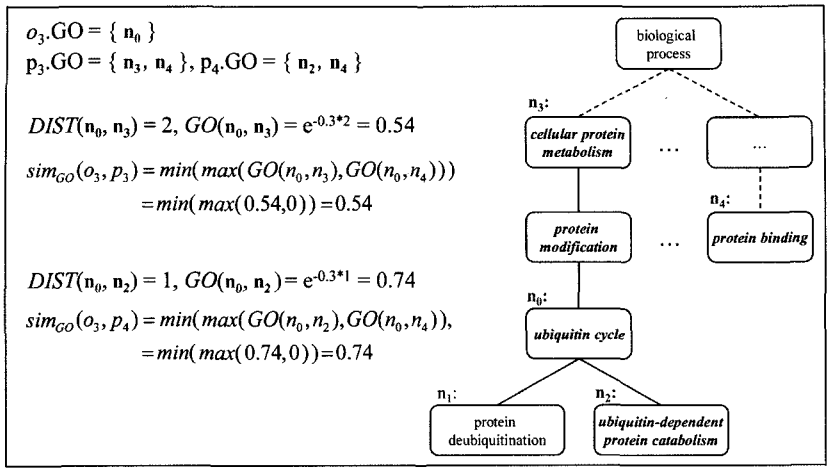


그림 8 GO의 일부 계층구조 및 개념 거리 계산

o_1 = F_O["name$_1$"/NAME].

o_2 = F_O["n_2"/GO].

o_3 = F_O["n_0"/GO].

개념 객체 $o1$의 경우, p_1.NAME={"name$_1$", "gene$_1$"} 라면, $sim_{NAME}(o_1, p_1)$=1이며, $sim_P(o_1, p_1)$=1로 평가된다. F_O["n_2"/GO]와 같이 GO 용어로 정의된 개념 객체 o_2의 경우, p_2.GO={n_2}이면, $sim_{GO}(o_2, p_2)$=1이며, $sim_P(o_2, p_2)$=1로 평가된다. 정확하게 일치하지 않지만 개념적으로 유사한 GO 용어를 가지는 단백질들도 정합될 수 있는 데, F_O["n_0"/GO]으로 정의된 개념 객체 o_3의 경우, "n_0"와 개념적으로 유사한 GO 용어들을 정합시키는 과정은 그림 8에서 GO 데이타베이스에 정의된 "Biological Process(BP)"의 일부 계층도를 이용하여 도식화하였다. $sim_{GO}(o_3, p_3)$=0.54, $sim_{GO}(o_3, p_4)$=0.74이므로, 개념 객체 o_3는 p_3, p_4와 $sim_P(o_3, p_3)$=0.54, $sim_P(o_3, p_4)$=0.74 정도로 개념적으로 유사하다고 평가된다.

정의 18. 두 관계 형태의 관련 정도 $sim_{TYPE}(type_1, type_2)$는 다음과 같이 $type_1$과 $type_2$이 일치하면 1, 일치하지 않지만 같은 분류에 속하면 ω로 평가된다.

여기서, CAT($type$)은 $type$이 속한 분류 용어를 나타내며, 분류 용어에 소속될 정도 값에 대한 정량화된 수치 값은 도메인에 따라 정의될 수 있다. 관계 형태에 대한 분류 용어들은 다음 표 2와 같이 GENIA[21]에서 정의한 목록을 수정하여 사용하였다.

표현 규칙 $c \rightarrow t$일 때, 개념 모듈 c를 만족하는 인스턴스 모듈 m_t 와의 유사 정도 SIM(c, m_t)는 트리플 t와 관계 r의 유사 정도 $sim_T(t, r)$로 평가된다.

예 13. 표현 규칙 $c \rightarrow t_2$, t_2=<o_2, o_3, type$_2$>이고, 평가된 인스턴스 모듈 $m_c \in ||c||$, R(m_c) = {r_3}, r_3=<p_2, p_3, type$_2$>일 때, 유사 정도 SIM(c, m_c)는 예 12에서 평가된 단백질 개념 객체들을 이용하여 다음과 같이 $sim_T(t_2, r_3)$로 계산된다.

$$sim_T(t_2, r_3) = 0.7 \times min(sim_P(o_2, p_3), sim_P(o_3, p_3)) + 0.3 \times sim_{TYPE}(type_2, type_2)$$
$$= 0.7 \times min(1, 0.54) + 0.3 \times 1 = 0.678.$$

다음은 개념 모듈과 인스턴스 모듈 사이의 유사 정도를 계산하는 예이다.

예 14. 표현 규칙 $c_1 \rightarrow t_1 * t_2$, $c \rightarrow c_1 \cdot t_3$이고, 개념 모듈에 사용된 각각의 트리플들과 관계들 사이의 유사 정도를 다음과 같다고 하자.

$sim_T(t_1, r_1)$ = 1.

$sim_T(t_2, r_3)$ = 0.678.

$sim_T(t_3, r_6)$ = 0.85.

