생물정보데이터베이스의 기능주석 검색을 위한 질의 확장 기법

유상원[0] 이강표 김형주
서 울 대 학 교 컴 퓨 터 공 학 부
swyoo@idb.snu.ac.kr kplee@idb.snu.ac.kr hjk@snu.ac.kr

A Query Expansion Technique for the Functional Annotation in Biological Databases

Sangwon Yoo[0] Kangpyo Lee Hyoung-Joo Kim
Department of Computer Science and Engineering, Seoul National University

요 약

생물학분야에 있어서 유전체학과 단백질체학의 발달로 인해 대량의 서열 분석과 이에 연관된 데이터들이 쏟아져 나오고 있다. 이러한 데이터는 전문가들에 의해 분석되어 그 기능에 대한 부가적인 설명들이 붙게 된다. 이 주석은 기관마다 고유한 용어와 체계를 가지고 있으며 갈수록 그 용어의 수는 증가하고 관계는 복잡해지고 있다. 본 논문에서는 간단한 SQL의 확장과 변환을 통해 다양한 생물정보학 주석 체계를 지원하는 기법을 제안한다. 우리가 제안하는 기법은 현재 사용 중인 단백질데이터베이스에 대한 실험을 통해 질의 수행시간의 큰 부담 없이 훨씬 더 많은 결과를 제시함을 보였다.

1. 서론

1.1 기능주석

생물학에서는 유전체학(Genomics)이나 단백질체학(Proteomics)과 같은 연구분야가 급속하게 발전하면서 각종 서열과 실험에 따른 데이터들이 갈수록 증가하고 있다. GenBank[1]의 발표에 따르면 한달에 300만개가 넘는 서열이 연구자들에 의해 제출되고 있으며 2005년에 그동안 제출된 서열의 총길이가 100 Gigabases(letters)에 달한다고 보고된바 있다.

이러한 서열자체에 대한 데이터 외에도 각각의 서열이 나타내는 유전자나, RNA, 단백질의 기능과 관련된 여러 가지 부가정보도 데이터베이스에 함께 기록된다. 이러한 부가정보는 해당 서열이 속한 종, 결과가 발표된 논문지, 실험관련 데이터, 유전자의 기능에 대한 명세 등 다양한 내용을 포함하고 있다.

이러한 여러 형태의 주석 정보 중에서 유전자나 RNA 또는 단백질의 기능을 정확하게 명세하기 위한 생물학 분야의 많은 노력이 있어왔다. 예를 들어, UniProt[2]은 유럽의 생물정보연구소(EBI)에서 운영하는 단백질에 관한 가장 포괄적인 정보를 담고 있는 공개 데이터베이스이다. 이 데이터베이스는 단백질의 기능을 명세하기 위해 해당 분야의 전문가들이 직접 논문이나 관련 자료를 검토하여 단백질의 기능 주석(functional annotation)을 추가하고 있다. 만약 어떤 단백질이 전사(transcription)를 조절하는데 관여한다는 것이 밝혀졌다면 해당되는 단백질의 엔트리에는 “transcription regulator”와 같은 기능을 명세하는 주석이 붙게 된다. 이러한 정보들은 연구자들이 이 단백질이나 관련이 있는 단백질들을 연구할 때 큰 도움을 주게 된다.

UniProt이외에도 많은 데이터베이스들이 특정한 종이나 특정한 분야에 대해 유전자나 단백질의 기능을 명세하기 위한 노력을 기울이고 있다.

1.2 문제점

많은 데이터베이스들이 앞서 언급한 기능관련 주석을 포함하고 있지만 이러한 기능관련 주석을 검색할 때 몇 가지 문제점을 고려해야만 한다. 첫번째 문제는 같은 기능을 기술하는 용어나 체계[3, 4, 5]가 다양한데 이 다양한 어휘들이 하나의 DB내에서 함께 쓰이고 있다는 점이다. 일반적으로 생물학DB는 여러 자료들을 통합하여 DB를 구성하는 경우가 많다. 이 때 통합 DB에 대해 질의를 수행하려면 모든 주석체

ID	Protein Name	Function	Function name
Q9NY61	AATF_HUMAN	GO:0003700	① Transcription factor activity
P05549	AP2A_HUMAN	GO:0003705	② RNA polymerase transcription factor activity
Q92876	HXB13_HUMAN	InterPro:IPR001356	③ Homeobox
O08686	BARX2_MOUSE	InterPro:IPR000047	④ Helix-turn-helix motif
P19622	HME2_HUMAN	InterPro:IPR000747	⑤ 'Homeobox' engrailed-type protein

그림 1. 서로 다른 주석 체계를 포함하는 단백질 테이블

계에 대해 알고 있지 않는 한 질의를 작성하는데 어려움이 발생하게 된다. 예를 들어 그림1의 단백질 테이블은 Gene Ontology[4] 와 InterPro[5]를 이용한 기능 주석을 포함한 예제이다. 이 테이블에 대해 질의를 작성할 때 사용자가 둘 중 어느 한쪽 주석체계에만 익숙하다면 질의어 작성에 한쪽의 어휘만 포함됨으로 인해서 원하는 결과를 얻지 못할 수도 있다.

