유전체 데이터의 유래(Provenance) 관리를 위한 메타데이터의 설계+

송명선*, 장재우*, 엄정호**, 최동훈**
*전북대학교 컴퓨터공학과
**한국과학기술정보연구원
ms_song@dblab.chonbuk.ac.kr* , jwchang@chonbuk.ac.kr*, jhum@kisti.re.kr**,
choid@kisti.re.kr**

Design of Metadata for Provenance Management of Genome Data

Myoung-Seon Song*, Jae-Woo Chang*, Jung-Ho Um** , Dong-Hoon Choi**
*Dept of Computer Engineering, Chonbuk National University
**Korea Institute of Science and Technology Information

요 약

최근 의료 분야에 대한 관심이 높아짐에 따라 유전체 데이터를 수집하고 관리하여 분석하는 기술에 대한 많은 연구가 수행되고 있다. 유전체 데이터는 크게 유전체 데이터를 분석하는 전처리단계와 유전체 데이터로부터 변이된 유전체 데이터를 생성하는 후처리단계를 통해 분석된다. 이러한 분석 과정은 많은 시간이 소요되며, 후처리단계에서 결과 데이터는 분석 알고리즘 및 처리 기법에 따라 상이한 결과 데이터를 생성한다. 또한, 유전체 데이터의 각 파이프라인 별 분석된 데이터의 관리가 필요하다. 본 논문에서는 유전체 데이터의 특성을 고려하여, 유전체 데이터 유래 관리를 위한 메타데이터를 설계한다. 아울러 데이터 유래 메타데이터는 자신의 이전데이터들의 결과데이터에 신속한 접근이 가능해야하며, 자신과 유사한 데이터 유래를 지닌 파이프라인의 상세 정보를 신속하게 검색하는 색인구조가 필요하다. 따라서 이를 고려한 유래 메타데이터 검색 알고리즘을 설계한다.

1. 서론

최근 신약 개발, 질병치료 등 의료 분야에 관심이 높아짐에 따라 이를 위한 기반 기술인 생명정보학(bioinfomatics)에 대한 다양한 연구가 수행되고 있다. 생명정보학이란, 생물자원 정보를 수집 및 관리하여 분석하는 기술로써, 대표적인 응용분야에는 DNA 염기서열분석, RNA 특질 발현 확률 분석 등이 존재한다. 이러한 생명정보학에서 사용되는 데이터를 유전체 데이터라 한다.

유전체 데이터는 여러 파이프라인을 걸쳐 분석된다. 분석 파이프라인은 크게 유전체 데이터를 분석하는 전처리단계와 유전체 데이터로부터 변이된 유전체 데이터를 생성하는 후처리단계로 구성된다. 전처리단계는 전체 유전체 데이터에 대해 색인을 구성하고, 미리 생성되어 있는 유전체 지도를 기반으로 데이터 맵핑(Mapping) 및 태깅(Tagging)을 수행한다. 유전체 데이터는 약 30억 개 이상의 염기 서열을 가진 3GB 이상의 대용량 데이터로써, 모든 염기 서열 조합에 대해 분석 작업을 수행하기 위해서는 많은 시간이 소요된다. 그러나 입력된 데이터가 동일할 경우, 동일한 결과 데이터를 생성한다. 한편, 후처리단계에서는 각종 변이 알고리즘에 의해 변이 데이터를 생성하고, 이를 분석하여 최종 결과 집합들을 생성한다. 이때 변이 알고리즘의 생성 및 알고리즘 적용 순서 등에 의해, 매번 생성되는 분석 결과가 상이하다. 아울러 동일하게 데이터의 일부에 대한 삽입 및 삭제 작업을 수행하더라도, 삽입되거나 삭제되는 염기 서열의 위치에 따라 다른 결과가 생성된다. 이와 같이 유전체 데이터 분석 시 다음과 같은 특성이 존재한다. 첫째, 유전체 데이터는 대용량 데이터로써 분석 시 파이프라인별 많은 시간이 소요된다. 둘째, 후처리단계에서 결과 데이터는 분석 알고리즘 및 처리 기법에 따라 상이한 결과 데이터를 생성한다. 마지막으로 유전체 데이터 각 분석 과정을 통해 분석된 데이터의 관리를 필요로 한다.