평가된 인스턴스 모듈 $m_c \in ||c||$, R(m_c)={r_1, r_3, r_6}일 때, 개념 모듈 c와 탐색된 인스턴스 모듈 m_c 사이의 유

표 2 분류된 관계 목록

$sim_{TYPE}(type_1, type_2)$ = 1, $type_1$ = $type_2$일 때. = ω, $0 \le \omega \le 1$, $CAT(type_1)$ = $CAT(type_2)$일 때.	CAT	Terms
	influence	*effect, affect, role, response, …*
	regulation	*mediate, regulate, regulation, …*
	activation	*induce, activate, activation, …*
	…	…

사 정도 SIM(c, m_c)는 다음과 같이 계산된다.

$$\text{SIM}(c, m_c) = min(\text{SIM}(c_1, m_{c1}), \text{sim}_T(t_3, r_3))$$
$$= min(min(\text{sim}_T(t_1, r_1), \text{sim}_T(t_2, r_3)), \text{sim}_T(t_3, r_6))$$
$$= min(1, 0.678, 0.85) = 0.678.$$

3.3 모듈 탐색 질의 평가

단백질 상호작용 네트워크 N 상에서 질의를 평가하기 위해서는 각각의 트리플과 모든 관계 사이의 관련 정도를 매번 평가해야 한다. 특히, 온톨로지 용어들이 이용될 경우 매우 많은 계산 시간이 요구된다. 이런 단점을 해결하기 위해 네트워크에 포함된 모든 단백질들은 표현 규칙에 포함된 모든 단백질 개념 객체들에 의해 미리 색인되고, 각각의 관계들은 표현 규칙에 포함된 트리플들에 의해 다시 색인되도록 한다. 그림 9는 이 색인 과정을 설명하고 있다.

즉, N에서 규칙에 포함된 각각의 단백질 개념 객체 o_1과 o_2에 대해 이와 유사한 단백질 p_1/1.0, p_5/0.5 그리고 p_2/0.7, p_4/0.8를 각각 색인 한다. 또한, 트리플 t_1에 대해 이와 유사한 관계 r_1/1.0, r_2/0.86을 색인해서 트리플 t_1에 유사한 관계 r_1과 r_2를 개념 모듈 평가 과정에서 바로 참조할 수 있도록 한다.

계속해서, 그림 10은 위의 색인 구조를 참조하여 표현 규칙에 의해 정의된 개념 모듈들을 평가하는 전체적인 절차를 설명하기 위한 예이다. 먼저, 개념 모듈 $c_1 \rightarrow t_1 * t_2$는 미리 색인된 트리플 t_1, t_2에 해당되는 인스턴스 모듈들을 사용하여 평가되며, 평가된 m^1_{c1}={<{p_1, p_2, p_3}, {r_1, r_3}>}∈||c_1||은 개념 모듈 색인데이타에 저장된다. 마찬가지로, 다른 개념 모듈을 포함하고 있는 개념 모듈 $c_2 \rightarrow c_1 \cdot t_3$는 미리 색인된 개념 모듈 c_1에 해당되는 인스턴스 모듈 m^1_{c1}∈||c_1||과 색인된 트리플 t_3에 해당되는 인스턴스 모듈 m^1_{t3}=<{p_3, p_5}, {r_6}>, m^2_{t3}=<{p_4, p_5}, {r_7}>∈||t_3||와의 연산을 바로 수행하여 인스턴스 모듈 m^1_{c2}, m^2_{c2}∈||c_2||을 얻을 수 있으며, 각각의 관계 집합은 R(m^1_{c2})={r_1, r_3, r_6}, R(m^2_{c2})={r_1, r_3, r_7}이다.

개념 모듈을 색인 할 때는 개념 모듈에 대한 XML로 정의된 데이타를 포함한다. 그 이유는 색인된 개념 모듈에 해당 되는 인스턴스 모듈이 현재 탐색 대상이 되는 단백질 상호작용 네트워크와 서로 다른 종이거나 다른 종류의 네트워크인 경우 단백질 데이타가 서로 다를 수 있으므로 정확히 일치하지 않을 경우에는 개념 모듈이 재 평가되어야 되기 때문이다.

다음은 그림 10과 같은 색인 구조를 기반으로 네트워크 N 상에서 사용자가 원하는 모든 기능 모듈들을 탐색하는 질의 처리 과정에 대해 기술한다. 질의 처리 과정에서는 N 내의 기능 모듈들이 어느 정도로 질의 내의 개념 모듈들을 만족하는 가에 대한 정도 값도 고려되어 질의가 평가된다.

정의 19. 기능 모듈 탐색 질의, 또는 간단하게 탐색 질의 Q는 다음과 같이 정의된다.

1. c가 개념 모듈이면 c는 탐색 질의 Q이다.
2. Q_1과 Q_2가 탐색 질의이면 $Q_1 \vee Q_2$와 $Q_1 \wedge Q_2$는 탐색 질의 Q이다.
3. 다른 것은 탐색 질의가 아니다.

탐색 질의 Q에서 ∨(Logical OR) 연산자는 인스턴스 모듈들의 두 집합에 대한 합집합(∪)으로, ∧(Logical AND)는 모듈들의 두 집합에 대한 교집합(∩)으로 평가된다. 다음은 이를 위한 정의이다.

정의 20. Q가 탐색 질의일 때, Q의 질의 평가 결과 ||Q||는 다음과 같이 정의된다.

1. ||Q|| = ||c||, Q = c일 때.
2. ||Q|| = ||Q_1|| ∪ ||Q_2||, Q = $Q_1 \vee Q_2$일 때.