두번째 문제는 이러한 주석체계 내에서 사용되는 어휘간의 계층적인 구조가 존재한다는 것이다[4, 5, 6]. 주석에 사용된 개념의 계층적인 구조 내에서 위로 올라갈수록 더 큰 개념을 나타내며 아래로 내려올수록 더 세부적인 개념을 나타나게 된다. 그런데 생물학 DB내에서 단백질이나 유전자에 대한 주석을 작성할 때 어느 정도 수준의 어휘를 선택할 것인가는 전적으로 DB를 작성하는 전문가의 몫이다. 따라서 질의어를 작성하는 사용자가 해당 기능에 대한 명세가 상위개념으로 명세가 되어 있는지 세부개념으로 명세가 되어있는지 미리 알고 질의어를 작성한다는 것은 매우 어려운 일이다.

마지막으로 고려해야 할 사항은 서로 다른 주석체계 사이에 주석체계를 다루는 전문가들이 작성한 대응관계가 존재한다는 점이다[7]. 대응관계들은 각 주석체계를 사용하는 기관에서 전문가들이 주기적으로 작성하여 업데이트가 이루어지고 있다. 따라서 우리가 유전자나 단백질의 기능을 기술하고 있는 주석에 대한 질의를 수행할 때 원하는 결과를 얻기 위해서는 위의 세가지 문제점을 질의 과정에서 해결해야만 한다.

그림 1에 나타난 테이블은 단백질의 기능을 서로 다른 체계로 명세하고 있는 데이터베이스의 예제이다. 처음에 나타나는 두 행은 GeneOntology[4]를 이용해서 기능을 명세하고 있고 나머지 세 개의 행은 InterPro[5]의 단백질 분류체계를 따라 이를 명세하고 있다. 일반적으로 하나의 데이터베이스 안에 다양한 종류의 주석체계를 포함할 수도 있고 여러 데이터베이스를 통합하여 위와 같은 문제들이 발생할 수도 있다.

연구자가 다음과 같은 질의를 가지고 있다고 가정하자. “transcription factor activity에 관여하는 단백질과 이의 하위 기능을 수행하는 단백질들은 찾아라.” 이 예제 질의를 가진 연구자는 GO:0003700이 transcription activity를 나타낸다는 사실만 알고 있다. 그러나 테이블 내에 명시적으로 나타나지는 않지만 GO:0003705가 GO:0003700의 하위개념을 나타내며 IPR001356이 GO:0003700에 대응되어 transcription activity를 수행하는 단백질 도메인을 가진다는 정보가 존재한다. 그렇다면 사용자가 원하는 질의 결과를 얻기 위해서는 DBMS에서 이러한 정보들이 함께 처리될 수 있도록 해야 할 것이다.

1.3 해결방법

이 논문은 위의 예제에서 나타날 수 있는 것과 같은 문제점들을 해결할 수 있는 간단하고 효율적인 방법을 제안하고 실험을 통해 이를 검증하였다. 우리는 SQL로 이루어진 질의를 확장할 수 있는 방법과 주석체계를 저장하는 인덱스 테이블을 제공하여 사용자가 질의의 대상이 되는 주석체계의 구조나 대응관계에 대해 알지 못하는 상황에서도 원하는 질의 결과를 얻을 수 있도록 한다.

우리가 제안하는 방법은 사용자에게 SQL과 더불어 EXPAND라는 절을 추가로 제공하여 사용자가 원하는 확장의 범위와 주석체계를 정할 수 있도록 하고 있다. 사용자가 EXPAND절을 사용하여 작성한 SQL은 일반적인 SQL로 변환되어 DBMS에 전달되고 사용자는 원하는 결과를 얻게 된다.

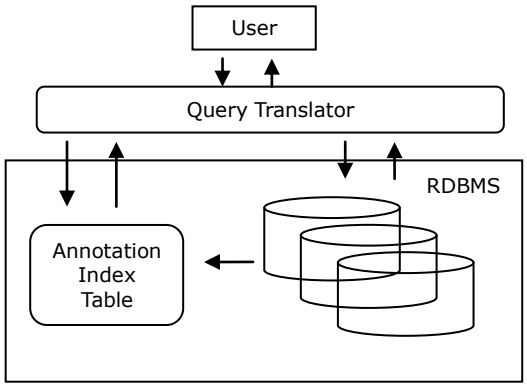


그림 2. 질의확장 시스템의 간단한 구조

그림 2에는 간단한 질의 확장시스템의 구조가 표현되어 있다. 그림의 왼쪽 하단에는 주석체계의 구조와 주석체계간의 대응관계를 담고 있는 테이블이 있다. 사용자가 입력한 질의는 질의변환기가 주석체계 색인 테이블을 이용해 일반적인 SQL로 변환한 후 주석 데이터가 들어있는 데이터베이스에 질의를 수

행하게 된다.

이 연구에서 기여하는 바는 여러 생물정보DB에서 포함하고 있는 기능주석들에 대한 간단한 질의 방법을 제공한다는 것이다. 여러 생물DB들을 자신의 DB에 복사해 사용하거나 다양한 출처의 데이터들을 통합하는 환경에서 사용자에게 편리한 질의 방법을 제공한다.

또다른 장점은 관계형 데이터베이스와 SQL을 활용한다는 것이다. 관계형 데이터베이스는 이미 30년 이상 여러 분야에서 사용되어왔고 많은 생물학데이터베이스 또한 관계형 데이터베이스의 형태로 제공되고 있다. 생물학분야에서 데이터를 처리하기 위한 방법으로 관계형 데이터베이스는 이미 널리 쓰이고 있는 환경이기 때문에 우리가 제안한 기법을 쉽게 적용할 수 있다.