본 논문에서는 이러한 유전체 분석 과정의 특성을 고려하여 유전체 데이터 유래(Provenance) 관리를 위한 메타데이터의 설계한다. 유전체 데이터 유래란 분석 시 알고리즘의 적용 순서, 알고리즘의 종류, 반복 수행 횟수 등 유전체가 분석되는 과정에 대한 모든 상세 사항을 기록하고 관리하는 것을 말한다. 이를 통해 유전체 분석 중 발생하는 장애에 쉽게 대응할 수 있으며, 사용자가 원하는 파이

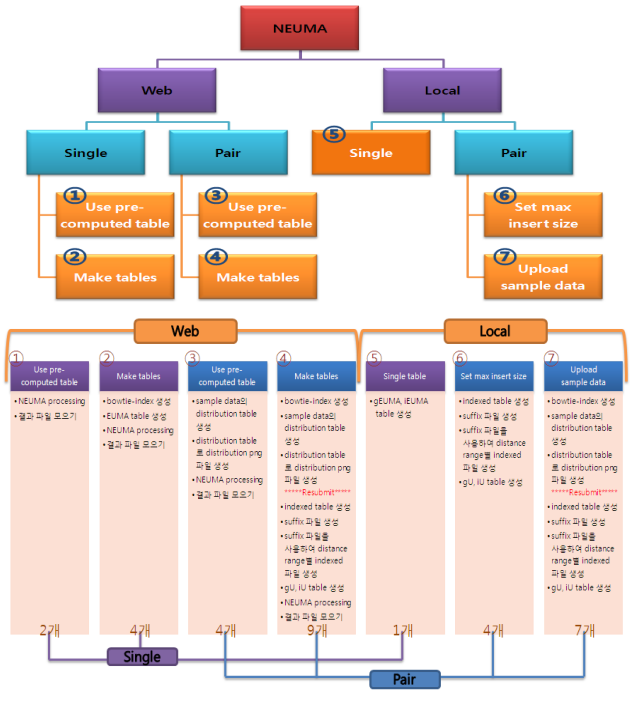
+ 이 논문은 한국과학기술정보연구원 주요산업 '사이버인프라 환경을 위한 차세대 기반기술 개발 산업'의 지원을 받아 수행한 연구결과임
+ 이 연구에 참여한 연구자는 2단계 BK21사업의 지원비를 받았음

프라인으로부터 유전체 분석 과정의 재수행이 가능하다. 아울러 변이의 순서 및 원인을 보다 원활하게 파악하여 최종 결과 데이터 분석을 용이하게 한다. 또한, 제안하는 유래 관리 메타데이터를 효율적으로 관리하기 위해 메타데이터 관리 알고리즘을 설계한다.

본 논문의 구성은 다음과 같다. 첫째, 2장에서는 기존의 유전체 데이터 유래의 기존 연구를 살펴보고, 3장에서는 제안 기법에 대해 기술한다. 마지막으로 4장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

유전체 데이터 분석에 관한 연구로 NEUMA[3]가 존재한다. NEUMA는 매핑 지역의 기대치 정규화 분석 도구로써, 유전체 데이터로부터 질병 등과 같은 염기 서열 그룹의 발현 확률을 측정한다. NEUMA는 참조데이터 존재 여부, 분할 작업 여부 등에 의해 (그림 1)과 같은 7가지의 작업 종류로 구분된다. 만약 참조데이터가 존재하지 않는 경우(그림 1의 2, 4, 6, 7의 경우) 유전체 데이터 분석을 위한 참조데이터를 생성해야 한다. 이때, 모든 유전자 염기 서열의 조합을 통해 유효한 조합인지 확인하고, 각 조합별 빈도를 측정하여 참조테이블을 생성하기 때문에 많은 시간이 소모된다. 한편, 7가지의 작업들은 서로 다른 수행 단계를 지니기 때문에, 한명의 사용자가 다양한 분석 작업을 수행하는 경우, 이에 대한 데이터 및 중간 분석 결과물을 관리되어야 한다. 그러나 NEUMA에서는 중간결과 데이터에 대한 상세 정보들을 사용자가 직접 관리해야 하는 단점이 존재한다. 이는 다수의 사용자 그룹에 의한 대용량 유전체 데이터 분석 작업 수행 시, 데이터의 관리를 어렵게 한다.