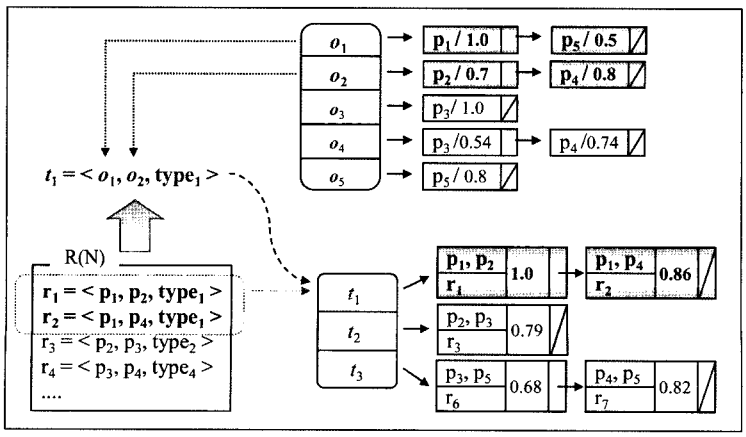


그림 9 트리플 평가 과정

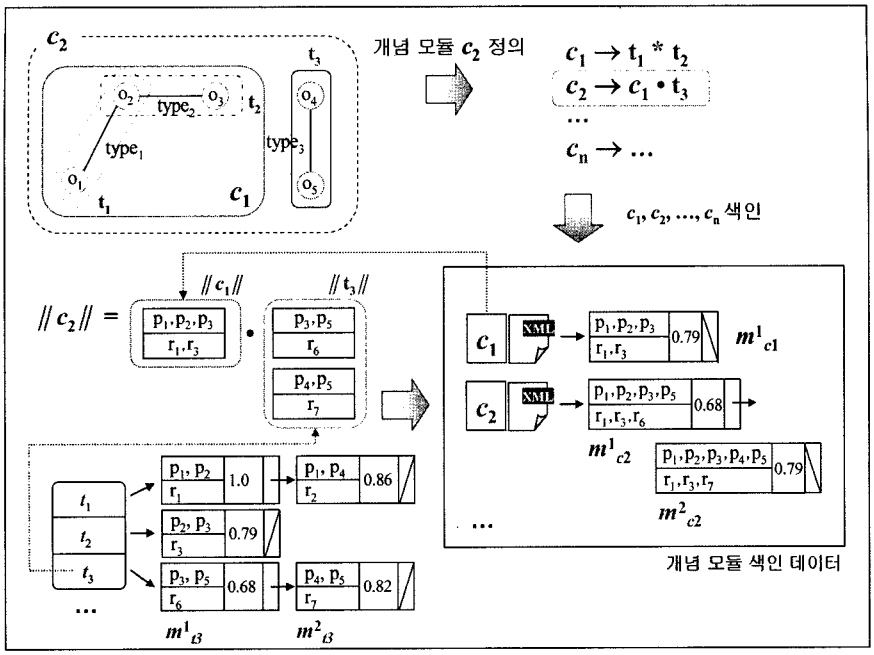


그림 10 개념 모듈 평가 과정

3. $||Q|| = ||Q_1|| \cap ||Q_2||$, $Q = Q_1 \wedge Q_2$일 때.

예를 들어, $R(m_1)=\{r_1, r_2\}$, $R(m_2)=\{r_1, r_3\}$, $R(m_3)=\{r_3, r_4\}$이고, $||Q_1||=\{m_1, m_2\}$, $||Q_2||=\{m_1, m_3\}$일 때, $Q=Q_1 \vee Q_2$의 경우에 $||Q||=\{m_1, m_2, m_3\}$가 되고, $Q=Q_1 \wedge Q_2$의 경우에 $||Q||=\{m_1\}$이 된다.

탐색 질의 Q를 만족하는 기능 모듈 m_Q와의 관련 정도 $SIM(Q, m_Q)$는 합집합($\cup$)의 경우 max를, 교집합($\cap$)의 경우 min연산을 통해 그 정도 값을 계산하게 된다. 정의 21은 그 계산 과정을 보여주고 있다.

정의 21. 탐색 질의 Q를 인스턴스 모듈 m_Q가 만족하는 정도 $0 \le SIM(Q, m_Q) \le 1$은 다음과 같이 정의된다.

$SIM(Q, m_Q) = SIM(c, m_c)$, $Q = c$일 때.

$= max(SIM(Q_1, m_{Q1}), SIM(Q_2, m_{Q2}))$, $Q=Q_1 \vee Q_2$일 때.

$= min(SIM(Q_1, m_{Q1}), SIM(Q_2, m_{Q2}))$, $Q=Q_1 \wedge Q_2$일 때.

이제, 유사도를 고려한 질의 평가 결과를 정의해 보면 다음과 같다.

정의 22. 탐색 질의 Q, 인스턴스 모듈 m_Q에 대해 $SIM(Q, m_Q) \ge a$일 때, $||Q||_a$, $0 \le a \le 1$는 다음과 같이 정의된다.

$$m_Q \in ||Q||_a,\ SIM(Q, m_Q) \ge a$$

마지막으로, 정의 23은 탐색 질의 Q의 평가 결과를 일반화한 것이다.

정의 23. $Q = \vee_{i=1}^{n}(\wedge_{j=1}^{m} Q_{ij})$, Q_{ij}가 탐색 질의이고 $SIM(Q, m_Q) \ge a \ge 0$, $SIM(Q_{ij}, m_{Qij}) \ge a_j \ge 0$일 때, 질의 평가 결과 $||Q||_a$ 는 다음과 같다.

$$||Q||_\alpha = \cup_{i=1}^{n}(\cap_{j=1}^{m} ||Q_{ij}||_{\alpha_j}),$$

여기서 $a = \max_{i=1}^{n} \alpha_i$ 이고 $a_i = \min_{j=1}^{m} \alpha_j$.