이 논문의 나머지 구성은 다음과 같다. 2장에서는 우리가 제안한 주석색인과 질의 변환기법에 대해 소개하고 3장에서는 실험방법과 결과에 대해 기술한다. 4장에서는 DB분야와 생물정보학 분야의 관련연구를 살펴보고 5장에서 결론을 내린다.

2. 질의확장

2.1 주석색인

그림1의 테이블 구조에는 나타나지 않지만 기능주석 사이에는 다음과 같은 관계가 존재한다.

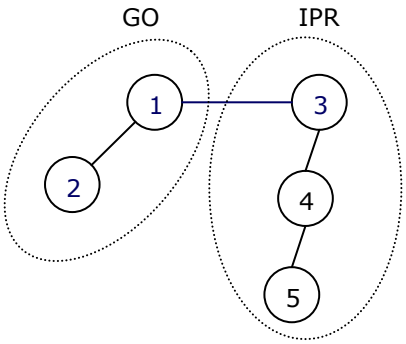


그림 3. 주석의 계층적인 구조

그림3에서 하나의 노드는 주석에 사용된 어휘를 나타내며 간선은 어휘간의 관계를 나타낸다. 상하로 연결된 간선은 어휘간의 부모-자식 관계를 나타내며 평행하게 연결된 간선은 서로 다른 주석체계간에 대응관계를 나타낸다. 우리는 주석체계를 DAG (Directed Acyclic Graph)로 모델링하였다. 사용자의 질의가 입력되면 확장질의를 생성하기 위해 DAG구조를 탐색하게 된다.

만약 사용자가 그림1에 나타난 테이블에 대해 다음과 같은 질의를 한다고 가정하자.

```
SELECT ID
FROM protein_table
WHERE Function = 'GO:0003700'
```

이 질의는 'Q9NY61'만을 결과로 제공할 것이다. 만약 사용자가 전사작용(transcription factor activity)과 연관된 단백질 또는 관련 단백질군(protein family)을 찾고 싶다면 그림3에 나타난 정보들을 질의 안에 포함해야만 원하는 결과를 모두 얻을 수 있다.

우리는 그림3과 같은 관계들을 질의 안에 포함시키기 위해 네 가지의 대표적인 주석체계들을 선택하여 그 계층구조를 추출하였다. Gene Ontology[4], InterPro[5], SwissProt[3], EC[6]의 네 가지 주석체계는 현재 여러 생물정보데이터베이스 내에서 사용중인 대표적인 주석체계들이다. 이들은 Gene Ontology와 대응관계를 가지고 있으며[7] 내부적으로 복잡한 계층구조를 가지고 있다[4, 5].

그림 3에 나타난 주석체계간의 관계는 다음의 형식으로 표현된다.

```
<go:term rdf:about="GO:0003705">
<go:is_a rdf:resource="GO:003700" />
</go:term>

<interpro id="IPR000047" type="Domain">
      <parent_list>
        <rel_ref ipr_ref="IPR001356" />
      </parent_list>
     <child_list>
        <rel_ref ipr_ref="IPR000747" />
     </child_list>
</interpro>
```

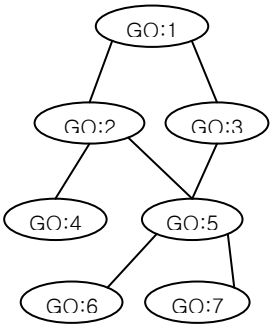
InterPro:IPR000047 Helix-turn-helix motif, lambda-like repressor > GO:transcription factor activity ; GO:0003700

그림 4. GO와 InterPro의 구조 및 대응관계

그림4에 나타난 것처럼 유전자와 단백질에 기능주석을 명세하기 위한 체계와 대응관계는 XML, RDF, 또는 자체적인 형식 등 다양한 형태를 가지고 있다. 형태는 다양하지만 이들이 가지고 있는 정보는 기능을 명세하기 위해 사용되는 어휘와 어휘에 할당된 코드간의 관계이다. 따라서 우리는 기능주석간의 관계를 기술하고 있는 형태에 무관하게 이를 전처리하여 DAG 구조로 변환하였다.

우리가 선택한 네 종류의 주석체계 중 Gene Ontology와 InterPro는 DAG구조를 가지고 있고 EC는 트리구조를 가지고 있으며 SwissProt의 경우는 계층적인 구조가 아닌 평면적인 구조를 가지고 있다.

질의확장의 가장 중심이 되는 것은 사용자가 검색조건으로 입력한 질의어를 기준으로 연관이 있는 어휘들을 추가하는 것이다. 이를 위해서는 연관성이 있는 어휘들을 그래프내에서 효율적으로 탐색하여 추가할 수 있어야 한다. 우리는 이러한 요구사항을



ID	Label
GO:1	03
GO:2	03.01
GO:3	03.02
GO:4	03.01.01
GO:5	03.01.02
GO:5	03.02.01
GO:6	03.01.02.01
GO:6	03.02.01.01
GO:7	03.01.02.02
GO:7	03.02.01.02

Annotation 1	Annotation 2
GO:0003700	IPR000047
GO:0003700	IPR001827

그림 5. DAG구조와 대응관계의 테이블 표현

지원하기 위해 Dewey Order[8]를 사용하였다.