(그림 1) NEUMA web Process

3. 유전체 데이터의 유래(Provenance) 관리를 위한 메타데이터의 설계

본 절에서는 유전체 데이터의 유래 관리를 위한 메타데이터 및 이를 관리하기 위한 알고리즘을 설계한다. 이를 위해 다음과 같은 유전체 분석 과정의 특성을 고려해야 한다. 첫째, 유전체 데이터는 대용량 데이터로써 분석 시 파이프라인별 많은 시간이 소요된다. 둘째, 후처리단계에서 결과 데이터는 분석 알고리즘 및 처리 기법에 따라 상이한 결과 데이터를 생성한다. 마지막으로 유전체 데이터 각 분석 과정을 통해 분석된 데이터의 관리를 필요로 한다.

<표 1>은 데이터 유래를 관리하기 위한 메타데이터이다. 유전체 데이터 별로 자신의 정보(분석 도구의 종류, 데이터 타입, 수행단계)를 저장하며, 자신의 유래 정보<표 2>를 저장한다. 이를 통해, 현재 분석중인 유전체 데이터를 정보를 관리하며, 초기 유전체 데이터부터 현재 분석을 진행 중인 유전체 데이터 사이의 데이터 유래를 관리한다. 아울러 데이터 유래를 저장하는 HistorySet은 구조체로 되어 있으며(<표 2>), 각 파이프라인 별 데이터의 분석 도구의 종류, 데이터의 수행 단계, 데이터가 기록된 경로를 저장한다.

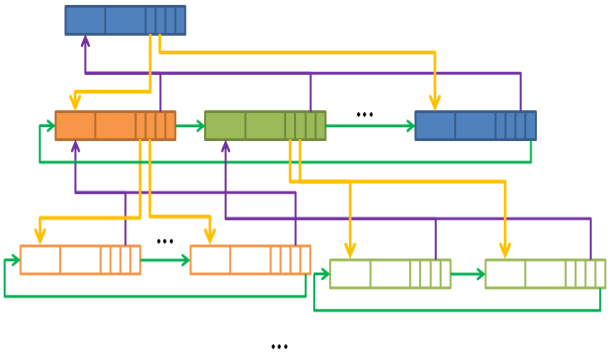
<표 1> 데이터 유래를 관리하기 위한 메타데이터

구성 요소	설 명
offset (int형)	레코드 오프셋
AppType (string형)	분석 도구 종류
DataType (string형)	데이터 타입
ProcessStep (int형)	수행 단계
HistorySet	데이터 유래 관리

<표 2> History 구조체

구성 요소	설 명
re_AppType[i] (int형)	데이터의 분석 도구 종류
re_ProcessStep[i] (string형)	데이터의 수행 단계
re_DataPath[i] (string형)	데이터가 기록된 경로

아울러, 제안하는 데이터 유래 메타데이터를 효율적으로 관리하기 위해 데이터 유래 관리를 위한 메타데이터 관리 알고리즘을 설계한다. 유래 메타데이터 관리를 위해 다음과 같은 사항을 고려한다. 첫째, 유전체 메타데이터의 유래는 시스템 장애에 의해 현재 진행 중인 작업이 중단되었을 때, 이를 신속하게 복구하기 위해 사용될 수 있다. 현재 중단된 작업의 바로 이전 단계의 결과 데이터를 로드하여 현재 수행하고자 하는 작업을 재수행 할 수 있어야 한다. 뿐만 아니라 이전 단계 결과 데이터에 대한 신속한 접근이 필요하다. 둘째, 전체 분석 과정에 대한 정보를 수집하여 최종 결과 데이터의 생성 원인을 확보하는 작업에 사용될 수 있다. 이를 위해 초기 유전체 데이터부터, 최종 결과 데이터까지 수행된 모든 파이프라인의 정보에 대해 접근이 용이해야 한다. 마지막으로, 유전체 데이터 변이 결과는 변이 알고리즘의 조합이나 순서, 반복 횟수 등에 의해 차이가 발생한다. 이러한 유전체 변이의 차이점은 유전체 데이터 분석 시 매우 중요하게 사용된다. 따라서 이전 단계까지의 파이프라인은 동일하지만 그 결과 데이터가 상이한 데이터에 대해 데이터 유래의 유사도에 기반을 둔 검색이 지원 가능해야한다. 즉, 현재 자신의 파이프라인을 기준으로 모든 이전 단계 파이프라인의 상세 정보에 신속한 접근을 지원하고, 자신과 유사한 데이터 유래를 지닌 파이프라인들의 상세 정보를 신속하게 검색하는 색인구조가 요구된다. 이를 위해 (그림 2)와 같은 Provenance Index Tree(PI-Tree)를 설계한다.