4. 설계 및 구현

본 논문에서 제안한 개념 기반 기능 모듈 탐색 시스템은 Java와 Oracle DB를 사용하여 구현되었다. 이 시스템은 크게 단백질 상호작용 네트워크를 통합하는 부분, 개념 모듈을 정의하는 부분 그리고 기능 모듈을 탐색하여 가시화하는 부분으로 구성된다.

먼저, 단백질 상호작용 네트워크는 BIND[6], DIP[7], MINT[8] 등 웹을 통해 공개된 단백질 상호작용 관계 데이타들을 수집하여 구축된다. 단백질 상호작용 네트워크 구축에 사용된 단백질 상호작용 관계 데이타들은 대부분 단백질 데이타베이스 식별자 또는 유전자 이름으로 구성되는 단순한 이진관계 형태로 공개되므로 단백질의 기능을 포함한 상세한 특징 정보들을 사용할 수 있도록 SWISS-PROT 데이타베이스[18]와 자동으로 동기화한다. 이렇게 상세한 정보들을 가지고 있는 단백질들과 상호작용 관계들로 구성된 네트워크들은 통합 데이타베이스에 저장되며 기능 모듈 탐색의 대상이 된다.

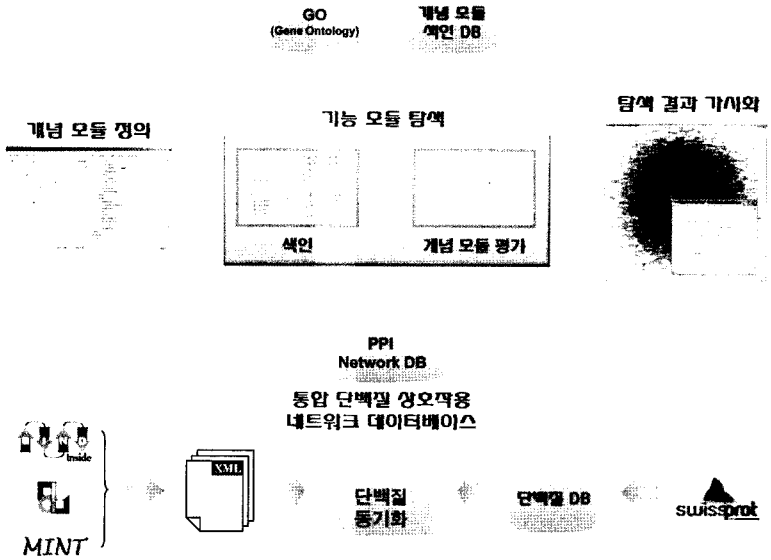


그림 11 개념 기반 기능 모듈 탐색 시스템

다음으로, 기능 모듈 탐색에서 개념 모듈 표현 규칙은 개념 모듈 정의 인터페이스를 통해 다양한 형태로 표현되며, 기능 모듈 탐색 과정인 색인 및 평가를 수행하여 정의된 개념 모듈과 일치하는 기능 모듈들을 탐색한다. 마지막으로, 탐색 대상인 단백질 상호작용 네트워크 및 탐색된 기능 모듈들을 분석하기 쉽도록 자동으로 단백질과 상호작용 관계들을 최적화하여 배치시켜 탐색 결과를 가시화한다. 위의 그림 11은 이러한 기능 모듈 탐색 시스템의 전체적인 구성을 도식화하였다.

본 시스템에서는 사용자가 관심 있는 개념 모듈을 정의하는 표현 규칙을 세가지 방법으로 정의할 수 있도록 구현하였다. 첫째, 위에서 설명한 규칙 정의와 동일하게 텍스트 형식으로 규칙을 정의할 수 있다. 둘째, 표현 규칙 내 트리플들을 노드과 링크로 표현하여 시각화된 화면에서 규칙을 작성할 수 있다. 여기서, 연산자들은 그래프 표현 방법과 유사하게 표현하였다. 마지막으로, 사용자는 모든 표현 규칙을 XML 형태로 정의할 수 있다. 텍스트 형태나 시각화된 형태의 규칙들도 탐색 전에는 최종적으로 XML 형태로 자동으로 변환되어 처리된다.