그림5는 왼쪽의 DAG 구조를 우리가 사용한 테이블을 통해 표현한 것이며 첫번째 칼럼은 각 어휘의 id값이나 이름을 가지며 두번째 칼럼은 Dewey Order를 이용한 레이블을 표현한다. 예를 들어 테이블만으로도 GO:4는 GO:2의 자식 노드라는 것을 쉽게 알 수 있다. 왜냐하면 GO:2와 GO:4는 공통접두사로 '03.01'을 가지기 때문이다. 마찬가지 방법으로 부모-자식 관계뿐 아니라 조상-후손 관계도 손쉽게 계산할 수 있다. 각각의 레벨은 점으로 분리되며 공통접두사 이외의 부분의 길이를 계산하면 자신과 조상 노드 사이의 거리를 알 수 있다.

GO:5, GO:6, GO:7은 그래프 내에서는 하나의 노드로 표현되지만 테이블 내에서는 두 개의 레이블을 가진다. Dewey Order는 원래 트리 구조의 분류체계를 만들기위해 고안된 것이었다. 트리 구조에서 하나의 노드는 둘 이상의 부모를 가질 수 없지만 DAG구조에서는 하나의 노드가 여러 개의 부모 노드를 가질 수 있기 때문에 루트까지의 경로에 따라 복수의 레이블을 가지게 된다. 주석체계에서 사용되는 어휘들은 복수의 부모어휘들을 가질 수 있으며 하나의 위치뿐 아니라 DAG구조내의 여러 곳에 위치할 수 있다. 우리는 이러한 정보를 표현하기 위해 테이블 내에 하나의 노드가 가질 수 있는 모든 경로를 레이블로 표현하였다.

사용자가 질의어에 입력한 기능주석어휘로부터 상위개념의 탐색을 위해 접두사함수를 만들었다. 접두사함수는 현재위치로부터 원하는 레벨의 상위개념을 얻어낼 수 있도록 해준다. 하위개념으로의 탐색을 위해서는 SQL의 LIKE함수를 이용하였다. 그림5에서 보면 각각의 레벨에는 일정한 수의 숫자가 할당되기 때문에 SQL로 다음과 같은 표현이 가능하다.

```
SELECT ID
FROM index
WHERE Label like '03.01.__'
```

이 질의는 GO:2의 자식에 해당하는 모든 노드들을 결과로 얻게 해준다. 만약 우리가 LIKE연산의 조건으로 '03.01.%'를 사용한다면 자식 노드뿐 아니라 모든 후손 노드들을 얻을 수 있다.

서로 다른 주석체계 사이의 대응관계는 그림5와 같이 매핑 테이블을 만들어 서로 다른 주석체계 사이에 유사한 의미 또는 대응되는 개념으로 사용되는 어휘들을 쉽게 찾을 수 있도록 해준다. 사용자가 질의에 사용한 기능주석이 다른 주석체계로 확장되도록 지정하면 이 테이블을 검색하여 어떤 어휘가 대응관계에 있는지를 알아낸 후 자동적으로 질의에 추가된다.

그림5에서 표현한 두 종류의 테이블은 질의변환기로 하여금 사용자가 입력한 기능주석어휘에 추가해야할 연관어휘들을 효율적으로 찾는 것을 지원해준다.

2.2 SQL 확장

사용자가 기능주석에 대해 질의를 생성할 때 가장 중점을 두는 부분은 어떤 어휘를 이용해서 질의를 해야 원하는 결과를 얻을 수 있는 가이다. 적절한 어휘가 원하는 결과를 얻게 해주지만 기능주석에서 포함하고 있는 어휘들을 살펴보면 그 모두를 기억하기에는 관계가 복잡하고 그 수가 너무 많다.

SQL의 경우 이러한 어휘들은 WHERE조건 절에 포함되게 된다. 사용자가 선택할 수 있는 방법은 자신이 알고 있는 검색어들을 WHERE절에 포함하는 것이다. 하지만 기능주석 내에 존재하는 상하위관계나 서로 다른 체계간의 대응관계들을 지원하는 편리한 방법은 존재하지 않는다.

우리는 SQL뒤에 덧붙여 'EXPAND'라는 절을 제공하여 사용자가 본인이 대상으로 하는 주석체계와 질의의 확장범위를 지정할 수 있도록 하였다.

예를 들어, 단백질정보를 담고 있는 테이블이 Gene Ontology와 InterPro 그리고 SwissProt 세가지 주석을 사용하고 있다고 가정하자. 아래의 질의는 EXPAND절을 사용하여 이 테이블에 질의를 작성한 것이다.

```
SELECT id
FROM protein_table
WHERE organism= 'Human' and
        function = 'GO:0003700'
EXPAND go>-3   ipr<+3   spkw=0
```

그림 6. EXPAND 질의

EXPAND절 이외의 내용은 일반적인 SQL과 동일하다. 'go', 'ipr', 'spkw'는 각각의 주석체계를 나타내며 질의의 대상이 되는 주석체계에 따라서 추가하거나 삭제할 수 있다. '+'와 '-'는 질의 확장의 방향을 나타낸다. '+'는 하위개념으로의 질의확장을 '-'는 상위개념으로의 질의확장을 나타낸다. '<'와 '>'는 질의확장의 범위를 나타내며 '='는 상하위 관계로의 확장이

아닌 대응관계를 표현한다.