(그림 2) 유래 메타데이터 관리를 위한 PI-Tree

PI-Tree의 노드는 (그림 3)과 같다. 동일한 유전체 분석 과정이 여러 번 반복 수행될 수 있기 때문에, 자식 노드 수가 급격히 증가할 수 있다. 따라서 각 노드는 첫 번째와 마지막 자식 노드의 링크를 지니기 때문에 자식 노드의 리스트를 관리한다. 이때, 자식 노드들은 자신의 형제 노드의 링크를 저장하기 때문에, 동일한 부모노드를 지니는 모든 노드들을 빠르게 검색 가능하다. 따라서 데이터 유래의 유사도에 기반을 둔 검색이 용이하다. 아울러 설계하는 PI-Tree는 부모노드의 링크를 지니고 있기 때문에 현재 자신의 파이프라인을 기준으로 모든 이전 단계 파이프라인의 상세 정보에 신속한 접근을 지원한다.

파이프라인 ID	메타데이터 offset	첫번째 자식 노드의 링크	마지막 자식 노드의 링크	부모 노드의 링크	다음 형제 노드의 링크

(그림 3) PI-Tree의 노드의 구성

Algorithm1. 유래 관리 메타데이터 삽입 및 삭제

```
input  : inputMeta     //삽입 혹은 삭제하고자 하는
                       //메타데이터
       : parentMeta    //부모 메타데이터
       : option        //검색 옵션(ins, del)
output : boolean       //메타데이터 삽입/삭제 성공 여부

- Gobal newoffset     //다음 메타데이터가 저장될 offset
- BUtree currentTree  //현재 메타데이터에 대한 색인 트리

1.  BUtreeNode node= new BUtreeNode();
2.  if(option == ins){          //데이터 삽입
3.      writeMeta(inputMeta);
4.      BUtreeNode pnode = findTree(parentMeta.offset);
5.      node.setInfo(inputMeta);
6.      pnode->lastChild->sibling = node;
7.      pnode->lastChild = node;
8.      node->parent = pnode;
9.      node->sibling = pnode->firstChild;
10. }else if(option == del){  //데이터 삭제
11.     node = findTree(inputMeta.offset);
12.     Nodeset descs = Finddescendants(node);
13.     //모든 후손의 메타데이터 삭제
14.     while(descs != NULL){
15.         Metadata tmpmeta=getMeta(descs.offset);
16.         deleteMeta(tmpmeta);
17.     }
18.     //메타데이터 인덱스 삭제 및 링크수정
19.     DeleteTreenodeset(descs);
20.     FixLinksToDel(node);
21.     DeleteTreenode(node);
22. }
```

(그림 4) 유래 관리 메타데이터의 삽입/삭제 알고리즘

(그림 4)는 유래 관리 메타데이터의 삽입 및 삭제 알고리즘이다. 입력받은 옵션을 판단하여 삽입 및 삭제를 판단한다. 메타데이터를 삽입하는 경우, 삽입하고자하는 메타데이터를 기록한 후, 부모노드의 링크, 첫 번째 자식 노드 및 마지막 자식 노드에 대한 링크를 저장한다(line 3~9). 한편, 데이터 삭제하는 경우, 삭제하고자하는 메타데이터의 오프셋을 계산하여 삭제하고자하는 노드의 인덱스에 접근 후 인덱스의 정보를 통해 자식 노드들의 정보를 저장한다(line 11~12). 저장한 자식 노드들의 정보가 null이 될 때까지 링크를 통해 자식노드들의 메타데이터 및 정보를 삭제한다(line 14~16). 마지막으로 자신의 메타데이터 인덱스 삭제 후 링크를 수정한다(line 19~21).