그림 12는 단백질 식별자(ID), 이름(Name), 유전자

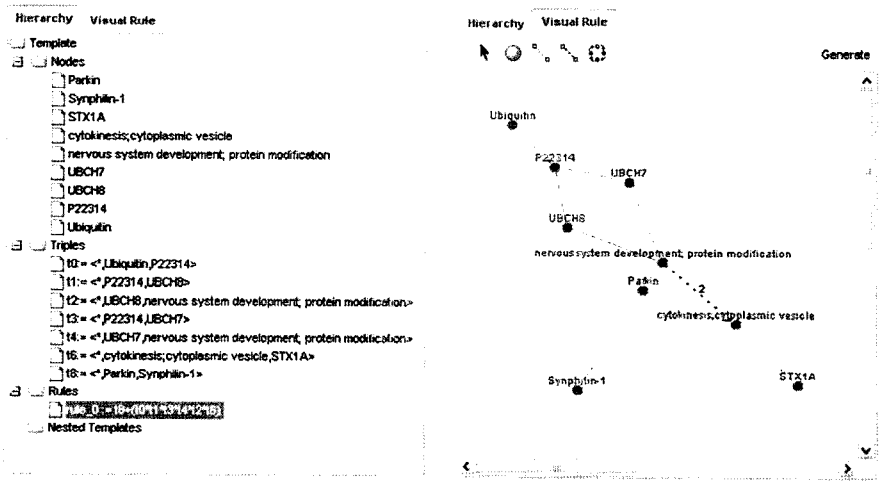


그림 12 개념 모듈 정의 인터페이스

(Gene), 생물학적 기능(GO)등과 같은 단백질의 다양한 특징 정보들을 사용하여 "Parkinson's disease" 패스웨이의 일부분을 개념 모듈 표현 규칙으로 정의한 화면이다. 왼쪽 화면은 텍스트 형태의 정의 인터페이스로 개념 모듈을 구성하는 개념 노드를 정의하고, 정의된 개념 노드를 이용하여 트리플을 정의한 다음, 정의된 트리플들 사이의 연산자를 적용하여 전체적인 개념 모듈의 구조를 정의한다. 오른쪽 화면은 시각화된 정의 인터페이스로 노드는 개념 노드를 나타내며, 노드들 사이의 실선은 트리플을 표현한다. 서로 연결된 실선들은 트리플들 사이의 * 연산을 의미하며, 두 노드들 사이의 점선은 경로 제약조건을 가지는 * 연산을 의미한다. 서로 연결되지 않은 그래프들은 그래프들 사이의 · 연산을 표현한다.

그림 13은 이러한 개념 모듈 표현 규칙을 정의하고

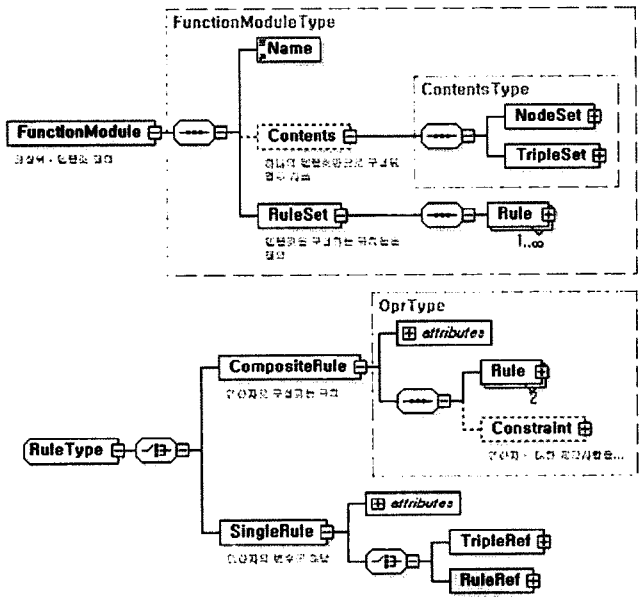


그림 13 개념 모듈 표현 규칙을 위한 XML 스키마

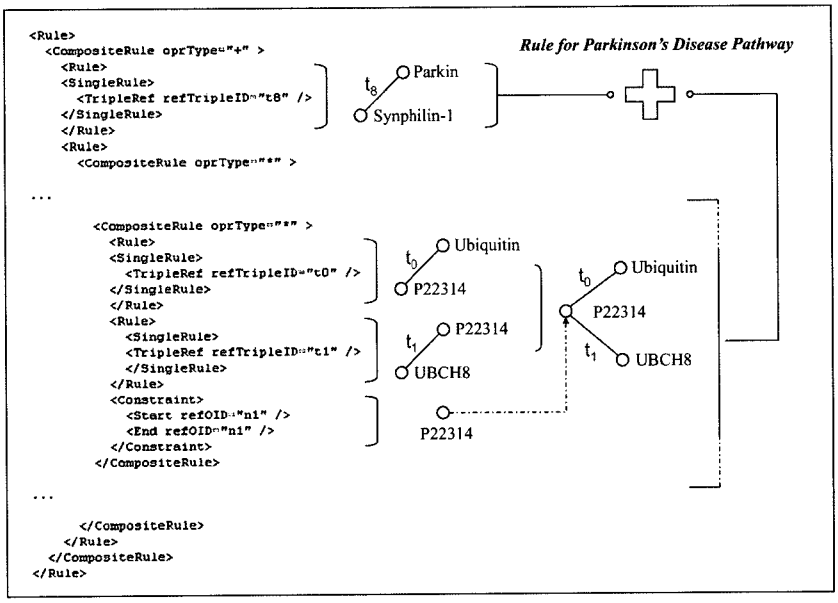


그림 14 트리플들 사이의 연산자로 표현된 규칙의 XML 표현

지식베이스로 저장하기 위한 XML 스키마이다. 여기서, "Contents" 부분은 단백질의 다양한 특성들로 정의된 개념 노드들로 구성된 "NodeSet"과 정의된 개념 노드들 사이의 관계를 정의한 트리플을 저장하기 위한 "TripleSet"으로 구성된다. 트리플들 사이의 연산자들로 표현된 개념 모듈의 전체적인 구조는 "RuleSet"에 저장되며 하나의 규칙("RuleType")은 "SingleRule"이나 "CompositeRule"로 표현된다. 여기서, "SingleRule"은 트리플이나 기존에 정의된 규칙들로 규칙을 참조하기 위해 사용되며, "CompositeRule"은 기존에 정의된 규칙들 사이에 연산자들을 사용하여 규칙을 정의하기 위해 필요하다. 그림 14는 그림 12에서 정의한 "Parkinson's disease" 패스웨이에 대한 XML 형식의 개념 모듈 표현 규칙의 일부이며, 트리플들 사이의 연산자로 구성된 규칙 부분을 표현하고 있다. 연산자들 사이의 우선 순위를 XML로 표현하기 위해 전위 표현식(Preorder Expression) 형태로 트리플 사이의 연산자로 구성된 규칙을 표현하였다.