위의 예제에서 go>-3 은 세 레벨의 어휘를 질의에 포함하게 된다. 사용자가 준 'GO:0003700'과 이 용어와 부모관계에 있는 용어 그리고 이 용어의 부모의 부모관계에 있는 용어를 포함한다. ipr<+3 은 InterPro내에서 'GO:0003700'과 연관있는 어휘들을 탐색하게 된다. 먼저 InterPro에서 사용중인 어휘들 중 'GO:0003700'과 대응되는 관계로 연결되어 있는 어휘들을 먼저 찾고 그 어휘들의 자식관계 그리고 다시 자식의 자식 관계에 해당하는 어휘들로 확장을 해 나가게 된다. 즉 부등호와 숫자를 통해 확장의 방향과 범위가 정해지게 된다. 마지막에 기술된 spkw=0 의 경우는 'GO:0003700'과 대응관계에 있는 SwissProt내의 어휘들을 질의에 추가해주는 것으로 끝나게 된다.

일반적으로 EXPAND를 사용한 확장질의의 형태는 다음과 같은 형식을 가지게 된다.

```
SELECT select_list
FROM from_list
WHERE where_list
EXPAND expand_list
```

확장질의를 일반적인 SQL문으로 변환할 때 우리가 고려하는 환경은 하나의 테이블 안에 다양한 주석체계가 사용되는 경우이다. 따라서 질의변환알고리즘의 핵심은 어떻게 where_list 안에 확장된 어휘들을 추가할 것인가이다. 위의 형식에서 SELECT 문과 FROM 문은 바뀔 필요가 없다. 왜냐하면 검색의 대상이 되는 어휘가 추가될 뿐 대상이 되는 테이블이나 다른 검색조건은 바뀌지 않기 때문이다. 그림 6의 예제를 일반적인 SQL문으로 변환하면 다음과 같다.

```
SELECT id
FROM protein table
WHERE organism='Human' and
(function='GO:0003700' or function='GO:0003677'
or function='GO:0003676' or function= 'GO:0030
528' or function='IPR001356' or function= 'IPR 00
0047' or function='IPR001827' or function= 'IPR 0
00747' or function = 'KW-0803' )
```

WHERE절에 3개의 Gene Ontology어휘와 4개의 InterPro어휘 그리고 1개의 SwissProt어휘가 추가되었다. 이와 같은 방식으로 사용자의 질의는 사용자가 입력한 기능주석을 중심으로 연관된 어휘들을 포함하는 형태로 확장되게 된다.

만약 우리가 하나의 테이블에서 다양한 형태의 주석을 사용하는 것이 아니라 하나의 테이블에서 하나의 주석시스템을 사용하는 테이블들이 모여있는 경우라고 가정해도 확장질의를 변환하는 것은 간단하다. 질의변환 알고리즘에서 해야 하는 것은 각 테이블의 기능주석칼럼을 나타내는 조건식에 OR 조건을 이용해 추가되는 어휘를 더해주기만 하면 되기 때문이다. 추가된 어휘들은 각각의 테이블들을 검사하여 그 결과를 얻게 된다.

질의 변환 알고리즘은 다음과 같다.

Input: expanded SQL
Output: SQL

1. Scan expand_list
2. Retrieve and save the names and ranges of annotation systems
3. Scan where_list
4. Retrieve and save the original query term with its annotation system name
5. Search the value in 4 from the mapping table
6. A mapping table returns the value which belongs to the annotation system in 2
7. Search the annotation index within the range of 2 from the values returned in 6
8. Add retrieved results in 7 to the where_list

1, 2에 나오는 EXPAND절에 관한 문법은 이미 앞에서 자세히 설명하였다. 3~8의 내용은 질의확장과정에서 주석색인 테이블이 어떻게 사용되는지 설명하고 있다. 3과 4에서 질의변환기는 사용자가 지정한 기능주석어휘의 값을 추출해낸다. 이 값들은 질의확장의 출발점으로 사용된다. 이 값들에 대응되는 값들을 다른 주석체계 사이에서 찾아내기 위해 5에서 매핑 테이블이 사용된다. 매핑 테이블은 임의의 서로 다른 주석체계 사이의 대응관계를 담을 수 있지만 우리가 구현한 테이블은 Gene Ontology 와 다른 주석체계 사이의 정보만을 담고 있다. 이러한 이진관계만을 가지고도 우리가 사용한 네 가지 주석체계 중 EC number 와 SwissProt 또는 InterPro간의 관계도 유추가 가능하다. 하지만 이러한 것들은 기술적인 문제가 아니라 그 대응관계가 전문가에 의해 검증이 되었느냐는 의미의 문제이기 때문에 이 논문에서는 고려하지 않았다. Gene Ontology와 다른 주석체계와의 관계는 Gene Ontology Consortium에서 제공하고 있다[7].

6에서는 매핑 테이블이 사용자가 지정한 값에 대응되는 값들을 반환한다. 어휘들간의 대응관계는 일반적으로 m:n 이며 대응되는 어휘가 없을 수도 있다. 따라서 대응되는 값의 숫자는 0에서 수십까지 다양할 수 있다.