Algorithm2. 유래 관리 메타데이터 검색

```
input  : in_pipeline  //유래(혹은 유사도)를 찾고자 하는
                      //파이프라인 노드
       : numMeta      //검색 결과 수 저장을 위한 변수
       : option       //검색 옵션(prov, sim, restore)

output : Resultset    //검색된 MetadataSet
       : numMeta      //검색된 Metadata의 수

- Metadataset Resultset = EMPTY;
- numMeta = 0;

1.  BUtreeNode node = in_pipeline;
2.  if(option == prov){  //데이터 유래 검색
3.      while(node != NULL){
4.          Metadata tmpmeta = getMeta(node.offset);
5.          Resultset.add(tmpmeta);
6.          numMeta++;
7.          node = node->parent;
8.      }
9.  }else if(option == sim){  //데이터 유사도 검색
10.     BUtreeNode startNode;
11.     startNode = node->parent->firstChild;
12.     node = startNode;
13.     if(startNode != NULL){
14.         do{
16.             Metadata tmpmeta = getMeta(node.offset);
17.             Resultset.add(tmpmeta);
18.             numMeta ++;
19.             node = node->sibling;
20.         }while(node != startNode);
21.     }
22. }else{  //복구를 위한 이전 파이프라인 검색
23.     if(node != NULL){
24.         node = node->parent;
25.         Metadata tmpmeta = getMeta(node.offset);
26.         numMeta ++;
27.         Resultset.add(tmpmeta);
28.     }
29. }
30. return Resultset;
```

(그림 5) 유래 관리 메타데이터의 검색 알고리즘

(그림 5)은 유래 관리 메타데이터의 검색 알고리즘이다. 유래 관리 메타데이터의 검색은 옵션에 따라 데이터 유래, 유사도, 그리고 복구를 위한 이전 파이프라인 검색으로 구성된다. 첫째, 데이터 유래 검색은 이전의 모든 파이프라인의 결과 데이터에 대해 신속하게 접근하기 위해 설계한다. 오프셋을 사용하여 자신의 파이프라인에 대한 정보를 저장한 후, 부모노드링크를 통해 원하는 파이프라인을 검색한다.(line 3~7). 둘째, 데이터 유사도 검색은 유전체 데이터 변이에 대해 알아보기 위해 설계한다. 우선 첫 번째 자식노드 링크를 통해 첫 번째 자식 노드의 데이터를 저장 후, 형제 노드의 링크를 통해 자신의 형제 노드들의 데이터에 접근하여 저장한다.(line 10~20). 마지막으로 복구를 위한 이전 파이프라인 검색은 유전체 데이터 분석 작업이 중단되었을 경우 신속하게 이전 단계의 결과 데이터에 접근하기 위해 설계한다. 자신의 부모노드 링크를 통해 보모노드의 메타데이터를 접근하여 부모노드의 결과데이터를 불러온다(line 23~28).

4. 결론

본 논문에서는 유전체 데이터 분석 과정의 특징을 고려하여 유전체 데이터 유래 관리 메타데이터를 설계하였다. 아울러, 제안하는 데이터 유래 메타데이터를 효율적으로 관리하기 위해 메타데이터 관리 알고리즘을 제시하였다.

유전체 데이터 유래 관리 메타데이터를 통해 데이터 분석 시간을 줄일 수 있으며, 비슷한 유래를 가진 결과 데이터와 비교 및 분석이 가능하다. 아울러, 각 분석 결과데이터의 관리가 용이하다. 또한, 유래 관리 메타데이터를 관리하기 위한 알고리즘을 통해 현재 진행 중인 작업에 이상이 발생했을 경우 신속하게 이전 단계의 결과 데이터에 접근 가능하다. 또한, 전체 분석 과정에 대한 정보 수집이 용이하여 데이터 유래가 유사한 데이터 및 자신을 기준으로 이전의 파이프라인의 결과데이터로 신속한 접근이 가능하다.

향후 연구로는 설계한 데이터 유래 메타데이터 및 메타데이터 관리 알고리즘을 실제로 구현하여 성능평가를 수행하는 것이다.

참고문헌

[1] LD Stein. "The case for cloud computing in genome informatics", Genome Biology, Vol.11:207, Issue 5, May, 2010

[2] Biotech Policy Research Center, 2009

[3] S. Lee et al, "Accurate quantification of transcriptome from RNA-Seq data by effective length normalization" Nucleic Acids Research, Vol.39, No.2, Jan , 2011

[4] Y.L. Simmhan, B.Plale, and D.Gannon, "A Survey of Data Provenance Techniques" in Technical Report TR-618:Computer Science Department, India University, 2005

[6] Mark Greenwood et al, "Provenance of e-Science Experiments - experience from Bioinformatics" In Proceedings of the UK e-Science All Hands Meeting. Nottingham, UK. pp223~pp226, 2003

[7] Zachary G.Ives et al, "The ORCHESTRA Collaborative Data Sharing System" ACM SIGMOD Record, V.37, No.6 September, 2008