실질적인 예로서, 그림 15는 Yeast Protein Complex Database[19]의 6번째 단백질 복합체를 DIP[7]에서 구축한 "Yeast"에 대한 단백질 상호작용 네트워크에서 탐색하는 과정을 도식화하였다. 기능 모듈 탐색에 사용된 "Yeast"에 대한 단백질 상호작용 네트워크는 4529개의 단백질과 16380개의 상호작용 관계들로 구성되어 있으며, 탐색하려는 단백질 복합체는 유전자 이름이 "SCP160", "GCD6", "CKA1", "ASC1"인 4개의 단백질로 구성되어 있다. 개념 모듈 표현 규칙은 4개의 유전자 이름으로 각각 정의된 개념 노드들로 구성되며, 실험 과정에서 "SCP160"이 "bait"로 사용되었으므로 "SCP160"을 중심으로 방사형 구조일 것으로 추측하여 규칙을 표현하였다. 전체 대상 단백질 상호작용 네트워크에서 탐색된 기능 모듈은 하나이며, 대상 단백질 상호작용 네트워크에 상호작용 관계 타입들이 정의되어 있었다면 기능 모듈을 구성하는 단백질들 사이의 구체적인 상호작용 관계를 파악할 수 있다. 좀 더 자세히 살펴보기 위해 탐색된 기능 모듈을 새로운 창에 가시화하거나 탐색된 기능 모듈을 중심으로 원하는 거리만큼 상호작용하는 이웃 단백질과 상호작용 관계들을 포함하여 새로운 창에 가시화할 수 있다.

그림 16에서는 그림 15의 예제를 좀 더 확장하여 "n4", "n5" 개념 노드를 추가하였다. 여기서, "n4" 개념 노드에는 "Molecular Function"에 해당 되는 기능으로 "casein kinase activity"라는 기능을 정의하였고, "n5" 개념 노드에는 "Biological Process"에 해당되는 기능으로 "DNA Repair"라는 기능을, "Celluar Component"에 해당되는 기능으로 "nucleus"라는 세포 내의 위치를 정의하였다. 정의된 개념 노드들을 이용하여 "CKA1"와 "n4", "n4"와 "n5"을 트리플로 정의하여 그림 16과 같이 추가적으로