7에서는 6에서 얻어낸 값들을 출발점으로 상하관계의 확장을 수행한다. 2에서 어떤 대상이 되는 주석체계와 범위를 얻어냈기 때문에 이 값들에서 지정하는 대로 주석색인 테이블을 검색하여 원하는 값들을 얻어낸다. 사용자가 대응관계만을 지정했다면 7은 수행되지 않는다. 마지막으로 확장된 어휘의 집합이 WHERE절에 추가된다. 사용자가 질의어로 입력한

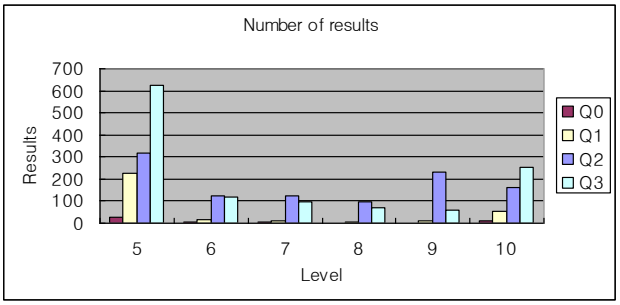


그림 7. 확장질의와 일반질의의 결과수 비교

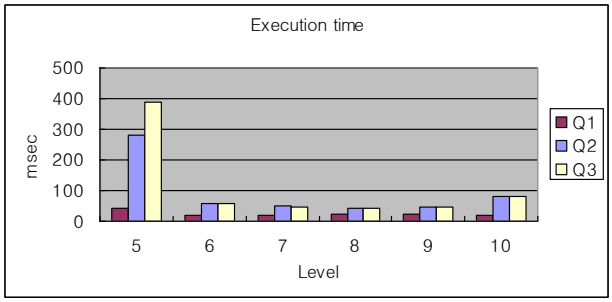


그림 8. 확장질의의 수행시간

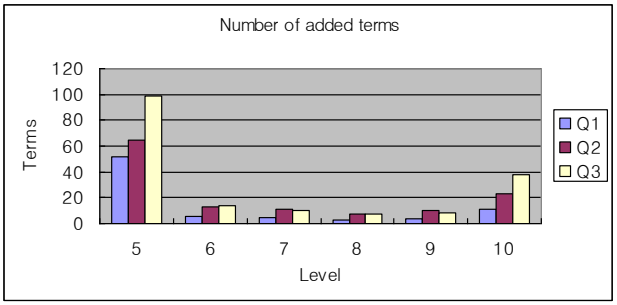


그림 9. 확장질의에 추가된 어휘의 숫자

어휘는 사용자가 알고 있는 한 두 개에 불과하지만 질의 확장의 결과로 SQL에는 많은 어휘들이 포함되게 된다. 나머지 질의처리는 DBMS의 몫이다.

3. 실험 및 실험결과

3.1 데이터와 실험환경

본 연구에서 사용한 데이터는 다음과 같다. 주석체계로는 Gene Ontology[9], InterPro[5], EC number[6], SwissProt keyword[3]를 사용하였다. 이 네 가지 주석체계는 서로 다른 주석체계로 이루어진 데이터들을 통합하는 주요 생물학 분야의 프로젝트에서 현재 사용중인 체계이다[10, 11]. [10, 11]에서 수행한 GOA project에서 서로 다른 주석체계간의 관계를 전문가들이 검토하고 작성하였으며 2007년 4월 기준으로 1600만여개의 기능주석을 포함하고 있다.

우리가 질의의 대상으로 삼은 데이터베이스는 이 프로젝트안에 포함된 SwissProt데이터베이스이다. SwissProt데이터베이스는 단백질에 대한 정보를 전문가들이 직접 문헌을 검증하고 주석을 입력하는 신뢰도가 높은 가장 규모가 큰 데이터베이스이다. 단순히 서열유사성을 이용한 것이 아니라 사람이 직접 정보를 입력하기 때문에 DAG내의 다양한 깊이의 어휘를 사용한 정보가 입력되어 있다. 우리는 여러 주석체계 중 앞에서 언급한 네 가지의 주석체계로 이루어진 220만개의 주석을 추출하여 사용하였다.

질의변환기는 JAVA를 이용하여 구현되었으며 MySQL5.0 DBMS를 사용하였다. 실험에 사용된 서버는 2.8GHz dual CPU, 4G RAM, Linux OS의 사양을 가지고 있다.

3.2 실험결과

우리가 실험에 사용한 질의는 다음과 같다. “Gene Ontology로 기술된 molecular function을 수행하는 인간 단백질을 찾아라” 이를 확장 이전의 SQL과 세 가지 형태의 확장SQL질의로 표현하였다. 확장이전의 SQL은 다음과 같은 형식을 취한다.

SELECT id
FROM protein_table
WHERE organism=’human’ and function=’GO에서 molecular function을 기술하는 어휘’

Q0: 일반질의
Q1: 대응관계를 이용한 확장질의
(EXPAND ipr=0 ec=0 spkw=0)
Q2: 상위개념으로의 확장질의
(EXPAND go>-3 ipr>-3 ec>-3 spkw=0)
Q3: 하위개념으로의 확장질의
(EXPAND go<+3 ipr<+3 ec<+3 spkw=0)

Q1~Q3에서 질의의 출발점은 Gene Ontology로 기술된 어휘이다. 우리는 Gene Ontology의 레벨 1에서 15중에 레벨 5에서부터 레벨 10까지에 해당하는 어휘를 랜덤하게 추출하여 질의의 출발점으로 사용하였다. 위 그래프의 결과는 각각의 레벨에서 1000회를 반복하여 랜덤하게 추출한 어휘를 대상으로 실험한 결과의 평균값이다.