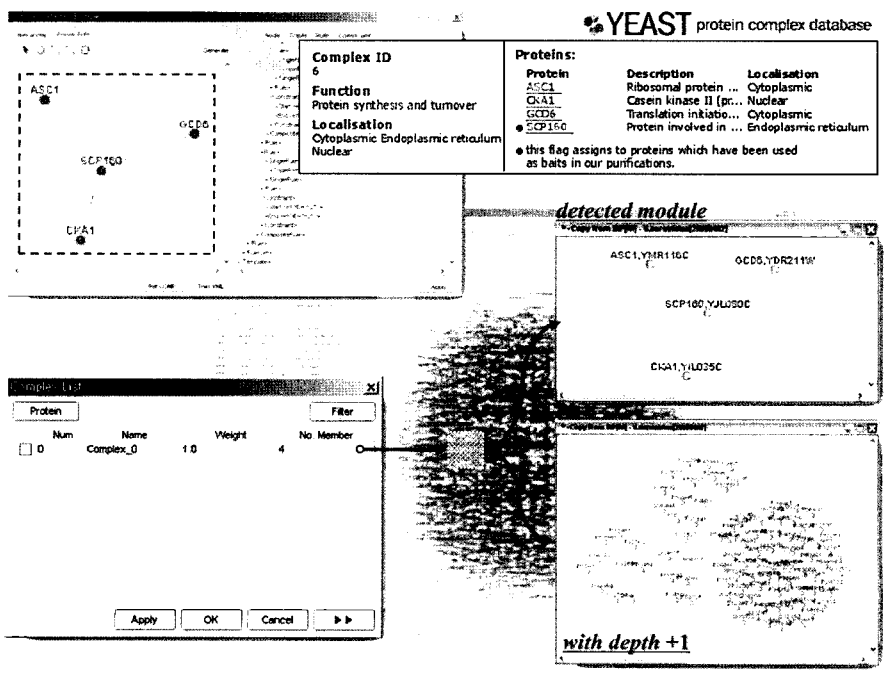


그림 15 Yeast Protein Complex 탐색 과정

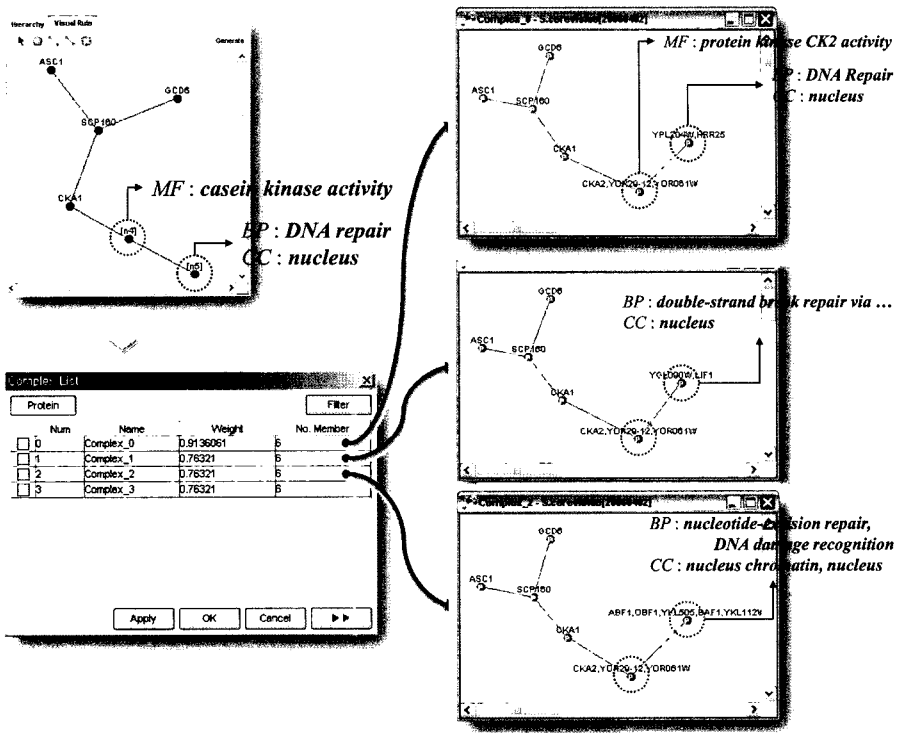


그림 16 확장된 Yeast Protein Complex 6 탐색

연결하였다. 정의된 개념 모듈 표현 규칙과 일치하는 4개의 기능 모듈을 탐색되었고, "n4"에는 "YOR061W"라는 유전자 이름을 가지는 단백질 하나만 매핑되고, "n5"에는 "YPL204W", "YGL090W", "YKL112W" 등 서로 다른 유전자 이름을 가지는 단백질이 매핑되는 것을 볼 수 있다. 그리고, 각각의 매핑된 단백질이 주어진 기능과 정확히 일치하지는 않지만 개념적으로 유사하다는 것을 알 수 있다.

5. 결론 및 향후 연구과제

본 논문에서는 방대하고 복잡한 단백질 상호작용 네트워크에서 생물학적으로 의미 있는 기능 모듈들을 개념에 기반하여 탐색할 수 있는 새로운 기법을 제안하였고 이를 시각화된 환경에서 구현하였다.

생물학적 기능 모듈들을 단백질 상호작용 네트워크에서 탐색하기 위한 기존의 방법들은 사용자가 먼저 핵심적 역할을 하는 단백질을 검색한 다음, 그 주위의 상호작용 관계들을 참조하면서 점차적으로 확장해나가는 방식으로 많은 시간과 노력이 요구된다. 본 논문에서는 사용자가 모듈 규칙을 시각화된 환경에서 정의하면, 복잡하고 방대한 단백질 상호작용 네트워크에서 정확히 일치하는 기능 모듈을 탐색할 수 있도록 하였다.

또한, 단백질 상호작용 네트워크에 존재하는 기능 모듈들을 자동으로 탐색하기 위한 클릭 또는 클러스터링 형태의 모듈 탐색 방법들은 제한된 구조의 기능 모듈들만을 탐색할 수 있다는 문제점이 있다. 본 논문에서 제안한 방법은 다양한 형태의 모듈들을 규칙으로 표현하여 탐색할 수 있다. 특히, 모듈에 대한 정확한 정보를 모를 경우, 온톨로지를 이용하여 탐색 규칙을 개념적으로 정의할 수 있다.

마지막으로, 비교적 다양한 구조의 기능 모듈을 탐색할 수 있는 기존의 PQL, BioNetSQL 등과 같은 질의 언어로는 복합적으로 구성된 기능 모듈 탐색이 지원되지 않는다. 본 논문에서 제안한 방법은 이미 잘 정의된 여러 규칙들을 조합하여 복합적인 형태의 기능 모듈을 탐색할 수 있으며, 이 기능 모듈을 구성하는 부분 모듈들 사이의 구체적인 연관 관계를 밝혀낼 수 있다. 이를 위해 본 시스템은 사용자가 부분 기능 모듈들을 탐색할 수 있는 규칙들을 시각화된 환경에서 쉽게 조합하여 복합 규칙을 정의할 수 있도록 지원한다.

본 논문에서 제안한 방법을 응용하면 다른 종에서 밝혀진 기능 모듈들을 이용하여 서로 다른 종의 단백질 상호작용 네트워크에서 유사한 기능 모듈들을 탐색할 수 있다. 예를 들어, 쥐(mouse)에서 생물학적 실험을 통

해 밝혀진 기능 모듈들을 일반화하여 규칙으로 정의하면, 사람(human)의 단백질 상호작용 네트워크에서 유사한 패턴의 기능 모듈들을 자동으로 탐색할 수 있다.

향후 연구로는 단백질의 아미노산 서열(amino acid sequence)나 단백질 구조(protein structure) 정보를 통해 규칙에 정의된 트리플과 상호작용 관계 사이의 유사도를 보다 정확하게 평가할 수 있어야 한다. 또한, 다양한 모듈의 형태를 기술할 수 있는 규칙 연산자들이 개발되어야 한다.