그림7은 확장질의에 의해 증가된 질의결과수를

보이고 있다. 질의확장을 이용하지 않고 얻어낸 단백질의 수는(Q0) level 5에서 27개가 최고 수치였다. 질의 확장을 적용하게 되면 모든 질의확장형태와 모든 레벨에서 훨씬 더 많은 단백질을 질의결과로 얻어낼 수 있다는 것을 알 수 있다. 특별히 level 5에서 다른 레벨에 비해 훨씬 더 많은 질의결과를 얻을 수 있는 이유는 Q1에서 보듯 Gene Ontology와 다른 주석체계간에 대응관계로 연결된 어휘들이 많기 때문이다. 여기서 우리가 주목할 점은 확장질의를 적용하지 않았을 경우 얻지 못할 많은 유용한 결과들을 얻어낼 수 있다는 점이다.

그림8은 질의변환과 질의수행시간의 합을 나타내고 있다. 질의변환에 걸리는 시간은 미미하므로 실제로는 DBMS의 질의수행시간 측정이라고 할 수 있다. 그래프를 보면 가장 시간이 많이 소요되는 level 5의 경우 수행시간이 대략 0.4초 가량인 것을 알 수 있다. 현재 전문가들이 직접 주석을 단 가장 큰 데이터베이스를 대상으로 한 시간측정이므로 현재 질의확장기법은 수행시간면에서 만족할만하다고 볼 수 있다.

그림9는 질의확장과정에서 추가된 어휘의 개수를 보여주고 있다. 그림9를 그림 7, 8과 비교해보면 추가된 어휘의 개수가 질의 결과의 수나 수행시간에 영향을 미친다는 것을 알 수 있다. 주석체계의 구조가 트리 구조라면 Q3의 경우가 추가되는 어휘의 수가 가장 많아야 하겠지만 우리가 수행한 실험결과를 보면 Q2와 Q3사이에 큰 차이가 없다는 것을 알 수 있다. 그 이유는 InterPro의 경우 복수의 부모노드를 가지는 많은 경로가 존재하여 상위개념으로의 확장시에도 많은 어휘가 추가되기 때문이다.

이 실험에서 우리가 얻은 결론은 확장질의가 적은 수행시간에 비해 훨씬 더 많은 질의결과를 제공한다는 것이다. 관계형 데이터베이스는 이미 질의처리에 있어서 최적화된 여러 기능을 가지고 있기 때문에 SQL을 활용하면 큰 시간의 부담 없이 유용한 많은 결과들을 확인할 수 있게 해준다.

4. 관련 연구

데이터베이스 분야에서는 최근 들어 데이터의 주석에 대한 관심이 증대하고 있다[12, 13, 14]. 이러한 연구들은 데이터의 주석들을 DBMS내에서 처리하는 노력을 기울이고 있다. 이 연구들에서 응용분야로 삼고 있는 부분은 과학데이터 그 중에서도 특히 생물학분야를 대상으로 하고 있는데 이는 데이터에 대한 부가정보들이 지식을 공유하는데 핵심적인 역할을 하기 때문이다.

이 연구들 중 [12]는 데이터의 주석을 질의 내에서 전달하는 문제에 대해 다루었다. 예를 들어 하나의 단백질에 대해 서로 다른 주석이 서로 다른 테이블에 있을 경우 질의를 어떻게 작성하느냐에 따라 단백질의 이름이나 키 값은 결과 안에 동일하게 나타나지만 원하는 주석을 얻지 못하는 문제가 발생할 수도 있다. 이를 해결하기 위한 SQL의 확장을 다루었다. [13]에서는 데이터베이스의 값들간의 연관관계에 대한 주석을 질의하는 문제에 대해 다루었다. 동일한 단백질에 대한 아이디가 데이터베이스마다 서로 다를 경우 전문가들은 이를 관리하기 위해 서로 같은 단백질을 가리킨다는 정보를 추가하게 된다. 이러면 서로 다른 데이터베이스의 값간에 연관관계에 대한 주석이 발생하는데 이 문제를 처리하는 방법에 대해 제안하였다. [14]에서는 데이터의 이동경로와 추가 삭제에 대한 기록을 주석으로 보고 이를 질의하는 과정에 대해 다루었다.

우리의 연구는 [12, 13]와 관계형 데이터베이스를 사용한다는 면에서 중복된다. 하지만 이들 연구에서는 주석자체가 서로 대응관계나 계층적인 구조를 가지고 있는 경우에 대해 다루고 있지 않다.

질의확장은 전통적으로 정보검색분야에서 많이 다루어 온 문제이다[15, 16]. 사용자가 제시한 검색어와 유사한 키워드들을 추가하여 더 나은 검색결과를 제시하는 연구가 지금까지 계속 이루어지고 있다. 정보검색분야의 초점은 어떻게 하면 유사한 또는 연관이 있는 검색어들을 찾아낼 것인가에 맞추어져 있다. 이것이 검색의 성능과 직결되기 때문이다. 본 연구의 경우 정보검색 분야의 관점에서 본다면 이미 전문가들에 의해 만들어진 유사한 어휘들을 추가하므로 유사어에 대한 고려는 확장의 범위와 질의처리에만 맞추어져 있다.

트리 구조를 레이블링하거나 DAG구조를 레이블링하는 문제는 XML 질의처리에서 이미 많이 다루어졌다[17, 18]. 일반적으로 최적화된 XML레이블링 기법은 특정한 XML질의처리를 수행하기 위해 고안된 것이다. 본 논문에서 다루는 주석체계는 상, 하위개념으로의 탐색만을 다루고 있기 때문에 Dewey Order[8]를 이용하는 것으로 충분히 해결 가능하다.

유전체학이나 단백질체학분야에서는 서열에 대해 주석을 달고 또 이 주석을 정확하게 달기 위한 많은 노력이 있어왔다. Gene Ontology[4]는 14개의 공개 생물학데이터베이스의 모임을 통해 기능주석의 표준안을 온톨로지의 형태로 제공하고 있다. 자연어로 단백질이나 유전자의 기능을 명세할 때 연구기관이나 연구자마다 서로 다른 용어를 사용함으로써 생기는 혼동을 피하기 위해 통일된 어휘체계를 만들고 계속해서 확장해 나가고 있는 상태이다. GOA project[10, 11]는 UniProt[2]의 데이터들에 Gene Ontology를 이용하여 주석정보를 추가하는 프로젝트이다. 이 과정에서 대규모의 데이터에 주석체계간의 대응관계를 이용해 Gene Ontology의 어휘가 추가되었고 별도로 전문가가 직접 문헌을 검증하여 Gene Ontology를 추가한 데이터베이스도 구축되었다. 이러한 노력은 신뢰도가 높은 데이터베이스를 구축하고 서비스하는데 목적이 있지만 DBMS를 이용한 질의나 검색의 지원에 관한 내용은 이 분야에서 다루어지고 있지 않다.

5. 결론 및 향후 연구

본 연구에서는 생물정보데이터베이스를 대상으로 한 질의 확장에 대해 다루었다. 우리는 실제 서비스되고

있는 단백질 데이터베이스를 대상으로 질의확장기법을 테스트하여 큰 시간의 부담 없이 많은 질의 결과를 얻어낼 수 있음을 보였다. 우리가 제안한 기법은 서로 다른 분야에 대해 명세하고 있는 주석체계들을 연결할 수 있는 간편한 방법을 제공하여 사용자가 손쉽게 질의를 작성할 수 있도록 지원한다.

향후 질의 결과의 순위화를 지원하게 되면 질의 확장으로 얻어진 결과를 분석하는데 더 많은 도움을 얻을 수 있을 것이다.

6. Acknowledgement

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성지원사업(IITA-2006-C1090-0603-0031)의 연구결과로 수행되었음

참고문헌

[1] Benson, D.A., et al., "GenBank", Nucleic Acids Research, 2006, 34(suppl_1): p. D16-20.

[2] Wu, C.H., et al., "The Universal Protein Resource (UniProt): an expanding universe of protein information", Nucleic Acids Research, 2006, 34(suppl_1): p. D187-191.

[3] Boeckmann, B., et al., "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003", Nucleic Acids Research, 2003, 31(1): p. 365-370.

[4] Consortium, G.O., "The Gene Ontology (GO) project in 2006", Nucleic Acids Research, 2006, 34(suppl_1): p. D322-326.

[5] Mulder, N.J., et al., "InterPro, progress and status in 2005", Nucleic Acids Research, 2005, 33(suppl_1): p. D201-205.

[6] Bairoch, A., "The ENZYME database in 2000", Nucleic Acids Research, 2000, 28(1): p. 304-305.

[7] Mappings of External Classification Systems to GO, http:// www. geneontology. org/GO.indices.shtml

[8] Online Computer Library Center, Introduction to the Dewey Decimal Classification

[9] Gene Ontology Annotation(GOA) Database, http://www.ebi.ac.uk/GOA/

[10] Camon, E., et al., "The Gene Ontology Annotation (GOA) Project: Implementation of GO in SWISS-PROT, TrEMBL, and InterPro", Genome Research, 2003, 13(4): p. 662-672.

[11] Camon, E., et al., "The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology", Nucleic Acids Research, 2004, 32(suppl_1): p. D262-266.

[12] Bhagwat, D., et al., "An annotation management system for relational databases", VLDB, 2005, 14(4): p. 373-396.

[13] Geerts, F., A. Kementsietsidis, and D. Milano, "MONDRIAN: Annotating and querying databases through colors and blocks", ICDE, 2006

[14] Peter, B., C. Adriane, and C. James, "Provenance management in curated databases", SIGMOD, 2006

[15] Jinxi, X. and W.B. Croft, "Query expansion using local and global document analysis", SIGIR, 1996

[16] Yonggang, Q. and F. Hans-Peter, "Concept based query expansion", SIGIR, 1993

[17] Edith, C., K. Haim, and M. Tova, "Labeling dynamic XML trees", SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2002

[18] Igor, T., et al., "Storing and querying ordered XML using a relational database system", SIGMOD, 2002