참 고 문 헌

[1] Tucker CL, Gera JF, and Uetz P, "Towards an understanding of complex protein networks," Trends Cell Biol., Vol.11, No.3, pp.102-106, 2001.

[2] Ravasz E, Somera AL, Mongru DA, et al., "Hierarchical organization of modularity in metabolic networks," Science, Vol.297, No.5586, pp.1551-1555, 2002.

[3] Ito T, Chiba T, Ozawa R, et al., "A comprehensive two-hybrid analysis to explore the yeast protein interactome," Proc. Natl Acad. Sci., Vol.98, No.8, pp.4569-4574, 2001.

[4] Gavin AC, Bosche M, Krause R, et al., "Functional organization of the yeast proteome by systematic analysis of protein complexes," Nature, Vol.415, No.6868, pp.141-147, 2002.

[5] Ho Y, Gruhler A, Heilbut A, et al., "Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry," Nature, 415, No.6868, pp.180-183, 2002.

[6] Bader GD, Betel D. and Hogue CW., "BIND: the Biomolecular Interaction Network Database," Nucleic Acids Res., Vol.31, No.1, pp.248-250, 2003.

[7] Xenarios I, Salwinski L, Duan XJ, et al., "DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions," Nucleic Acids Res., Vol.30, No.1, pp.303-305, 2002.

[8] Zanzoni A, Montecchi-Palazzi L, Quondam M, et al., "MINT: a Molecular INTeraction database," FEBS Lett., Vol.513, No.1, pp.135-140, 2002.

[9] Joshi-Tope G, Gillespie M, Vastrik I, et al., "Reactome: a knowledgebase of biological pathways," Nucleic Acids Res., Vol.33, Database issue, pp.D428-D432, 2005.

[10] Dogrusoz U, Erson EZ, Giral E, et al., "PATIKAweb: a Web interface for analyzing biological pathways through advanced querying and visualization," Bioinformatics, Vol.22, No.3, pp.374-375, 2006.

[11] Li XL, Tan SH, Foo CS, et al., "Interaction Graph Mining for Protein Complexes Using Local Clique Merging," Genome Informatics, Vol.16, No.2, pp.260-269, 2005.

[12] Zhang S, Ning X, Zhang XS., "Identification of functional modules in a PPI network by clique percolation clustering," Computational Biology and Chemistry, Vol.30, No. 6, pp.445-451, 2006.

[13] Bader GD and Hogue CW., "An automated method for finding molecular complexes in large protein interaction networks," BMC Bioinformatics, Vol.4, Article 2, 2003.

[14] Lubovac Z, Gamalielsson J, Olsson B., "Combining functional and topological properties to identify core modules in protein interaction networks," Proteins, Vol. 64, No. 4, pp.948-959, 2006.

[15] Koyuturk M, Grama A, and Szpankowski W., "An efficient algorithm for detecting frequent subgraphs in biological networks," Bioinformatics, Vol.20, Suppl. 1, pp.i200-i207, 2004.

[16] Leser U., "A query language for biological networks," Bioinformatics, Vol.21, Suppl.2, pp.ii33-ii39, 2005.

[17] Baitaluk M, Qian X, Godbole S, et al., "PathSys: integrating molecular interaction graphs for systems biology," BMC Bioinformatics, Vol.7, 2006.

[18] Boeckmann B, Bairoch A, Apweiler R, et al., "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003," Nucleic Acids Res., Vol.31, No.1, pp.365-370, 2003.

[19] Kanehisa M, Goto S, Kawashima S, et al., "The KEGG resource for deciphering the genome," Nucleic Acids Res., Vol.32, Database issue, pp.D277-D280, 2004.

[20] Harris MA, Clark J, Ireland A, et al., "The Gene Ontology (GO) database and informatics resource," Nucleic Acids Res., Vol.32, Database issue, pp.D258-D261, 2004.

[21] Rinaldi F, Schneider G, Kaljurand K, et al., "An environment for relation mining over richly annotated corpora: the case of GENIA," BMC Bioinformatics, Vol.7, Suppl.3, Article S3, 2006.

[22] Yeast Protein Complex Database, http://yeast.cellzome.com/

박 종 민

2000년 전북대학교 컴퓨터과학과, 학사 2002년 전북대학교 전산통계학과, 석사 2004년 전북대학교 전산통계정보학과, 박사 수료. 2004년~2007년 3월 한국전자통신연구원. 관심 분야는 바이오인포매틱스, 온톨로지, 시맨틱 웹, 데이타베이스

최 재 훈

1994년 전북대학교 컴퓨터과학과, 학사
1996년 전북대학교 전산통계학과, 석사
2000년 전북대학교 전산통계학과, 박사
2000년~현재 한국전자통신연구원. 관심분야는 온톨로지, 시맨틱 웹, u-헬스케어, 데이타베이스

박 수 준

1991년 University of Iowa, 학사. 1994년 Lehigh University, Computer Science, 석사. 1994년~현재 한국전자통신연구원 라이프인포매틱스팀 팀장. 관심분야는 행위추적, u헬스케어, Bioinformatics, 영상처리, HCI

양 재 동

1983년 서울대학교 컴퓨터공학과, 학사
1985년 한국과학기술원 전산학과, 석사
1991년 한국과학기술원 전산학과, 박사
1995년~1996년 Univ. of Florida, Visiting Scholar. 현재 전북대학교 전자정보공학부 교수, 영상·정보 신기술연구소 연구원. 관심분야는 OODBs, Expert System, CASE, 온톨로지, 시맨틱 웹